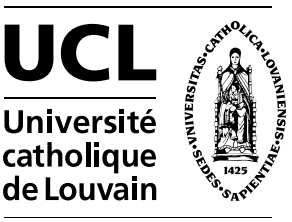
Institute of Information and Communication Technologies,
Electronics and Applied Mathematics

Nearest Stable System

François-Xavier Orban de Xivry



Thesis submitted in partial fulfillment of the requirements for the degree of
Docteur en sciences appliquées

Dissertation committee :
Yurii Nesterov (UCL, Advisor)
Paul Van Dooren (UCL, Advisor)
Philippe Lefèvre (UCL, Chair)
Pierre-Antoine Absil (UCL)
Michael Overton (New York University, USA)
Marc Van Barel (Katholieke Universiteit Leuven, Belgium)

Acknowledgements

Paul, Yurii, this is probably the best place to tell you how grateful I am for all the support you have shown during these four years. I would like to thank both of you for the inspiring ideas, the many useful pieces of advice and, most importantly, for your time and patience. I enjoyed our interactions and you were an inexhaustible source of knowledge. I was told several times that I had a dream team of supervisors and I could not agree more.

Many thanks to Marc, Michael, Pierre-Antoine and Philippe for accepting to be a part of this. I enjoyed interacting with each of you. Michael, thank you for sharing a small part of your knowledge on variational analysis. Marc thank you for reminding me that simple tricks are often more efficient than big developments.

My parents have encouraged me a lot during these past few months as well, and have provided a quiet place where I could find inspiration and concentration. I thank them deeply for all they did during this special time. Likewise, I thank my brothers and sisters for being supportive in their own ways : being holiday destinations counts, for sure. I thank my, already many, nephews for bringing animation in the quiet house at times.

I would like to thank the good old eulerians for providing a stimulating place to start a thesis; I would like to thank the young eulerians for providing an equally stimulating place to complete it; and I would like thank the everlasting eulerians for making sure it continues. Some lobbied to have their names here so I am writing all of them : Adeline, Adrien, Arnaud B., Augustin, Aurélien, Ben, Bob, Bruno, Caro, Etienne, Gaut K., Guillaume L., Guillaume V., Heba, JJ, Julien, Oli Gou, Marie, Nico B., Nico G., Nico H., Pierre B., Pierre D., P.-Y., Quentin, Romain, Seb C., Vincent T., and the others : thank you. A special thank to Nico B. and Nico G. for helping me with manifolds and optimization related questions respectively. I would also like to emphasize the essential help of Isabelle and Nathalie, I am grateful to them.

Finally, a huge thank to all my friends for their support and for sometimes making the effort of understanding what I was doing. I will still have numerous occasions to further thank you.

This thesis could not have been completed without the financial support of the DYSCO Interuniversity Attraction Poles and the Université catholique de Louvain, their contribution is gratefully acknowledged.

"Es-tu prêt spadassin?
- Ma main ne tremblera pas!
- Va!"

-Molière

Contents

Notations

Basics

j	:	Imaginary unit;
$\mathrm{Re}\{s\}$	:	Real part of the complex number s;
$\mathrm{Im}\{s\}$	:	Imaginary part of the complex number s;
$\bar{s}$	:	Conjugate of s;
$\lvert s\rvert$	:	Modulus of s;

Inner products and norms

$\langle \cdot, \cdot \rangle$	:	Real-valued scalar product;
$\langle \cdot, \cdot \rangle_{\mathbb{C}}$	:	Complex-valued scalar product;
$\langle \cdot, \cdot \rangle_E$,	:	Generic scalar product on a vector space E;
$\lVert \cdot \rVert$	:	Norm;
$\lVert \cdot \rVert_*$	:	Dual norm of $\lVert \cdot \rVert$;
$\lVert \cdot \rVert_2$	:	2-norm;
$\lVert \cdot \rVert_F$	:	Frobenius norm;

Sets

$\mathbb{N}$	:	Natural numbers;
$\mathbb{R}$	:	Real numbers;
$\mathbb{C}$	:	Complex numbers;
$\mathbb{F}$	:	Generic field;
$\mathbb{R}^n$, $\mathbb{C}^n$	:	Real and complex vectors of size n;
$\mathbb{R}^{m\times n}$, $\mathbb{C}^{m\times n}$	:	Real and complex matrices of size $m \times n$;
$\mathbb{S}^n$	:	Set of stable systems of size n, matrices and polynomials alike;
$\mathbb{S}_P$	:	Set of systems determined stable in the Lyapunov sense by stability certificated P;
$\mathbb{P}_X$	:	Set of valid stability certificates in the Lyapunov sense for a given matrix X;
$\mathbb{R}_+$	:	Nonnegative real numbers;
$\mathbb{K}_+^n$	:	Positive semidefinite matrices of size n;
$\mathbb{K}_{++}^n$	:	Positive definite matrices of size n;
$N_{\mathcal{C}}(x)$	:	Normal cone to $\mathcal{C}$ at x;
$T_{\mathcal{C}}(x)$	:	Tangent cone to $\mathcal{C}$ at x;

Topology

$\subset,\ \cup,\ \cap,\ \setminus$	:	Inclusion, union, intersection and difference of two sets;
$\operatorname{int}\mathcal{C}$	:	Interior of $\mathcal{C}$;
$\operatorname{cl}(\mathcal{C})$	:	Closure of $\mathcal{C}$;
$\operatorname{co}\mathcal{C}$	:	Convex hull of $\mathcal{C}$;
K^*	:	Dual cone associated with the cone K;
$\operatorname{pos}\mathcal{C}$	:	Positive hull of a set;

Functions

$\mathcal{A}^*$	:	Adjoint of a linear operator $\mathcal{A}$;
$\operatorname{dom} f$	:	Domain of the function f;
f_*	:	Optimal value of a real-valued function f;
$f \circ g$	:	Compostion of g into f;
$f * g$	:	Convolution of f and g;
$f(x_0)$ or $f(x)\big\vert_{x=x_0}$	:	f evaluated at $x = x_0$;
$\partial f(x)$	:	Set of subgradient of f at x;
$\frac{\partial f}{\partial x}$	:	Partial derivative of f with respect to a scalar x;
∇f (or f')	:	Gradient of a real-valued function f;
$\nabla_Y f$	:	Partial gradient of a real-valued function f with respect to Y;
$\nabla^2 f$ (or f'')	:	Hessian of a real-valued function f;
$Df(x)[h]$	:	Directional derivative of f in a direction h at x;
$b_X(P)$	:	Log-det self-concordant barrier for the set $\mathbb{P}_X$;
$b_P(X)$	:	Log-det self-concordant barrier for the set $\mathbb{S}_P$;

Vectors, polynomials and matrices

e_i	:	Vector of zeros with a 1 at the ith position;
$\mathbb{1}$	:	Vector of all ones;
$\operatorname{diag}(x)$	:	Diagonal matrix obtained from the elements of x;
$p(\lambda)$	:	Polynomial in λ;
$p(\lambda; r)$	:	Polynomial in λ formed from the vector of roots r;
$[p(\lambda)]_c$	:	Vector of coefficient of a polynomial;
$p_\star(\lambda)$	:	Paraconjugate polynomial;
π_λ^n	:	Vector of monomials of λ from λ^n to 1;
I, I_n	:	Identity matrix, identity matrix of size n;
X^T	:	Transpose of X;
X^*	:	Conjugate transpose of X;
X_*	:	Optimal solution;
$X_{re},\ X_{im}$	:	Real and imaginary parts of X;

$\lambda_i(X)$	:	ith eigenvalue of X;
$\lambda_i(x)$	:	ith roots of $x(\lambda)$;
$\lambda_{\min}(X),\ \lambda_{\max}(X)$	:	Minimum and maximum eigenvalues of a matrix $X = X^*$;
$\lambda_{\min}(x),\ \lambda_{\max}(x)$	:	Minimum and maximum roots of $x(\lambda)$;
$\alpha(X),\ \rho(X)$	:	Spectral abscissa and spectral radius of the matrix X;
$\alpha(x),\ \rho(x)$	:	Root abscissa and root radius of the polynomial $x(\lambda)$;
$\mathrm{diag}(X)$	:	Diagonal elements of a matrix;
$\mathrm{Tr}(X)$	:	Trace of X;
$\det(X)$	:	Determinant of X;
$\otimes$	:	Kronecker product;
$\odot$	:	Hadamard product;
$\mathrm{vec}(X)$	:	Vector obtained by stacking up the columns of X;
$\succ 0,\ 0 \prec$	:	Positive definite, negative definite.
$\succeq 0,\ 0 \preceq$	:	Positive semidefinite, negative semidefinite.

Miscellaneous

sup, inf, max, min	:	Supremum, infimum, maximum and minimum;
$\arg f$	:	Argument of a function f;
$\approx$	:	Approximately equal;
$\propto x$	:	Proportional to x;
$\mathcal{O}(\cdot)$	:	*Big-Oh* asymptotic complexity bound;
$o(t)$	:	terms that are decreasing at least proportionally to t, i.e with the property that $\lim_{t\to 0} o(t)/t = 0$;
$\stackrel{\text{def}}{=}$	:	Equals by definition;
$[X]_+$	:	Maximum of the elements of X and 0;
$[X]_-$	:	Minimum of the elements of X and 0;

Preface

Stability is a universal concept which we experience in our everyday lives. When sitting in a airliner bound to a remote destination, one particularly dislike the trembling, shaking and the other uncomfortable behaviors that result from the encounter with a perturbation such as a sudden strong wind. Under the action of the perturbation the plane can start to oscillate about its main axis. But as soon as the perturbation is left behind, the undesirable oscillations are damped over time and the flight quickly becomes enjoyable again. Such a behavior is typical of a dynamically stable airliner. Had the plane not been stable, it would not have returned to its initial equilibrium forcing the pilot to compensate for the oscillations. Many additional examples could be given and all of them would emphasize the central place held by stability in the control field.

The question of finding the nearest stable system appears to be a challenging problem for the research community where one sometimes faces unstable models created from the perturbed data of a stable system. Though the question has been mentioned frequently in the literature, the absence of a reference which solves the problem motivates our research of dedicated algorithms. Very few existing methods are available to solve the problem due to the nonsmooth, nonconvex nature of the stable set. The importance and the complexity of the problem nonetheless calls for additional methods that are specifically designed to tackle the nearest stable system problem.

The goals that we want to achieve are, broadly, the development of new and efficient optimization methods for solving the nearest stable system problem. Many different approaches can be considered and each of them can potentially lead to an innovative way of solving the problem. We will base our work on the well-established theory developed by the optimization community, and more specifically by the convex optimization community, and favor an approach where the structure of the different formulations is fully exploited.

Under some specific conditions closed form solutions for the nearest stable system are available. This is the case for example when additional constraints on the shape of the solution are formulated. But a closed form solution is otherwise illusory since the stable set is nonconvex and even checking the optimality of a solution once it is computed is challenging. Therefore, numerical methods that are able to find a nearest stable matrix locally are our only available possibility. An approximate solution is still satisfactory provided that it is stable. As systems can be continuous-time or discrete-time, algorithms should ideally be able to deal with any possible setting.

Many references will be used to build our theory and it is already worth underlining the thesis of Tom d'Haene [33] whose work on a related topic emphasized the control point of view whereas our contribution is best appreciated from an optimization point of view. We focus on the development of innovative formulations and efficient methods dedicated to the problem. We thus develop several directions of research which potentially produce as many algorithms. The first direction of research consists in developing an iterative algorithm which works on the most usual formulation of the problem. We thus work inside a nonconvex feasible domain. The resulting algorithm is conceptually and surprisingly simple given the complicated setting it must deal with.

The second direction we develop focuses on algorithms which avoid the complexity encountered in the first case by reformulating the problem. The purpose of the reformulations is to provide an easier framework in which the problem can be solved even though we will see they can appear more complex at first.

The third and final direction of research exploits the structure brought into the problem by polynomials. We thus adapt our algorithms when possible and investigate other ways of solving the polynomial problem.

The main contributions of this thesis are given by the algorithms that we detail throughout this thesis. Those algorithms are able to solve the nearest stable system problem in several configurations : matrices and polynomials, continuous-time and discrete-time, real and complex. Several possible alternative formulations are also suggested which could be used to solve the problem as well. Another contribution is given by the survey from the literature related to the nearest stable system and its particularities and the geometry of the stable set.

The results produced by our research were communicated in two publications in peer-reviewed journals [77, 76].

Our manuscript is divided in four chapters.

Chapter 1 recalls several theoretic concepts that are used in this thesis. The content is intentionally limited to the most important definitions and theorems which are used.

Chapter 2 introduces the nearest stable system problem, explores the geometry of the stable set and describes optimality conditions for the problem. The chapter is essentially a survey of the literature.

Chapter 3 presents the results we obtained for matrices. Two methods for solving the nearest stable matrix problems are investigated in the chapter and convergence results are given for both of them. The chapter also contains several reformulations of the nearest stable system problem for the continuous-time case.

Chapter 4 is dedicated to polynomials. Some of the algorithms contained in the previous chapter are extended to handle polynomials. We also presents other algorithms

which exploit the structure of the polynomials. The emphasis is put on the efficiency of the methods and some time is thus taken to discuss related aspects.

Chapter 1

Preliminaries

In this chapter we give a general introduction to the background areas of our research. The nearest stable system is at the crossroad of three main fields : control, optimization and numerical linear algebra.

Control is the essential field of our problem and it also provides the original motivation for our problem. With its well-established theory, the optimization field provides us with the tools to solve the problem in an efficient way. Because of our focus on analytical solutions, numerical linear algebra is at the heart of our work, and its use in our developments is widespread.

This chapter is organized as follows :

Section 1.1 recalls the most basic concepts of linear algebra and also provides a more selective introduction to the mathematical objects we will use throughout : matrices and polynomials.

Section 1.2 introduces important notions of analysis such as convexity of a set, normal and tangent cone to a set, differentiability and illustrates them through several examples that will be reused at multiple times.

Section 1.3 emphasizes several results related to optimality conditions, both necessary and sufficient. We choose to present some results under the form of variational inequalities which will be helpful to present necessary optimality conditions for the nearest stable system in the second chapter.

Section 1.4 first presents several basic elements of systems theory. Some time is then spent to present the spectral abscissa and the spectral radius which are going to be used occasionally in replacement of the Lyapunov stability criterion.

1.1 Linear Algebra

1.1.1 Norms and scalar products

A vector space is a set endowed with a commutative addition law and an external composition law with respect to a a scalar field $(\mathbb{F}, +, \cdot)$. In our work, $\mathbb{F}$ is most often the field of complex numbers $\mathbb{C}$ or the field of real numbers $\mathbb{R}$.

Definition 1.1. *Let E be such a vector space on a field $\mathbb{F}$. Then a norm on E is a function*

$$\|\cdot\| \ : \ E \to \mathbb{R}$$

such that $\forall x \in E$,

$$\begin{aligned} &1) \ \ \|x\| \geq 0 \\ &2) \ \ \|x\| = 0 \quad \Leftrightarrow \quad x = 0 \\ &3) \|\alpha x\| = |\alpha|\|x\| \quad \forall \alpha \in \mathbb{F} \ . \end{aligned}$$

Definition 1.2. *An inner product on E is a function*

$$\langle \cdot, \cdot \rangle \ : \ E \times E \to \mathbb{F}$$

that satisfies the following properties

$$\begin{aligned} &1) \langle \alpha x + \beta y, z \rangle = \alpha \langle x, z \rangle + \beta \langle y, z \rangle, \quad \forall x, y, z \in E \quad \textit{and} \quad \forall \alpha, \beta \in \mathbb{F} \\ &2) \qquad \langle x, x \rangle \geq 0 \quad \forall x \in E \quad \textit{and} \quad \langle x, x \rangle = 0 \Leftrightarrow x = 0 \\ &3) \qquad \langle x, y \rangle = \overline{\langle y, x \rangle} \quad \forall x, y \in E \ . \end{aligned}$$

In this thesis, we work nearly exclusively with real-valued scalar products and the third condition in Definition (1.2) is thus reduced to $\langle x, y \rangle = \langle y, x \rangle$.

All scalar products induce a norm by the relation $\|\cdot\| = \langle x, x \rangle^{\frac{1}{2}}$ but not all norms are induced by a scalar product. The standard real-valued scalar product on $\mathbb{C}^n$ is defined as

$$\langle x, y \rangle = \mathrm{Re} \left\{ \sum_i x_i \bar{y_i} \right\} \qquad \forall x, y \in \mathbb{C}^n$$

and induces the usual $2-$norm $\|\cdot\|_2$ on $\mathbb{C}^n$. We denote by $\langle \cdot, \cdot \rangle_{\mathbb{C}}$ the standard complex-valued scalar product on $\mathbb{C}^n$ by

$$\langle x, y \rangle_{\mathbb{C}} = \sum_i x_i \bar{y_i} \qquad \forall x, y \in \mathbb{C}^n \ .$$

For a given norm $\|\cdot\|$, we define the *dual norm* as

$$\|y\|_* = \max_{\|x\| \leq 1} \langle x, y \rangle \ . \tag{1.1}$$

A norm is self-dual if $\|\cdot\| = \|\cdot\|_*$.

Example 1. *The 2-norm is self-dual since*

$$\|y\|_2 = \max_{\|x\|_2 \leq 1} \langle x, y \rangle \ .$$

Induced norms satisfy a number of useful properties some of which will be extensively used in this manuscript.

Theorem 1.3 (Cauchy-Schwarz)**.** *Let E be a vector space over a field $\mathbb{F}$, equipped with a scalar product $\langle \cdot, \cdot \rangle$. Then $\forall x, y \in E$*

$$|\langle x, y \rangle| \leq \|x\|_* \|y\| \ .$$

The Cauchy-Schwarz theorem is used to derive other algebraic properties; probably the most well-known is the triangle inequality which we recall now.

Property 1.4 (Triangle inequality)**.** *Let E be a vector space equipped with an induced norm $\| \cdot \|$. Then, $\forall x, y \in E$*

$$\|x + y\| \leq \|x\| + \|y\| \ .$$

The *reverse triangle inequality* that we now introduce is a consequence of the triangle inequality applied to $\|x + y - y\|$.

Property 1.5 (Reverse triangle inequality)**.** *Let E be a vector space equipped with a induced norm $\| \cdot \|$. Then, $\forall x, y \in E$*

$$\|x + y\| \geq \|x\| - \|y\| \ .$$

Adjoint of an operator

Definition 1.6 (Linear operator)**.** *Let E and F be two vector spaces over a field $\mathbb{F}$ such as $\mathbb{R}$ or $\mathbb{C}$. A linear operator $\mathcal{L}$ is a function*

$$\mathcal{L} \ : \ E \to F$$

that satisfies the following properties

$$\mathcal{L}(\alpha x + \beta y) = \alpha \mathcal{L}(x) + \beta \mathcal{L}(y) \quad \forall x, y \in E \quad \alpha, \beta \in \mathbb{F} \ .$$

Suppose that we have a linear operator $\mathcal{L} \ : \ E \to F$, where the two vector spaces E and F are each equipped with a scalar product $\langle \cdot, \cdot \rangle_E$ and $\langle \cdot, \cdot \rangle_F$ respectively.

Definition 1.7 (Adjoint of a linear operator)**.** *The adjoint $\mathcal{L}^*$ of $\mathcal{L}$ is a linear operator from $F \to E$, such that for any $x \in E$ and any $y \in F$*

$$\langle \mathcal{L} x, y \rangle_F = \langle x, \mathcal{L}^* y \rangle_E \ . \tag{1.2}$$

Moreover if $\mathcal{L} = A, \ A \in \mathbb{C}^{n \times n}$, then $\mathcal{L}^*$ is the conjugate transpose A^* of A which is also denoted by the asterisk. Naturally, the expression of the adjoint of an operator is dependent on the scalar product which is used.

1.1.2 Matrix, eigenvalue and positive definiteness

Let x be a vector in a vector space E. Vectors are represented using the column convention, i.e,

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \quad , \quad v_i \in \mathbb{F} \forall i$$

A vector $x = 0$ if and only if all its elements are 0. By e_i we denote a vector of zeros with a 1 at the ith position.

Let X be a matrix in a vector space E of dimension $n \times m$ over a field $\mathbb{F}$. A matrix X is composed of elements $X_{ij} \in \mathbb{F}$ placed in an array which we write

$$X = [X_{ij}]_{i,j=1}^{n,m} \ .$$

By $[X]_{ij} = X_{ij}$ we denote the element of the matrix at the intersection of the i^{th} row and the j^{th} column. Given a matrix $X \in \mathbb{C}^{n \times m}$, the transpose of a matrix is denoted by $X^T \in \mathbb{C}^{m \times n}$ and satisfies $[X^T]_{ij} = X_{ji}$. Its conjugate matrix X^* is an element of $\mathbb{C}^{m \times n}$ and satisfies $[X^*]_{ij} = \bar{X}_{ji}$.

A squared matrix is said to be symmetric if $X^T = X$, anti-symmetric if $X^T = -X$, Hermitian if $X = X^*$ and skew-Hermitian (or anti-Hermitian) if $X^* = -X$.

The trace operator of $X \in \mathbb{C}^{n \times n}$, denoted by $\mathrm{Tr}(\cdot)$, returns the sum of its diagonal elements. The determinant of a matrix is denoted by $\det(\cdot)$.

Scalar product for matrices

The standard real-valued scalar product on $\mathbb{C}^{n \times n}$ uses the trace operator $\mathrm{Tr}(\cdot)$

$$\langle X, Y \rangle = \mathrm{Re}\left\{\mathrm{Tr}(XY^*)\right\} = \mathrm{Re}\left\{\sum_{i,j} X_{ij}\bar{Y}_{ij}\right\} \tag{1.3}$$

and induces the Frobenius norm $\|X\|_F = \sqrt{\sum_{i,j} |X_{ij}|^2}$. We will frequently use the following properties associated to the Frobenius norm.

Property 1.8. *Let $X, Y, Z \in \mathbb{C}^{n \times n}$ and let $\langle \cdot, \cdot \rangle$ be the scalar product associated with the Frobenius norm. Then*

$$\langle XY, Z \rangle = \langle X, ZY^* \rangle = \langle Y, X^*Z \rangle$$

Given the importance of this property in this thesis, we prove the above result.

Proof. We first prove that $\langle XY, Z \rangle = \langle Z, XY \rangle$. Indeed, since the real part of the trace does not change under conjugation, we have

$$\langle XY, Z \rangle = \mathrm{Re}\left\{\mathrm{Tr}(XYZ^*)\right\} = \mathrm{Re}\left\{\mathrm{Tr}((XYZ^*)^*)\right\} = \langle Z, XY \rangle \ .$$

Then, we can prove the first equality $\langle XY, Z\rangle = \langle X, ZY^*\rangle$ by using associativity of the matrix product,

$$\langle XY, Z\rangle = \mathrm{Re}\left\{\mathrm{Tr}(XYZ^*)\right\} = \mathrm{Re}\left\{\mathrm{Tr}(X(ZY^*)^*)\right\} = \langle X, ZY^*\rangle \ .$$

The second equality $\langle XY, Z\rangle = \langle Y, X^*Z\rangle$ is proved by using the identity $\mathrm{Tr}(AB^*) = \mathrm{Tr}(B^*A) \quad \forall A, B \in \mathbb{C}^{m\times n}$. This identity implies that

$$\begin{aligned}\langle XY, Z\rangle &= \mathrm{Re}\left\{\mathrm{Tr}(XYZ^*)\right\}\\ &= \mathrm{Re}\left\{\mathrm{Tr}(Z^*XY)\right\}\\ &= \langle Z^*X, Y^*\rangle\\ &= \langle Y^*, Z^*X\rangle\\ &= \langle Y, ZX^*\rangle \ .\end{aligned}$$

□

The identity matrix of size $n \times n$ is denoted by I_n. When its dimensions are clear from the context the subscript is omitted and we simply write I.

A matrix $U \in \mathbb{C}^{n\times n}$ is said to be unitary if $UU^* = I = U^*U$. The real equivalent of a unitary matrix is called an orthogonal matrix $V \in \mathbb{R}^{n\times n}$ and satisfies $VV^T = I = V^TV$.

Example 2. *A norm is said to be unitarily invariant if $\|UAV^*\| = \|A\|$ for all unitary matrices U and V. The Frobenius norm and the $2-$norms are unitarily invariant.*

Matrix theory

We define two additional operator products in addition to the usual matrix products.

Definition 1.9 (Hadamard product)**.** *Let $X, Y \in \mathbb{C}^{n\times n}$. Then the Hadamard product is the element-wise product defined by*

$$[X \odot Y]_{ij} = X_{ij}Y_{ij} \ .$$

Definition 1.10 (Kronecker product)**.** *For any two matrices $X \in \mathbb{C}^{n_a\times n_b}$ and $Y \in \mathbb{C}^{n_c\times n_d}$, the Kronecker product of X and Y is given by*

$$Z = X \otimes Y = \begin{bmatrix} X_{11}Y & \cdots & X_{1n}Y \\ \vdots & \ddots & \vdots \\ X_{n1}Y & \cdots & X_{nn}Y \end{bmatrix}$$

where $Z \in \mathbb{C}^{n_an_c\times n_bn_d}$.

Note that unlike the Hadamard product, the Kronecker product is not commutative. The Kronecker product is often used to obtain the matrix representation of a linear operator.

Definition 1.11. *Let $X \in \mathbb{F}^{m\times n}$. Then the operator* $\mathrm{vec}(X)$ *returns the vector obtained by stacking up the columns of X, i.e, if $X = [x_1 \cdots x_n]$ where $x_i \in \mathbb{F}^n$ is the i^{th} column of X, then*

$$\mathrm{vec}(X) = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} .$$

Example 3. *The following identity is well-known. For any three given complex matrices X, Y and Z such that the product XYZ makes sense, then*

$$\mathrm{vec}(XYZ) = (Z^T \otimes X)\,\mathrm{vec}(Y) .$$

Let us now introduce a class of matrices that will be frequently used in this manuscript.

Definition 1.12 (Positive definite)**.** *A Hermitian matrix $X \in \mathbb{C}^{n\times n}$ is called positive definite, denoted $X \succ 0$, if*

$$v^* X v > 0 \qquad \forall v \in \mathbb{C}^n \backslash \{0\} ,$$

and is called negative definite if $-X \succ 0$.

Definition 1.13 (Positive semidefinite)**.** *A Hermitian matrix $X \in \mathbb{C}^{n\times n}$ is called positive semidefinite, denoted $X \succeq 0$, if*

$$v^* X v \geq 0 \qquad \forall v \in \mathbb{C}^n ,$$

and is called negative semidefinite if $-X \succeq 0$.

We denote the set of all positive definite matrices of size n by $\mathbb{K}^n_{++}$ and the set of all semidefinite positive matrices of size n by $\mathbb{K}^n_{+}$.

The following definition will be very useful.

Definition 1.14 (Cholesky factor)**.** *A matrix $L \in \mathbb{C}^{n\times n}$ is a Cholesky factor of $X \in \mathbb{C}^{n\times n}$ if it is lower triangular and*

$$X = LL^* .$$

Property 1.15 (Cholesky decomposition)**.** *The Cholesky decomposition of a matrix $X \in \mathbb{K}^n_{++}$*

$$X = LL^*$$

where L has a positive diagonal is unique.

Let us now recall a number of properties related to eigenvalues.

Definition 1.16 (Eigenvalue)**.** *Let $X \in \mathbb{C}^{n\times m}$ and let $(\lambda_i, v) \in \mathbb{C} \times \mathbb{C}^m$ be such that*

$$Xv = \lambda_i v, \quad v \neq 0 . \tag{1.4}$$

Then (λ_i, v) is called an eigenpair, λ_i is an eigenvalue of X and v is a right-eigenvector associated with λ_i.

We sometimes denote the eigenvalue λ_i as $\lambda_i(X)$ to prevent confusion. A matrix X living in a space of dimension n has n, possibly non-distinct, eigenvalues. The set formed of all, possibly non-distinct, eigenvalues of X is called the spectrum of X.

A positive definite matrix $X \in \mathbb{K}^n_{++}$ has n positive eigenvalues.

Let $\{\lambda_1, \ldots, \lambda_n\}$ be the spectrum of a matrix $X \in \mathbb{C}^{n\times n}$. A matrix X is called singular if $\lambda_i(X) = 0$ for some $i \in \{1, \ldots, n\}$.

Property 1.17 (Trace and determinant of a matrix)**.** *For any square matrix $X \in E$, two eigenvalue functions of X are given by the trace and the determinant of X and satisfy*

$$\mathrm{Tr}(X) = \sum_i \lambda_i(X) \ ,$$
$$\det(X) = \prod_i \lambda_i(X) \ .$$

Definition 1.18 (Nonnegative matrix [50])**.** *A matrix $X \in \mathbb{R}^{n\times n}$ is said to be nonnegative, denoted $X \geq 0$, if*

$$X_{ij} \geq 0 \qquad \forall i, j = 1, \ldots, n \ .$$

Definition 1.19 (Irreducibility [50])**.** *A nonnegative matrix $X \in \mathbb{R}^{n\times n}$ is said to be irreducible if there exists no permutation matrix P such that*

$$P^T X P = \left[\begin{array}{c|c} X_{11} & X_{12} \\ \hline 0 & X_{22} \end{array}\right] \ .$$

The following well-known theorem is due to Perron and Frobenius.

Theorem 1.20 (Perron-Frobenius [50])**.** *Suppose that a matrix $X \in \mathbb{R}^{n\times n}$ is nonnegative and irreducible. Then*

1. $\lambda_{\max} \in \mathbb{R}_+$*;*
2. $\lambda_{\max}$ *is a simple eigenvalue of X;*
3. $\exists u \in \mathbb{R}^n, u > 0 \ : \ u^T X = \lambda_{\max} u^T$ *.*

Even though we denote them differently, nonnegative matrices should not be confounded with positive definite and positive semidefinite matrices.

We now define the algebraic and geometric multiplicity of an eigenvalue. Those definitions will be used for discussions related to the Jordan form of a matrix and the singularities of the stable set.

Definition 1.21 (Algebraic multiplicity)**.** *Let λ_i be an eigenvalue of $X \in \mathbb{C}^{n\times n}$. Then we define the algebraic multiplicity $m_a(\lambda_i)$ as the cardinality of λ_i in the spectrum of X.*

Definition 1.22 (Geometric multiplicity)**.** *Let λ_i be an eigenvalue of $X \in \mathbb{C}^{n\times n}$. The geometric multiplicity $m_g(\lambda_i)$ is the dimension of the eigenspace related to λ_i, i.e.*

$$m_g(\lambda_i) = \dim(\mathrm{Ker}(\lambda_i I - X)),$$

where the operator Ker *denotes the kernel (or nullspace) of a matrix.*

Property 1.23. *Let λ_i be an eigenvalue of $X \in \mathbb{C}^{n\times n}$. Then*

$$1 \leq m_g(\lambda_i) \leq m_a(\lambda_i) \ .$$

In particular, if X has n distinct eigenvalues, then X has n distinct eigenvectors and $m_a(\lambda_i) = m_g(\lambda_i) = 1$.

Algebraic and geometric multiplicities are useful to characterize the eigenstructure of a matrix, i.e. the number of distinct eigenvalues and dimension of the eigenspaces associated with them. The same information can be deduced from some canonical matrix forms.

Theorem 1.24 (Jordan form)**.** *For any matrix $X \in \mathbb{C}^{n\times n}$, there exists a nonsingular matrix T and a matrix X_J called the Jordan form of X, such that $T^{-1}XT = X_J$ which has the block diagonal structure*

$$X_J = \begin{bmatrix} B_{11} & & \\ & \ddots & \\ & & B_{jj} \end{bmatrix}$$

where B_{ii} is a block diagonal matrix, i.e.,

$$B_{ii} = diag(J^{(i)}_{n_1}, \ldots, J^{(i)}_{n_{s_i}})$$

with only one (repeated) eigenvalue λ_i and

$$J^{(i)}_{n_k} = \lambda_i I_{n_k} + Z_{n_k} = \underbrace{\left.\begin{bmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}\right\} n_k}_{n_k} \ .$$

The block $J^{(i)}_{n_k}$ is called a Jordan block associated with λ_i.

The Jordan form of a matrix completely reveals the eigenstructure of a matrix X. Indeed, as each diagonal block B_{ii} contains only Jordan blocks associated with λ_i, the dimension of each block B_{ii} gives the algebraic multiplicity of λ_i, i.e,

$$m_a(\lambda_i) = \sum_j n_j \ .$$

Since the kernel of each Jordan block has dimension 1, the number of Jordan blocks associated with λ_i gives the geometric multiplicity of λ_i, i.e.

$$m_g(\lambda_i) = s_i \ .$$

Definition 1.25. *An eigenvalue λ_i is called simple if $m_a(\lambda_i) = m_g(\lambda_i) = 1$, or equivalently, if there is only one Jordan block associated with it and its dimension is 1.*

Definition 1.26. *An eigenvalue is called semisimple if* $m_g(\lambda_i) = m_a(\lambda_i)$*, or equivalently, if the size of the largest Jordan block associated to it is* 1*.*

Definition 1.27. *An eigenvalue* λ_i *is said to be nonderogatory if and only if there is only one Jordan block* $J^{(i)}$ *associated with it, i.e.* $m_g(\lambda_i) = 1$*. The size of this block then equals the algebraic multiplicity of the eigenvalue.*

Definition 1.28 (Similarity [50])**.** *A matrix* $B \in \mathbb{C}^{n\times n}$ *is said to be similar to a matrix* $X \in \mathbb{C}^{n\times n}$ *if there exists a nonsingular matrix* $T \in \mathbb{C}^{n\times n}$ *such that*

$$B = T^{-1}XT \ .$$

Definition 1.29. *The subset of* $\mathbb{C}^{n\times n}$ *which contains all the matrices similar to* X *is called the (similarity) orbit of* X *and is written* $\mathrm{orb}(X)$*. Equivalently, we can write*

$$\mathrm{orb}(X) = \{T^{-1}XT \ : \ T \in \mathbb{C}^{n\times n}, T \text{ is nonsingular}\} \ .$$

1.1.3 Polynomials

A polynomial $p(\lambda)$ of degree n on a field $\mathbb{F}$ is a function

$$p(\lambda) \ : \ \mathbb{C} \to \mathbb{C} \ : \ p(\lambda) = p_n\lambda^n + \cdots + p_1\lambda + p_0, \quad p_n \neq 0 \tag{1.5}$$

whose coefficients $p_i \in \mathbb{F}, \quad i = 1, \ldots n$. The vector $p^T = \begin{bmatrix} p_n & \cdots & p_0 \end{bmatrix} \in \mathbb{F}^{n+1}$ is called the vector of coefficients. The polynomial $p(\lambda)$ is said to be a complex polynomial when $\mathbb{F} = \mathbb{C}$ and to be a real polynomial when $\mathbb{F} = \mathbb{R}$. A polynomial is said to be *monic* if $p_n = 1$.

An alternative formulation to (1.5) is obtained through the use of the standard complex-valued scalar product :

$$p(\lambda) = \langle p, \pi_\lambda^n \rangle_{\mathbb{C}}$$

where π_λ^n is the vector

$$\pi_\lambda^n = \begin{bmatrix} \lambda^n \\ \vdots \\ \lambda \\ 1 \end{bmatrix} \ .$$

We say that $r_i \in \mathbb{C}$ is a root of $p(\lambda)$ if $p(r_i) = 0$.

Theorem 1.30 (Fundamental theorem of algebra)**.** *A polynomial of degree* n *has* n*, possibly non-distinct, roots.*

The roots of a real polynomial are either real or occur in complex conjugate pairs. By $\bar{p}(\lambda)$ we denote the polynomial obtained from the conjugate $\bar{p}$ of p

$$\bar{p}(\lambda) = \langle \bar{p}, \pi_\lambda^n \rangle \ .$$

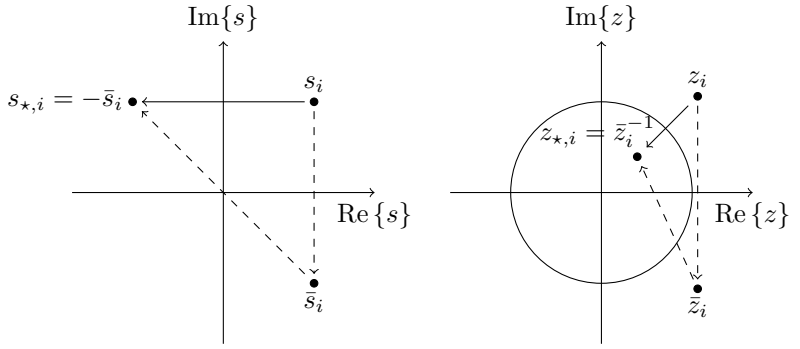


Figure 1.1: Location of the roots of a paraconjugate polynomial computed from the expressions (1.6) in continuous-time, left, and (1.7) in discrete-time, right.

The *paraconjugate polynomial* of a polynomial $p(\lambda)$ is the polynomial whose roots are conjugate with respect to a particular curve Γ in the complex plane, we have

$$p_\star(s) \stackrel{\text{def}}{=} \overline{p(-\bar{s})} = \bar{p}(-s) \qquad \text{if } \Gamma = j\mathbb{R} \tag{1.6}$$

$$p_\star(z) \stackrel{\text{def}}{=} \overline{\bar{z}^n p(1/\bar{z})} = z^n \bar{p}(1/z) \qquad \text{if } \Gamma = e^{j\mathbb{R}} \ . \tag{1.7}$$

Note that we have used s and z instead of λ to stress the connection to continuous-time systems (in the Laplace variable s) and discrete-time systems (in the delay variable z), respectively.

The location of the roots of a paraconjugate polynomial are represented in Figure 1.1.

Alternatively to the definition (1.5) of a polynomial, we can define a monic polynomial from its vector of roots. Let $r \in \mathbb{C}^n$ be the vector of n roots of $p(\lambda)$; a polynomial $p(\lambda; r)$ of degree n is a function

$$p(\lambda; r) \ : \ \mathbb{C} \times \mathbb{C}^n \to \mathbb{C} \ : \ p(\lambda; r) = \prod_{i=1}^{n} (\lambda - r_i) \ . \tag{1.8}$$

Since each coefficient of $p(\lambda; r)$ in (1.8) is a multilinear expression of all the roots contained in r, an expression for the vector of coefficients of (1.8) does not come easily thereof prompting us to denote by

$$p(r) = [p(\lambda; r)]_c$$

the vector of coefficients $p \in \mathbb{C}^{n+1}$ of the polynomial $p(\lambda; r)$.

Polynomials and matrices are linked through the characteristic polynomial. Let $X \in \mathbb{C}^{n \times n}$. Then the characteristic polynomial $x(\lambda)$ is obtained as

$$x(\lambda) = \det(\lambda I - X)$$

and the eigenvalues of X are the roots of the characteristic polynomial $x(\lambda)$.

Let

$$x(\lambda) = \lambda^n + \langle x, \pi_\lambda^{n-1} \rangle_{\mathbb{C}}$$

be a given monic polynomial of degree n. Then the matrix

$$X(x) = \begin{bmatrix} -x_{n-1} & 1 & & \\ \vdots & & \ddots & \\ \vdots & & & 1 \\ -x_0 & & & \end{bmatrix}$$

is called the companion matrix associated with x. One can also write the analytic expression of $X(x)$ which is

$$X(x) = Z - xe_1^T \ ,$$

where Z is the upper shift matrix

$$Z = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ & & & 0 \end{bmatrix} \ .$$

1.2 Analysis

1.2.1 Sets

In the following we consider a vector space E equipped with the standard inner product $\langle \cdot, \cdot \rangle$ on E and the associated induced norm $\| \cdot \|$.

We note by $\mathcal{C} \subset E$ the inclusion of $\mathcal{C}$ in E. The union of two sets $\mathcal{C}_1$ and $\mathcal{C}_2$ is $\mathcal{C}_1 \cup \mathcal{C}_2$, their intersection is written $\mathcal{C}_1 \cap \mathcal{C}_2$ and the difference of two sets is denoted $\mathcal{C}_1 \backslash \mathcal{C}_2$.

Let a set $\mathcal{C} \subset E$. We denote the interior of $\mathcal{C}$ by $\operatorname{int} \mathcal{C}$. The closure of $\mathcal{C}$ is denoted by $\operatorname{cl}(\mathcal{C})$. The boundary of $\mathcal{C}$ are the elements obtained as

$$\operatorname{cl}(\mathcal{C}) \backslash \operatorname{int} \mathcal{C} \ .$$

A set is said to be *closed* if it contains all its limit points. It is *bounded* if it contains no half line. A set which is both closed and bounded is called *compact*.

Definition 1.31 (Convex set)**.** *Consider a set $\mathcal{C} \subset E$. We say that $\mathcal{C}$ is convex if and only if for any $x, y \in \mathcal{C}$*

$$\theta x + (1 - \theta) y \in \mathcal{C} \quad \forall \theta \in [0, 1] \ .$$

A set which is not convex will be called *nonconvex*.

We also introduce the general definition of *cone* and present the cones that we will use at some point in this manuscript.

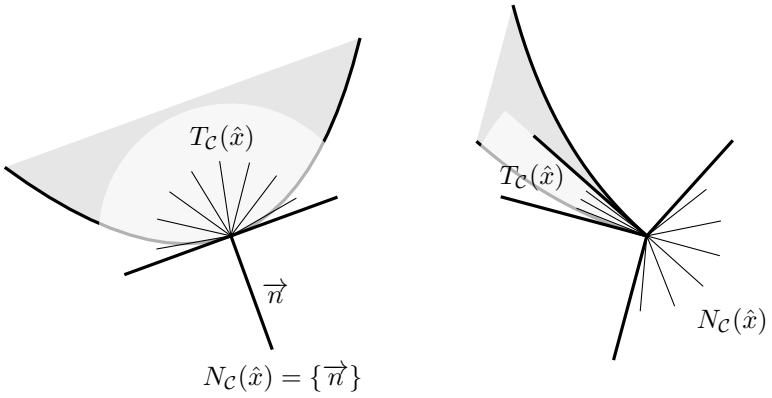


Figure 1.2: Representation of the tangent and the normal cones on two different sets.

Definition 1.32 (Cone)**.** *A set $K \subset E$ is a cone if and only if* $0 \in K$ *and* $\forall x \in K$

$$tx \in K, \quad \forall t > 0 \ .$$

An example of a cone is given by the set $\mathbb{K}^n_+$ of positive semidefinite matrices.

Definition 1.33 (Convex cone)**.** *A cone K is said to be convex if K is nonempty and*

$$\sum_{i=1}^{p} \lambda_i x_i \in K, \forall x_i \in K \text{ and } \lambda_i \geq 0, \ p \in \mathbb{N} \ .$$

Definition 1.34 (Dual cone)**.** *For any cone $K \subset E$ the dual, or* polar*, of K noted K^* is given by*

$$K^* = \{y \mid \langle y, x\rangle \leq 0, \ \forall x \in K\} \ .$$

The dual of a cone is always closed and therefore the dual of the dual, noted $(K^*)^*$ is $\mathrm{cl}(K)$. We now give three examples that will be useful to our developments.

Example 4 (Tangent and normal cone to a set [86])**.** *Let $\hat{x} \in \mathrm{cl}(\mathcal{C}) \subset E$. Then the normal and tangent cones*[1] *to $\mathcal{C}$ at $\hat{x}$ denoted by $N_{\mathcal{C}}(\hat{x})$ and $T_{\mathcal{C}}(\hat{x})$ respectively are given by*

$$N_{\mathcal{C}}(\hat{x}) = \{y \in E \ | \langle y, x - \hat{x}\rangle \leq o(\|x - \hat{x}\|), \quad \forall x \in \mathcal{C}\} \ ,$$
$$T_{\mathcal{C}}(\hat{x}) = N^*_{\mathcal{C}}(\hat{x})$$

[1]The definition of normal and tangent cones given here holds only under the assumption that the set $\mathcal{C}$ is regular in the sense of Clarke at $\hat{x}$. However, in this manuscript we will not deal with the nonregular case. The definition of Clarke regularity can be found in Definition 6.4 in [86]

where the term $o(t)$ denotes a term which tends to zero at least linearly as $t \to 0$.

The tangent and the normal cones are closed and convex. Though the formulas make sense for any point in $\mathcal{C}$, they both reduce to $\{0\}$ when $\hat{x} \in \operatorname{int} \mathcal{C}$. They can be easily represented as is shown in Figure 1.2.

Example 5 (Positive hull [86])**.** *Any set can be lifted into a cone. We give one way of performing this. Indeed, consider a set $\mathcal{C} \subset E$, then the set*

$$\operatorname{pos} \mathcal{C} = \{tx \mid x \in \mathcal{C}, t \geq 0\} \cup \{0\}$$

is called the positive hull of $\mathcal{C}$ and is clearly a cone on $E \times \mathbb{R}$.

1.2.2 Functions

We now explore a number of concepts related to functional analysis. We suppose throughout that the functions we are working with are lower-semicontinuous and are Clarke regular in the sense of [86]. These assumptions simplify our developments and avoid the necessity to address a number of cases which will not be encountered in this manuscript.

For a function $f : E \to \mathbb{R} \cup \{+\infty\}$ the domain of f, denoted $\operatorname{dom} f$ is the set

$$\operatorname{dom} f = \{x \in E \mid |f(x)| < \infty\} .$$

In order to prove the existence of the solutions to minimization problems, we will frequently use the notion of *level sets* of a function which we introduce next.

Definition 1.35 (Level sets)**.** *Let f be a real-valued function. The level sets of f is*

$$\mathcal{L}_{C_0} = \{x \in \operatorname{dom} f \mid f(x) \leq C_0,\ C_0 \text{ fixed}\} .$$

A function f on a set $\mathcal{C}$ with closed bounded level sets is said to be *closed* on $\mathcal{C}$.

Differentiability

We will frequently talk about smooth functions. The notion of smoothness is intrinsically linked to the one of differentiability and directional derivatives. Let E and F be two vector spaces.

Definition 1.36 (Directional derivative)**.** *Given a function $f : E \to F$, the directional derivative $Df(x)[h]$ of f at x in the direction $h \in E$ is given by*

$$Df(x)[h] = \lim_{t \to 0^+} \frac{f(x+th) - f(x)}{t} . \tag{1.9}$$

A function $f : E \to F$ is differentiable at $x \in E$ if there exists an approximation of the function such that

$$f(x+h) = f(x) + Df(x)[h] + o(\|h\|) \tag{1.10}$$

where the notation $o(t)$ denotes terms that are decreasing at least linearly. This can be defined more precisely using the notion of limits.

Definition 1.37 (Differentiability)**.** *A function $f : E \to F$ is said to be differentiable at $x \in \operatorname{dom} f$ if and only if $\exists Df(x)[h] \in F$*

$$\lim_{\|h\|\to 0} \frac{\|f(x+h) - f(x) - Df(x)[h]\|}{\|h\|} = 0 \ .$$

A function f is differentiable if it is differentiable everywhere on $\operatorname{dom} f$. A function f is said to be of class $C^{(k)}$ if it is k times continuously differentiable everywhere on $\operatorname{dom} f$ and the kth derivative is continuous on $\operatorname{dom} f$. A function f is said to be smooth if f is of class $C^{(2)}$ at least. When f is a differentiable real-valued scalar function, Definition 1.37 implies that $Df(x)[h]$ is linear in h and is given by

$$Df(x)[h] = \langle \nabla f(x), h \rangle \ ,$$

where ∇f denotes the gradient of f.

Instead of using Definition 1.36, one can also use the relation (1.10) and deduce the expression of the gradient from the first order approximation of the function; this is however equivalent to the use of the Definition 1.36 of directional derivative. Directional derivatives of higher order can be obtained by applying recursively the definition (1.9). For example, we have

$$D^2 f(x)[h,h] = \lim_{t\to 0^+} \frac{Df(x+th)[h] - Df(x)[h]}{t}$$

Given a real-valued scalar function f of class $C^{(3)}$, we have the following identities

$$\begin{aligned} Df(x)[h] &= \langle \nabla f(x), h \rangle \\ D^2 f(x)[h,h] &= \langle \nabla^2 f(x)[h], h \rangle \\ D^3 f(x)[h,h,h] &= \langle D^3 f(x)[h,h], h \rangle \ . \end{aligned}$$

Directional derivatives satisfy the symmetry property, i.e.

$$D^2 f(x)[h_1, h_2] = D^2 f(x)[h_2, h_1] \qquad \forall h_1, h_2 \in E \ .$$

Let us illustrate the use of directional derivatives in the computation of the gradient of a function.

Example 6. *Let $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ be given by*

$$f(X) = X^{-1} \ .$$

If we use the definition given by equation (1.9), we can compute the directional derivative of f in a given direction $H \in \mathbb{C}^{n\times n}$

$$\begin{aligned} Df(X)[H] = \lim_{t\to 0^+} \frac{f(X+tH) - f(X)}{t} &= \lim_{t\to 0^+} \frac{(X+tH)^{-1} - X^{-1}}{t} \\ &= \lim_{t\to 0^+} \frac{(X+tH)^{-1}(I - (X+tH)X^{-1})}{t} \\ &= \lim_{t\to 0^+} \frac{(X+tH)^{-1}(X - (X+tH))X^{-1}}{t} \\ &= -X^{-1}HX^{-1} \ . \end{aligned}$$

Example 7. *Let us now compute the directional derivative of the log det function* $f(X) : \mathbb{K}_+^n \to \mathbb{R} : X \to -\log\det(X)$. *Let* $H \in \mathbb{K}_+^n$ *and let us compute the first order approximation of the function from which we will deduce the gradient of the function*

$$\begin{aligned} -\log\det(X+tH) &= -\log\det(X^{\frac{1}{2}}(I+tX^{-\frac{1}{2}}HX^{-\frac{1}{2}})X^{\frac{1}{2}}) \\ &= -\log\det(X) - \log\det(I+tX^{-\frac{1}{2}}HX^{-\frac{1}{2}}) \\ &= -\log\det(X) - \sum_i \log(1+t\lambda_i(X^{-\frac{1}{2}}HX^{-\frac{1}{2}})) \end{aligned}$$

The first order approximation of $\log(1+t\lambda_i)$ *around* $t=0$ *gives* $\log(1+t\lambda_i) \approx t\lambda_i$. *Therefore,*

$$-\log\det(X+tH) = -\log\det(X) - t\sum_i \lambda_i(X^{-\frac{1}{2}}HX^{-\frac{1}{2}}) \ .$$

Following the application of the limit (1.9), we get

$$\begin{aligned} Df(X)[H] &= -\sum_i \lambda_i(X^{-\frac{1}{2}}HX^{-\frac{1}{2}}) \\ &= -\operatorname{Tr}(X^{-\frac{1}{2}}HX^{-\frac{1}{2}}) \\ &= \langle -X^{-1}, H\rangle \end{aligned}$$

where the last equality is deduced from the properties of the scalar product given in Property 1.8.

In many cases, the computation of the derivatives can be simplified by using the chain rule of the function. The frequent use of the chain rule that we make in this manuscript encourages us to develop and illustrate the concept here.

Let us consider three nonempty sets $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z} \in E$. We define the composition function

$$F : \mathcal{X} \to \mathcal{Z} : F(x) = (f \circ g)(x) \ ,$$

where

$$\begin{aligned} g &: \mathcal{X} \to \mathcal{Y} \\ f &: \mathcal{Y} \to \mathcal{Z} \end{aligned}$$

are differentiable on their domain. Then the chain rule states that

$$DF(x)[h] = Df(g(x))[Dg(x)[h]] \ . \tag{1.11}$$

Let us see two examples of its application.

Example 8. *Let us compute the directional derivative of the function*

$$F(X) = -\log\det(\mathcal{A}(X)) \ ,$$

where $\mathcal{A}(X)$ *is a quadratic operator in* X. *Thus,*

$$\begin{aligned} g(X) &= \mathcal{A}(X) \\ f(Y) &= -\log\det(Y) \end{aligned}$$

We have seen in example 7 that $\nabla f(Y) = -Y^{-1}$. *Therefore*

$$\begin{aligned} DF(X)[H] &= \langle \nabla f, Dg(X)[H] \rangle \\ &= \langle -Y^{-1}, D\mathcal{A}(X)[H] \rangle \\ &= \langle -D^*\mathcal{A}(X)[Y^{-1}], H \rangle \ . \end{aligned}$$

The first order directional derivative being always linear in H, *its adjoint is easily obtained.*

The chain rule can also be applied for higher order derivatives. Indeed, we can compute the directional derivative of the chain rule following the usual rule of derivation

$$D\left(Df(x)[Dg(x)[h]]\right)[h] = D^2 f(x)[Dg(x)[h], Dg(x)[h]] + Df(x)[D^2 g(x)[h,h]] \ .$$

A good example is obtained if we continue the preceding example.

Example 9. *Let* $F(X)$ *be given by*

$$F(X) = -\log\det(\mathcal{A}(X)) \ ,$$

as before. We have seen that the first order derivative of X^{-1} *was given by*

$$D(-X^{-1})[H] = X^{-1}HX^{-1} \ .$$

Therefore, the second order derivative of $F(X)$ *is thus*

$$D^2F(X)[H,H] = \langle X^{-1}D\mathcal{A}(X)[H]X^{-1}, D\mathcal{A}(X)[H] \rangle + \langle -X^{-1}, D^2\mathcal{A}(X)[H,H] \rangle$$

where $D^2\mathcal{A}(X)[H,H]$ *is now a quadratic function of* H.

When a function is nonsmooth at a point $\hat{x}$, the gradient ∇f at $\hat{x}$ cannot be computed. Still an alternative can be used to replace the gradient.

Definition 1.38 (Subdifferential [86])**.** *Let* $f : E \to \mathbb{R} \cup \{+\infty\}$, *the subdifferential of* f *at* $\hat{x} \in \mathrm{cl}(\mathrm{dom}\, f)$ *is the set*

$$\partial f = \{ y \in E \ | f(x) \geq f(\hat{x}) + \langle y, x - \hat{x} \rangle + o(\|x - \hat{x}\|), \forall x \in \mathrm{dom}\, f \ \} \ .$$

If f *is infinite at* $\hat{x}$, *then* $\partial f(\hat{x}) = \emptyset$.

Even when the function is differentiable, the few rules we provided until now do not always suffice to obtain an accurate expression of the gradient. As an example consider that x is a point on a manifold $\mathcal{M}$. Then by definition, the gradient $\nabla f(x)$ of the function $f : \mathcal{M} \to \mathbb{R}$ must belong to the tangent plane of $\mathcal{M}$ at x. Whenever the computed gradient does not belong to the tangent plane, it can be projected onto it to obtain the correct expression. We illustrate the underlying principles in an illustrative example obtained for the manifold of orthogonal matrices. The theory related to it is too long to be included in this manuscript and we therefore refer to [1] for the theory along with the proofs and developments associated with the example.

Example 10 (Tangent plane to the manifold of orthogonal matrices)**.** *Let the manifold $\mathcal{M} \subset \mathbb{C}^{n\times n}$ be defined by*

$$\mathcal{M} = \{U \in \mathbb{C}^{n\times n} \mid U^*U = I\}$$

A simple but not rigorous derivation of the tangent plane $\mathcal{T}_\mathcal{M}$ to $\mathcal{M}$ consists in computing the directional derivative of $U^*U = I$ in a direction S to obtain

$$\mathcal{T}_\mathcal{M} = \{S \in \mathbb{C}^{n\times n} \mid S^*U + US^* = 0\} \ . \tag{1.12}$$

Now let us suppose that we have a real-valued differentiable function whose domain is the manifold $\mathcal{M}$, i.e

$$f \ : \ \mathcal{M} \to \mathbb{R} \ : \ U \to f(U) \ .$$

The directional derivative of f at $\hat{U}$ along a direction $H \in \mathbb{C}^{n\times n}$ is given by $Df(\hat{U}) = \langle \nabla f(\hat{U}), H\rangle$ where $\nabla f(\hat{U})$ belongs to $\mathcal{T}_\mathcal{M}$. Suppose now that the computation yields a computed gradient, noted $\nabla \hat{f}(\hat{U})$, which does not belong to the tangent plane. Then, $\nabla \hat{f}(\hat{U})$ can be projected onto $\mathcal{T}_\mathcal{M}$ using the orthogonal projector $P_{\mathcal{T}_\mathcal{M}}$ defined by

$$P_{\mathcal{T}_\mathcal{M}}(\nabla \hat{f}(\hat{U})) = \nabla \hat{f} - \frac{1}{2}\hat{U}(\hat{U}^*\nabla \hat{f} + \nabla \hat{f}^*\hat{U}) \ . \tag{1.13}$$

One can thus verify easily by injecting (1.13) into (1.12) that

$$P_{\mathcal{T}_\mathcal{M}}(\nabla \hat{f}(\hat{U})) \in \mathcal{T}_\mathcal{M} \ .$$

By definition, for any direction $H \in \mathcal{T}_\mathcal{M}$ we have

$$Df(\hat{U})[H] = \langle \nabla \hat{f}(\hat{U}), H\rangle = \langle P_{\mathcal{T}_\mathcal{M}}(\nabla \hat{f}(\hat{U})), H\rangle \ .$$

Indeed, the component of $\nabla \hat{f}(\hat{U})$ orthogonal to $\mathcal{T}_\mathcal{M}$ is orthogonal to H as well.

In the same way, the theory in [1] can be used to show that $\nabla f(X)$ is symmetric if X is symmetric and diagonal if X is diagonal.

Convexity

The notion of convexity which we briefly recalled in the previous section applies to functions as well.

Definition 1.39 (Convex function)**.** *A function $f \ : \ E \to \mathbb{R} \cup \{+\infty\}$ is convex if and only if* $\operatorname{dom} f$ *is convex and for any $x, y \in \operatorname{dom} f$,*

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y), \quad \forall \theta \in [0,1] \ . \tag{1.14}$$

By definition, a convex function of class $C^{(2)}$ satisfies both $\nabla^2 f(x) \succeq 0$ and the inequality

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle \quad \forall x, y \in \operatorname{dom} f \ . \tag{1.15}$$

If $-f$ is convex, f is said to be concave.

Property 1.40 (Positive sum of convex functions [16]). *Let f_1 and f_2 be two convex functions on* $\operatorname{dom} f_1$ *and* $\operatorname{dom} f_2$ *respectively. Then for any two* $w_1, w_2 \geq 0$

$$f = w_1 f_1 + w_2 f_2$$

is convex on $\operatorname{dom} f_1 \cap \operatorname{dom} f_2$.

Property 1.41 (Pointwise maximum of convex functions [16]). *Let $f_1(x)$ and $f_2(x)$ be two convex functions on* $\operatorname{dom} f_1$ *and* $\operatorname{dom} f_2$ *respectively. Then the pointwise maximum of f_1 and f_2 defined as*

$$f(x) = \max f_1(x), f_2(x)$$

is convex on $\operatorname{dom} f = \operatorname{dom} f_1 \cap \operatorname{dom} f_2$.

Property 1.42 (Affine invariance [16]). *Let $f : E \to \mathbb{R}$ be a convex function and let $\mathcal{A}(x) + b : F \to E$ be an affine operator then*

$$f(\mathcal{A}(x) + b)$$

is convex.

Lipschitz continuity

The concept of Lipschitz continuity is very useful to estimate the local change of magnitude of a function f. It is used to bound these variations. In that prospect and in this subsection, $\|\cdot\|$ denotes any norm induced by a scalar product.

Definition 1.43 (Lipschitz continuity [86]). *A differentiable function $f(x) : E \to \mathbb{R}$ has a Lipschitz continuous gradient on a convex set $\mathcal{C}$ if there exists a positive constant L, called the Lipschitz constant, such that for any two points x and $\hat{x} \in \mathcal{C}$*

$$\|\nabla f(x) - \nabla f(\hat{x})\|_* \leq L\|x - \hat{x}\| \ , \tag{1.16}$$

A sufficient condition for a Lipschitz continuous gradient is given by the following condition

Property 1.44 ([73, 86]). *A function f of class $C^{(2)}$ has a continuous Lipschitz gradient with constant L on $\mathcal{C}$ if and only if*

$$\max_{\|h\|=1} \|\nabla^2 f(x)[h]\|_* \leq L \ .$$

As a consequence of this last property, functions with Lipschitz continuous gradient can be bounded from above as the next property explains.

Property 1.45 ([86]). *If a function f of class $C^{(1)}$ has a Lipschitz continuous gradient on the convex set $\mathcal{C}$ with constant L, then for any x and $\hat{x} \in \mathcal{C}$,*

$$f(x) \leq f(\hat{x}) + \langle \nabla f(\hat{x}), x - \hat{x} \rangle + \frac{L}{2}\|x - \hat{x}\|^2 \ .$$

Self-concordant functions

Definition 1.46 (Self-concordant function [73]). *A closed convex function $F(y)$ of class $\mathcal{C}^{(3)}$ with open domain is a self-concordant function if*

$$|D^3F(y)[h,h,h]| \leq M_F(D^2F(y)[h,h])^{\frac{3}{2}} \qquad \forall h \in E \tag{1.17}$$

holds for any $y \in \operatorname{dom} F$ and some constant $M_F \geq 0$.

If the constant $M_F = 2$ the function is called a *standard* self-concordant function.

Example 11 (Quadratic function). *Any quadratic function is self-concordant. Indeed, let $F = \langle Qx, x\rangle + \langle c, x\rangle + d$, then*

$$D^2F(x)[h,h] = \langle 2Qh, h\rangle \qquad D^3F(x)[h,h,h] = 0$$

Therefore, it is self-concordant with constant $M_F = 0$.

Example 12 (log function). *The function $F : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$*

$$F(x) = \begin{cases} -\log(x) & \text{if } x \geq 0 \\ +\infty & \text{otherwise} \end{cases}$$

is a standard self-concordant function. Indeed we have

$$F'' = \frac{1}{x^2} \qquad F''' = \frac{-2}{x^3}\ ,$$

which satisfy the inequality (1.17).

Self-concordant functions have interesting properties, notably of being affine-invariant and of being a barrier function on the closure of their domain.

Property 1.47 (Affine-invariant property [73]). *Let $\mathcal{A}(x) + b$ be an affine operator and assume that a function $F(y)$ is self-concordant with the constant M_F. Then the function $\phi(x) = F(\mathcal{A}(x) + b)$ is also self-concordant and $M_\phi = M_F$.*

Property 1.48 (Positive sum [73]). *Let F_1 and F_2 be two self-concordant functions with constants M_1 and M_2 and let $w_1, w_2 > 0$. Then the function*

$$F = w_1F_1 + w_2F_2$$

is a self-concordant function for $\operatorname{dom} F = \operatorname{dom} F_1 \cap \operatorname{dom} F_2$ with constant

$$M_F = \max\left(\frac{1}{\sqrt{w_1}}M_1, \frac{1}{\sqrt{w_2}}M_2\right).$$

Two fundamental results associated with self-concordant functions are given by the following theorems.

Theorem 1.49 ([73]). *Let the function F be self-concordant. If $\operatorname{dom} F$ contains no straight line, then the Hessian F'' of F is nonsingular at any $x \in \operatorname{dom} F$.*

Theorem 1.50 (Barrier function [73]). *Let F be a self-concordant function. Then for any point $\hat{x}$ belonging to the boundary of* $\operatorname{dom} F$ *and any sequence*

$$\{x_k\} \subset \operatorname{dom} F \ : \ x_k \to \hat{x}$$

we have $F(x_k) \to +\infty$.

Definition 1.51 (Self-concordant barrier [73]). *A standard self-concordant function $F(x)$ is a ν-self-concordant barrier for the set* $\operatorname{dom} F \subset E$, *if*

$$\max_{u \in E} [2\langle F'(x), u\rangle - \langle F''(x)[u], u\rangle] \leq \nu \ . \tag{1.18}$$

In particular if $F''(x)$ is nonsingular, (1.18) amounts to

$$\langle (F''(x))^{-1} F'(x), F'(x)\rangle \leq \nu \ .$$

The definition of self-concordant barrier is only needed to introduce the following definition which will be central in our developments.

Definition 1.52 (Analytic center [73]). *Let $F(x)$ be a ν-self-concordant barrier for the set* $\operatorname{dom} F$. *The point*

$$x_{*,F} = \arg \min_{x \in \operatorname{dom} F} F(x)$$

is called the analytic center of the convex set $\operatorname{dom} F$, *generated by the barrier $F(x)$.*

Self-concordant functions are a good class of functions on which one can apply the Newton method. Indeed, the Newton method is affine invariant and therefore it does not depend on the metric which is used. Likewise, the metric defined by the Hessian of any self-concordant function is also affine-invariant. Therefore the convergence results obtained from an application of the Newton method on a self-concordant function are very favorable as underlined in [73].

1.3 Optimality conditions

Optimization is a field of mathematics and engineering focused on the construction of algorithms and/or mathematical conditions for characterizing the solutions of a minimization problem. Because of its existent and future implications in many other fields and real-life applications, optimization is a very active field.

In the following, we understand by optimization problem any minimization problem, and accordingly, we agree on the fact that an optimization problem always is a minimization problem unless stated otherwise.

The goal is to find the minimum of a real-valued function $f(x)$ over some *feasible set* $\mathcal{C}$. As our focus will go to constrained problems we now present optimality conditions for them only. In a first step, we can differentiate between implicitly and explicitly constrained minimization problems. A minimization problem is explicitly constrained if explicit functional constraints are used to describe the feasible set. Whenever explicit functional constraints are not given, the problem is implicitly constrained.

1.3.1 Optimality conditions for implicitly constrained optimization problems

In its most basic form, an unconstrained optimization problem is written

$$\inf_{x\in\mathcal{C}} \; f(x)$$

where we suppose that $\mathcal{C} \subset E$ is nonempty and $f(x)$ is continuous on $\mathcal{C}$. The notation inf denotes the fact that the optimal solution might not be obtained, hence the search for an infimum. If additionally we suppose that $f(x)$ is closed on $\mathcal{C}$, i.e. its level sets are compact on $\mathcal{C}$, then a minimum exists and we can write

$$\min_{x\in\mathcal{C}} \; f(x) \; . \tag{1.19}$$

A minimization problem is said to be *convex* if both the objective function and the feasible set are convex. As we will see, convex optimization problems have much favorable properties compared to nonconvex ones.

A candidate solution $\hat{x}$ is a *global* optimal solution if and only if

$$f(x) \geq f(\hat{x}) \qquad \forall x \in \mathcal{C} \; .$$

Checking global optimality is very hard in general and alternative optimality conditions are sought. If we suppose that f is differentiable at $\hat{x}$, we can locally approximate $f(x)$ by its Taylor expansion

$$f(x) = f(\hat{x}) + \langle \nabla f(\hat{x}), x - \hat{x} \rangle + o(\|x - \hat{x}\|)$$

where we recall that the notation $o(t)$ denotes terms that satisfy

$$\lim_{t\to 0} \frac{o(t)}{t} = 0 \; .$$

Therefore, $\hat{x}$ is locally optimal if

$$f(x) - f(\hat{x}) = \langle \nabla f(\hat{x}), x - \hat{x} \rangle + o(\|x - \hat{x}\|) \geq 0 \; .$$

We can rewrite this last condition as

$$\langle -\nabla f(\hat{x}), x - \hat{x} \rangle \leq o(\|x - \hat{x}\|)$$

or equivalently, following our definition of the normal cone to a point (2.41)

$$-\nabla f(\hat{x}) \in N_{\mathcal{C}}(\hat{x}) \; .$$

This last result amounts to a necessary condition for local optimality and is equivalent to the following condition related to the tangent cone.

Theorem 1.53 (First order necessary local optimality conditions [86, Theorem 6.12])**.** *If a point $\hat{x} \in \mathcal{C}$ is a local optimal solution of the minimization problem (1.19) with f being differentiable at $\hat{x}$ then*

$$\langle \nabla f(\hat{x}), x - \hat{x} \rangle \geq 0 \quad \forall x \in \hat{x} + T_{\mathcal{C}}(\hat{x}) \; . \tag{1.20}$$

In particular, if $\hat{x} \in \operatorname{int} \mathcal{C}$, (1.20) implies that $\nabla f(\hat{x}) = 0$.

Any point which satisfies Theorem 1.53 is called a *stationary point*. If additionally $\mathcal{C}$ is a convex set then (1.20) is equivalent to

$$\langle \nabla f(\hat{x}), x - \hat{x} \rangle \geq 0 \qquad \forall x \in \mathcal{C} \ .$$

Let us now state a global result for convex optimization problems.

Theorem 1.54 (Necessary and sufficient first order optimality conditions [16])**.** *Let $f \in C^{(1)}$ be the objective function of a convex minimization problem. Then $\hat{x} \in \mathcal{C}$ is a global optimal solution of the minimization problem if and only if*

$$\langle \nabla f(\hat{x}), x - \hat{x} \rangle \geq 0 \qquad \forall x \in \mathcal{C} \ . \tag{1.21}$$

In particular, if $\hat{x} \in \operatorname{int} \mathcal{C}$, the condition (1.21) simplifies to

$$\nabla f(\hat{x}) = 0 \ .$$

1.3.2 Optimality conditions for explicitly constrained optimization problems

A constrained minimization problem with scalar constraints takes the form

$$\begin{array}{ll} \min\limits_{x \in E} & f_0(x) \\ s.t & h_i(x) = 0 \qquad i = 1, \ldots, k \\ & f_i(x) \leq 0 \qquad i = 1, \ldots, m \end{array} \tag{1.22}$$

where we suppose that the constraints define a nonempty set in the vector space E and $f_0(x)$ is closed on the domain $\mathcal{C}$ defined by the intersection of the equality constraints $h_i(x) = 0,\ i = 1, \ldots, k$ and inequality constraints $f_i(x) \leq 0,\ i = 1, \ldots, m$.

The problem (1.22) is called the *primal* problem in contrast to the *dual problem* which we introduce now.

In order to solve the primal problem (1.22) one can basically lift the constraints into the objective using a weighted sum of the constraints. We obtain the function

$$L(x, \mu, \lambda) = f_0(x) + \sum_{i=1}^{k} \mu_i h_i(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \ , \tag{1.23}$$

where μ and λ are called the vectors of *Lagrange multipliers* or *dual variables*. We call $L(x, \mu, \lambda)$ the Lagrangian of the problem.

If f_0 is differentiable, the *dual function* is obtained by minimizing the Lagrangian (1.23) over x to get

$$g(\mu, \lambda) = \inf_{x \in \mathcal{C}} \ L(x, \mu, \lambda) \ .$$

It is well-known that $g(\mu, \lambda)$ is concave independently of whether f_0 is convex or not. Therefore, the *dual problem* defined as

$$\begin{array}{ll} \max\limits_{\mu, \lambda} & g(\mu, \lambda) \\ s.t. & \lambda \geq 0 \end{array} \tag{1.24}$$

is a convex problem. The optimal value g_* of (1.24) satisfies the *weak duality* property

$$g_* \leq f_{0,*} \ . \tag{1.25}$$

Whenever (1.25) is active, i.e.

$$g_* = f_{0,*}$$

we say that *strong duality* holds.

Theorem 1.55 (Necessary local optimality conditions [16])**.** *Let x_* and (μ_*, λ_*) be any two primal and dual points such that strong duality holds and the gradient of the active constraints are linearly independent, then it also satisfies the Karush-Kuhn-Tucker (KKT) conditions defined as*

$$\nabla L(x_*, \mu_*, \lambda_*) = 0 \tag{1.26}$$
$$h_i(x_*) = 0 \qquad i = 1, \ldots, k \tag{1.27}$$
$$f_i(x_*) \leq 0 \qquad i = 1, \ldots, m \tag{1.28}$$
$$\lambda_{i,*} \geq 0 \qquad i = 1, \ldots, m \tag{1.29}$$
$$\lambda_{i,*} f_i(x_*) = 0 \qquad i = 1, \ldots, m \ . \tag{1.30}$$

Moreover if the primal is convex, the KKT conditions are sufficient for the existence of a local minimum at x_*.

The conditions (1.26)-(1.29) can easily be derived as the first order optimality conditions that the primal and the dual must both satisfy. Equation (1.30) is called the *complementary slackness* condition and implies that the Lagrange multipliers associated with non-active inequality constraints are zero at the optimum.

Note that Theorem 1.55 states a necessary condition for local optimality on the condition that strong duality holds but independently of whether the primal is convex or not. Still, strong duality frequently does not hold for nonconvex functions.

Theorem 1.56 (Necessary and sufficient conditions for global optimality [7])**.** *Let the primal (1.22) be convex and have a feasible set with nonempty interior. Then x_* is a global optimal solution of (1.22) if and only if there exists a dual pair (μ_*, λ_*) such that (x_*, μ_*, λ_*) satisfies the KKT conditions (1.26)-(1.30).*

1.4 Systems and Stability

1.4.1 Systems

A system is generally written as a set of differential equations

$$\begin{cases} \dot{x}(t) = f(t, x(t), u(t)) \\ y(t) = h(t, x(t), u(t)) \ , \end{cases}$$

where t is the time, $x(t)$ is the state of the system, $\dot{x}$ is the derivative of x with respect to the time, $u(t)$ is called the input of the system and $y(t)$ is called the

output of the system. In this thesis our attention goes to Linear Time Invariant (LTI) systems which have a state-space representation of the form

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \ , \end{cases} \tag{1.31}$$

where the dependence on time of x, y and u is made implicit. In (1.31), A is the state matrix. When the system is a single-input-single-output system, $x \in \mathbb{C}^n$, $y, u \in \mathbb{C}$, $A \in \mathbb{C}^{n\times n}$ $B, C^T \in \mathbb{C}^n$ and D is a scalar.

By s we denote the continuous-time differentiation operator obtained by the Laplace transform. By z we denote the discrete-time displacement operator obtained by the z-transform. When developments are common to the continuous-time and discrete-time we prefer the use of the more general notation λ which thus denotes expressions that are applicable to both s and z.

The transfer function of the system (1.31) is thus written

$$H(\lambda) = C(\lambda I - A)^{-1}B + D \ . \tag{1.32}$$

Transfer functions are rational functions of λ. The transfer function(1.32) admits a representation as a fraction of two polynomials

$$H(\lambda) = \frac{\langle b, \pi_\lambda^n \rangle_{\mathbb{C}}}{\langle a, \pi_\lambda^n \rangle_{\mathbb{C}}} \tag{1.33}$$

where $\langle a, \pi_\lambda^n \rangle_{\mathbb{C}} = \det(\lambda I - A)$ and $\langle b, \pi_\lambda^n \rangle_{\mathbb{C}} = (C\Phi B + D)$, where $\Phi/(\det(\lambda I - A)) = (\lambda I - A)^{-1}$. It should be noted that $a(\lambda) = \langle a, \pi_\lambda^n \rangle_{\mathbb{C}}$ is a monic polynomial and thus $a_n = 1$.

The roots of the numerator of (1.33) are called the *zeros* of the transfer function. The roots of the denominator of (1.33) are called the *poles* of the transfer function.

Remark 1.57. *In order to avoid unnecessary repetitions in our manuscript, we will use the system terminology for discussions common to polynomials and matrices. Therefore, by "system" one should understand the state space matrix A or its characteristic polynomial $a(\lambda)$. By "poles" of the system one should understand the corresponding eigenvalues of A or the roots of $a(\lambda)$.*

Definition 1.58 (Möbius transform)**.** *The Möbius transform is a continuous-discrete transformation defined as*

$$s \to z = \frac{1+s}{1-s} \ .$$

Moreover, the inverse transformation is given by

$$z \to s = \frac{z+1}{z-1} \ .$$

Property 1.59. *The Möbius transform is a mapping that satisfies*

$$|z| \leq 1 \text{ for } \operatorname{Re}\{s\} \leq 0 \ ,$$

and vice-versa. In particular the imaginary axis is mapped onto the unit circle.

1.4.2 Stability

Definition 1.60 (Continuous-time stability)**.** *In continuous-time a system is said to be (asymptotically)* stable *if and only if its poles are all located in the open left half-plane in* $\mathbb{C}$*.*

Definition 1.61 (Discrete-time stability)**.** *In discrete-time, a system is said to be stable if and only if all its poles are all located in the open unit disk of* $\mathbb{C}$ *centered at the origin.*

These definitions are somehow easy to check because one only needs to consider the poles of the transfer function but they imply that one is working in the spectral space. Another way of evaluating the stability of a system, this time in the coefficient space, was developed by Lyapunov.

Theorem 1.62 (Lyapunov)**.** *[65, 58] A matrix* $X \in \mathbb{C}^{n\times n}$ *is continuously (resp. discretely) stable if and only if there exists* $P \in \mathbb{K}^n_{++}$ *such that the continuous-time map*

$$\mathcal{A}_P(X) \stackrel{def}{=} \Psi_X(P) = XP + PX^*, \tag{1.34}$$

or, respectively, the discrete-time map

$$\mathcal{A}_P(X) \stackrel{def}{=} \Psi_X(P) = XPX^* - P \tag{1.35}$$

is negative definite. Moreover, if X *is stable and* $\Psi_X(P) \prec 0$*, then* $P \succ 0$*.*

We call such P a Lyapunov (or stability) certificate for X. Such a certificate is not unique, xo we can define the set $\mathbb{P}_X$ of Lyapunov certificates of a matrix X,

$$\mathbb{P}_X = \{P \in \mathbb{K}^n_{++} | -\Psi_X(P) \succ 0\} .$$

We can also give a definition for the set $\mathbb{S}^n$ of stable matrices in terms of Lyapunov operators,

$$\mathbb{S}^n = \{X \in \mathbb{C}^{n\times n} | \exists P \in \mathbb{K}^n_{++} \ : \ -\mathcal{A}_P(X) \succ 0\} .$$

Let us now cite a corollary to the last theorem.

Corollary 1.63 ([58, Theorem 11 and 12])**.** *For any continuous-time (resp. discrete-time)* $X \in \mathbb{S}^n$ *and any* $Q \in \mathbb{K}^n_{++}$*, there always exists a unique* $P \in \mathbb{K}^n_{++}$ *such that*

$$XP + PX^* = -Q ,$$

or, respectively,

$$XPX^* - P = -Q ,$$

and vice-versa.

Remark 1.64. *Note that Theorem 1.62 makes use of the definition of negative definite Hermitian matrices, which implies that $\mathcal{A}_P(X)$ has to be Hermitian. This is always guaranteed by the fact that P is Hermitian, and the whole expression makes sense.*

Remark 1.65. *The Lyapunov (1.34) and Stein (1.35) operators are linear in P but the Stein operator is quadratic in X. Nonetheless, in order to ease the notational burden, we will denote both operators by $\mathcal{A}_P(X)$ when considering X as the variable and will only distinguish the two when necessary.*

1.4.3 Other measures of stability

The Lyapunov (1.34) and Stein (1.35) inequalities are not the only available options to check the stability of a matrix. The *spectral abscissa* and the *spectral radius* provide descriptions of the stability regions in continuous- and in discrete-time respectively through simple constraints. Both are analytic spectral functions that imply the knowledge of the eigenvalues of the matrix.

Definition 1.66. *Given $X \in \mathbb{C}^{n\times n}$, the spectral abscissa is defined as*

$$\alpha(X) \stackrel{def}{=} \max_i \operatorname{Re}\{\lambda_i(X)\} \ . \tag{1.36}$$

Definition 1.67. *Given $X \in \mathbb{C}^{n\times n}$, the spectral radius is defined as*

$$\rho(X) \stackrel{def}{=} \max_i |\lambda_i(X)| \ . \tag{1.37}$$

The definitions (1.66) and (1.67) are easily adapted to polynomials. Indeed, if one uses companion matrices $X(x), x \in \mathbb{C}^n$, the definitions (1.66) and (1.67) particularize to

$$\alpha(x) \stackrel{\text{def}}{=} \max_i \operatorname{Re}\{\lambda_i(X(x))\} \ , \tag{1.38}$$

and

$$\rho(x) \stackrel{\text{def}}{=} \max_i |\lambda_i(X(x))| \ , \tag{1.39}$$

respectively, where $\lambda_i(X(x))$ now denotes the i^{th} root of the characteristic polynomial $x(\lambda) = \det(\lambda I - X)$. Equations (1.38) and (1.39) are called the *root abscissa* and *root radius* respectively.

Our definition of spectral abscissa enable us to define the concept of *active set*.

Definition 1.68. *The active set is the set of indices $\mathcal{M}$ corresponding to active eigenvalues, defined by $\mathcal{M} = \{i \ : \ \operatorname{Re}(\lambda_i(X)) = \alpha(X)\}$ in continuous-time, and by $\mathcal{M} = \{i \ : \ |\lambda_i(X)| = \rho(X)\}$.*

From our definitions, a continuous-time (resp. discrete-time) system $X \in \mathbb{C}^{n\times n}$ will be stable if and only if

$$\alpha(X) < 0 \quad \text{and} \quad \rho(X) < 1 \ ,$$

respectively.

It is well-known, [23, 21, 25, 24, 79, 97] that the spectral abscissa and the spectral radius, and their polynomial counterpart, are nonsmooth and non-differentiable functions of their arguments. Those functions indeed feature singularities at the points corresponding to multiple eigenvalues.

Whenever a single simple eigenvalue of X is active, the directional derivatives of the spectral abscissa and the spectral radius are smooth functions of their argument and, in that case, they are thus differentiable functions. When two or more simple eigenvalues are active, the spectral abscissa and the spectral radius are nonsmooth but they are nonetheless Lipschitz continuous.

Directional derivatives of the spectral abscissa and radius. Let us consider the following spectral function

$$\gamma(X) \stackrel{\text{def}}{=} \max_i \kappa(\lambda_i(X))$$

where $\kappa : \mathbb{C} \to \mathbb{R}$ is either the real part function or the modulus function and consequently $\gamma(X) = \alpha(X)$ or $\gamma(X) = \rho(X)$ depending on the instance we consider.

Let us further suppose that there is one and only one i which verifies

$$\gamma(X) = \kappa(\lambda_i(X)) \ .$$

In other words, there is only one active eigenvalue. In the following, we use the notations $\lambda_i = \lambda_i(X)$ when the context makes it possible.

Following our assumptions, the directional derivative of $\gamma(X)$ corresponds to the directional derivative of $(\kappa \circ \lambda_i)(X)$ in the direction H. In virtue of the chain rule, its expression is given by

$$D\gamma(X)[H] = \langle \tfrac{\partial \kappa(\lambda_i)}{\partial \lambda_i}, D\lambda_i(X)[H] \rangle \ , \tag{1.40}$$

where expressions for $\frac{\partial \kappa(\lambda_i)}{\partial \lambda_i}$ and $D\lambda_i(X)[H]$ are still needed. The expression of $\frac{\partial \kappa(\lambda_i)}{\partial \lambda_i}$ depends on the actual form taken by $\kappa(\lambda)$. In continuous-time we have that,

$$\frac{\partial \kappa(\lambda_i)}{\partial \lambda_i} = 1 \ . \tag{1.41}$$

If we consider the discrete-time case, then

$$\frac{\partial \kappa(\lambda_i)}{\partial \lambda_i} = \frac{\lambda_i}{|\lambda_i|} \ . \tag{1.42}$$

An expression for $D\lambda_i(X)[H]$ is in turn easily obtained from the definition of eigenvalue. Suppose v_i and u_i are the left and right eigenvectors associated with λ_i. Without loss of generality we can further assume that they are normalized such that [50, Theorem 6.3.12]

$$\|u_i\| = 1 = v_i^* u_i \ .$$

Then

$$v_i^* X u_i = \lambda_i \ ,$$

and we get

$$D\lambda_i(X)[H] = v_i^* H u_i \ . \tag{1.43}$$

When λ_i is a simple *active* eigenvalue, the equations (1.41), (1.42) and (1.43) can be plugged successively in the theoretic expression of the directional derivative (1.40) to obtain the two expressions [20]

$$D\alpha(X)[H] = \langle \nabla\alpha(X), H\rangle = \langle v_i u_i^*, H\rangle \ , \tag{1.44}$$

$$D\rho(X)[H] = \langle \nabla\rho(X), H\rangle = \langle \tfrac{\lambda_i}{|\lambda_i|} v_i u_i^*, H\rangle \ . \tag{1.45}$$

Directional derivatives of the root abscissa and radius. Directional derivatives for the root abscissa and the root radius can be obtained likewise. With the use of the same notations, a general expression for the directional derivative of $\gamma(x)$ is given by

$$D\gamma(x)[h] = \langle \tfrac{\partial\kappa(\lambda_i)}{\partial\lambda_i}, D\lambda_i(x)[h]\rangle \ . \tag{1.46}$$

The previous developments can be adopted without much change. Still one must pay attention to the space in which one is working. When working with a monic polynomial of degree n, $x(\lambda) \in \mathbb{C}^{n+1}$ but there are only n degrees of freedom such that the directional derivative is computed in a direction $h \in \mathbb{C}^n$. The problem is not encountered when working with companion matrices since then the associated polynomial is always monic.

The expressions for $\frac{\partial\kappa(\lambda_i)}{\partial\lambda_i}$ do not change. The only difference is found in the expression of $D\lambda_i(x)[h]$. Its expression can be found by writing the total variation of the polynomial $p(\lambda, \epsilon)$ obtained by perturbing $x(\lambda)$ as follows :

$$p(\lambda, \epsilon) = (x + \epsilon h)(\lambda),$$

where $h \in \mathbb{C}^n$, $\|h\| = 1$, is the direction in which we want to compute the derivative and $\epsilon > 0$ is some small scalar increment. As λ_i is a root of $x(\lambda)$, its directional derivative must satisfy

$$p(\lambda_i(\epsilon), \epsilon) = 0 \ .$$

The total variation of this last equation takes the expression for some small displacements $\delta\lambda_i$ and $\delta\epsilon$,

$$\frac{\partial p(\lambda_i(\epsilon), \epsilon)}{\partial\lambda_i}\delta\lambda_i + \frac{\partial p(\lambda_i(\epsilon), \epsilon)}{\partial\epsilon}\delta\epsilon = 0 \ ,$$

which in turn implies

$$\frac{\delta\lambda_i}{\delta\epsilon} = -\frac{\partial p(\lambda_i(\epsilon), \epsilon)}{\partial\epsilon} \Big/ \frac{\partial p(\lambda_i(\epsilon), \epsilon)}{\partial\lambda_i} \ . \tag{1.47}$$

An expression for the factors of the right-hand side in (1.47) is obtained easily. We have

$$\frac{\partial p(\lambda_i(\epsilon), \epsilon)}{\partial\epsilon} = h^T \pi_{\lambda_i}^{n-1}$$

and

$$\frac{\partial p(\lambda_i(\epsilon),\epsilon)}{\partial \lambda_i} = -\prod_{j\neq i}(\lambda_i - \lambda_j) \ .$$

Plugging these results in (1.47) and in turn in (1.46) along with the definition of adjoint yields

$$D\alpha(x)[h] = \langle \nabla\alpha(x), h\rangle = \langle \bar{\xi}_i, h\rangle \ , \tag{1.48}$$

$$D\rho(x)[h] = \langle \nabla\rho(x), h\rangle = \langle \tfrac{\lambda_i}{|\lambda_i|}\bar{\xi}_i, h\rangle \ , \tag{1.49}$$

where $\xi_i \in \mathbb{C}^n$ is defined by

$$\xi_i = \frac{\pi_{\lambda_i}^{n-1}}{\prod_{j\neq i}(\lambda_i - \lambda_j)} \ .$$

Conclusion and references

This opening chapter concisely presents the mathematical background of this thesis. We have restricted ourselves to the most important concepts and have thus discarded whatever will be useless in the following. We provided many useful examples that illustrate the theoretic content. The examples were also chosen to help the reader in the future as they detail several mathematical relations that will be used at some point in this manuscript. This preliminary chapter was constructed from several sources which we now acknowledge.

The idea of a first section which would present the most basic results on scalar products and norms comes from the related thesis of Yvan Hachez [41]. The definitions contained in the first section can be found in any introductory book of linear algebra and matrix theory, [50] is one of such.

The second section contains results from several works. The books of Boyd and Vandenberghe [16] and Rockafellar [86] were used for the introduction to convex analysis, the lectures notes of Nesterov [73] were used for the self-concordancy. Differential calculus is mostly taken from Rockafellar [86] with additions from the book of P.-A. Absil [1] which contains several elements of theory related to differential calculus.

Optimality conditions are found in most books on optimization. We used the references [16, 86, 7].

The last section on systems contains very basic statements and any introductory book on systems contains them. The expressions for the directional derivatives of the spectral abscissa and the spectral radius can be found in [19] as well.

Chapter 2

Nearest Stable System

This chapter explores the Nearest Stable System problem. Due to its intrinsic difficulty, this problem did not attract so much attention. If in some simple settings closed-form solutions can be found, they do not hold in general and heuristics are our only hope. The difficulties arise mainly from the fact that we are working in the space of coefficients of the system while stability is more easily described in the spectral space, i.e. described in the space of the eigenvalues of the matrix or roots of the polynomials. Likewise, the problem can be tackled from two viewpoints, either we see it from the point of view of the spectral space or of the coefficient space. Understanding the links between the spectrum and the coefficients and studying the geometry of the stable set will help us understand the important factors and the complexity that surrounds the problem. *In fine*, our goal is to write down optimality conditions for our problem.

This chapter is in essence a survey of the literature on the topic in which we reproduce results of interest. The common thread is to reproduce elements of theory that will be directly applicable to our case study at the end of the third section.

The chapter is organized as follows :

Section 2.1 introduces the problem of finding the nearest stable system and its particularities. It also puts into light the diversity of related problems. Finally, it addresses the question of marginal stability and indicates ways to avoid such undesirable solutions.

Section 2.2 focuses on the poles of the system, i.e. eigenvalues or roots and thus corresponds to the first viewpoint of our problem. The section is driven by the study of the effects of a perturbation on the poles. The central piece of this section is dedicated to Lidskii theory and what can be learned from it. The study of perturbations is directly linked to our problem since they predict the behavior of the poles under the action of some perturbation of the coefficients.

Section 2.3 explores the second viewpoint of our problem. The set of stable systems is recognized as nonconvex and in addition, its boundary is nonsmooth. The goal of the section is to try to answer two questions : "How does the stable set look? What can we expect from our solution?". As the questions are complex, quite often only partial answers are available. The section concludes with the introduction of a benchmark that will be used throughout this thesis to test our methods.

2.1 The Nearest Stable System problem and its particularities

Stability is a crucial property in the study of dynamical systems. We focus on the problem of enforcing the stability of a system *a posteriori*. The system can be a matrix or a polynomial either in continuous-time or in discrete-time.

Let us introduce the most general formulation of the problem which we aim to solve. Intuitively, the question is to find the smallest perturbation E that stabilizes a given unstable matrix $A \in \mathbb{F}^{n\times n}$, $\mathbb{F} = \mathbb{C}$ or $\mathbb{R}$ for some norm $\|\cdot\|$, i.e.

$$\begin{aligned} \arg \min_{E\in\mathbb{F}^{n\times n}} \quad & \|E\| \\ s.t. \quad & A + E \text{ is stable.} \end{aligned} \tag{2.1}$$

Equivalently, we can say that we are looking for the closest stable matrix $X = A + E$ to a given unstable matrix A.

The nearest stable system to an unstable one is a well-known problem deemed hard [82]. As we have just seen, it can be very simply stated and has been frequently mentioned in the literature.

From an optimization point of view, the problem is appealing for several reasons. It is a nonsmooth nonconvex optimization problem and only a limited number of algorithms are able to solve it and even fewer algorithms will be able to guarantee that the final solution will actually be stable. There is thus ample room for new methods. Also, the stable set can be expressed as a Bilinear Matrix Inequality (BMI). BMIs are an active research topic and our work could be useful to that field as well as for people working on the more frequent Linear Matrix Inequalities (LMIs). Finally, the nearest stable system problem is related to many research fields and the mathematical conditions to impose stability are actually fairly simple to write, even though the set of stable systems is not easy to describe, and have been studied extensively by the research community giving to the problem an intrinsic interest.

In addition to its intrinsic interest, this kind of problem occurs in system identification when one needs to identify a stable system from observations. Measurements being subject to truncation and noise, it may happen that the identified model is unstable while the original system is stable [33]. Our goal is then to correct those errors and obtain a suitable system to work with.

In this aspect the purpose of this technique differs fundamentally from the one of stabilizing by state or output feedback. Indeed, the purpose of stabilizing by feedback is to stabilize a system which was correctly modelled as unstable and hence the search for a proper feedback is required [6, 53, 37, 88], whereas our algorithms provide new tools that can correct measurement errors arising from the identification step by intrinsically modifying the identified system and finally obtaining a stable model.

The difficulty of the nearest stable system problem comes from the sensitivity of the eigenvalues to perturbations and in order to stabilize an unstable matrix A one needs to perturb the coefficients of A so that *all* its eigenvalues move to the stability region. In this prospect, the so-called *stability radius* problem is somehow complementary since there one looks for a perturbation

that moves only *one* eigenvalue outside the stability region. Various results have been achieved depending on the adopted characterization.

In its most general form, the stability radius problem has a formulation similar to the nearest stable system (2.1)

$$\begin{aligned} \min_{E \in \mathbb{F}^{n \times n}} \quad & \|E\| \\ s.t. \quad & X + E \quad \text{is unstable} \end{aligned} \tag{2.2}$$

where X is a stable matrix. The stability radius is used to evaluate the *robustness* of a system. The question of robustness is to determine if a system will be robust, i.e. stay stable, under the action of some perturbation.

Besides the fact that one looks for an unstable matrix is (2.2) and for a stable matrix in (2.1), what makes the difference between the stability radius problem (2.2) and the nearest stable system (2.1) is that it is the norm of the perturbation that matters for the stability radius and not the perturbation itself. On the contrary, it is the solution that matters for the nearest stable system rather than the optimal distance which is realised. Moreover, the norm of the perturbation required to find the nearest stable system solution of (2.1) is always bounded, at most, by the norm of $\|A\|$ since the zero matrix is an acceptable solution which belongs to $\mathrm{cl}(\mathbb{S}^n)$.

We can detail further the stability radius problem described by (2.2). When E is given some structure, the number of degrees of freedom decreases and computable formulas can be derived, see [48], but algorithms to efficiently compute the stability radius are only available for some special cases [15, 48, 92]. When E does not have any special structure as in (2.2) and a unitarily invariant norm is used, it can be proved using the SVD of the matrix that the optimal E is a rank-1 perturbation that will move one eigenvalue outside of the stability region, or equivalently will move one singular value to 0. This is also true for subordinate norms $\|\cdot\|_{p,q}$ where $2 \leq p, q < \infty$. For other special norms the problem is NP-hard. A generalization of the heuristics presented in this thesis in order to tackle the stability radius problem is not straightforward as Figure 2.1 suggests.

Other problems that are similar and/or complementary to the nearest stable problem can be listed. Examples of such are, for example, distance problems [46] and the structured singular value problem [80].

Some other closely related problems with stabilization have been studied as well. In [17], Burke, Henrion, Lewis and Overton stabilize fixed-order controllers using nonsmooth, nonconvex optimization. In his thesis and related papers Tom D'haene [32] stabilizes transfer functions in a given frequency band, but the optimization is carried out on the roots of the denominator whereas we concentrate on stabilizing the denominator of the transfer function with respect to its coefficients.

2.1.1 Problem setting

Let us describe the nearest stable system more precisely. The goal of this section is to introduce a more precise mathematical formulation of the problem

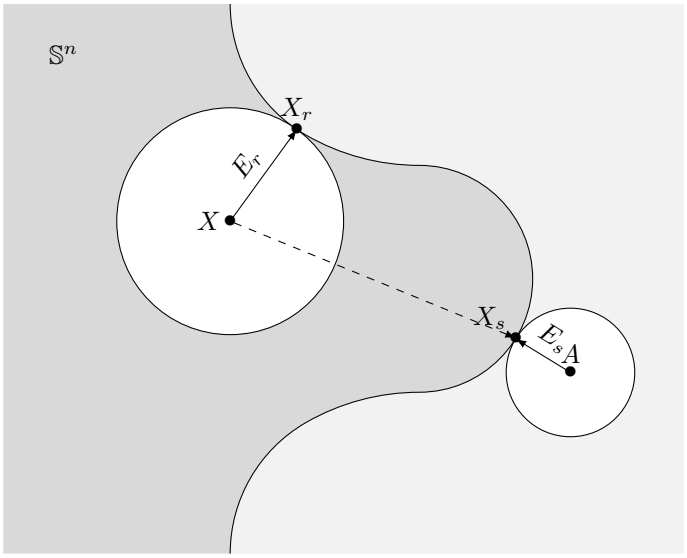


Figure 2.1: Starting from X, heuristics to solve the stability radius problem and to the nearest stable system problem must have completely different search direction since the solutions they must reach are unrelated.

---

which we will solve. Let K be a vector space on a field $\mathbb{F}$ and let $A \in K$ be an unstable system, typically A can be a square matrix of dimension n or a monic polynomial of degree n (in which case we will prefer the notation $a(\lambda)$). The nearest stable system problem can be formulated as

$$\begin{aligned} \min_{X \in K} \quad & \frac{1}{2}\|X - A\|_F^2 \\ s.t. \quad & X \in \mathrm{cl}(\mathbb{S}^n) \end{aligned} \tag{2.3}$$

where $\mathrm{cl}(\mathbb{S}^n)$ denotes the closure of the set $\mathbb{S}^n$ of stable systems. Let us make a few remarks about (2.3). First, note that we seek a solution X which belongs to the same vector space as A. This has some implications when A is real since then there is no theoretic indication that the nearest stable system X is actually also real and one might consider complex solutions. Nonetheless, if one is dealing with a real system whose state matrix A is real, it is preferable to return a real solution as well. Hence our choice makes sense. Our second remark relates to the use of a squared norm in the objective function of (2.3). This choice is also purely arbitrary and is made to ease our equations in the future, however this does not change the solution X_* which will be obtained.

Following our formulation (2.3), marginally stable solutions are thus accepted. This is necessary in order to write a minimization problem since the set $\mathbb{S}^n$ of stable systems is an open set.

Such an optimization problem arises in many contexts and with many variations depending on the models associated to X and A. For example, the

stabilization of transfer functions is widely studied [4, 32, 57] and different techniques are used. Other constraints than stability can be studied : in [30] passivity of the transfer function is enforced.

Virtually any norm could be used in replacement of the Frobenius norm $\|\cdot\|_F$ in (2.3). In this thesis we look for the nearest stable matrix in the sense of the coefficients of the matrix; the choice of the standard scalar product on $\mathbb{C}^{n\times n}$ is thus natural. The choice of the norm affects significantly the methods that are available for its resolution. In [56], Mengi derives a 2-norm characterization for this problem and is able to provide a simple 2 by 2 example of its resolution. The paper of Mengi is the only occurrence which we are aware of in which another norm than the Frobenius one was used.

One could also be interested in finding the nearest stable system in the similarity orbit $\mathrm{orb}(A)$. By definition, $\mathrm{orb}(A)$ contains all the matrices which are similar to A. It is well known, that the closure of $\mathrm{orb}(A)$ contains all the matrices which have the same spectrum as A. In that case, the problem may become simpler. If $\mathrm{Re}\{\mathrm{Tr}(A)\} \leq 0$ the distance between the stable set and the closure of the orbit is zero. In order to illustrate this consider the following example.

Example 13. *Let $A \in \mathbb{C}^{n\times n}$. Then $\mathrm{orb}(A)$ contains a matrix $\hat{A}$ which satisfies*

$$\hat{A} = \begin{bmatrix} \frac{\mathrm{Tr}(A)}{n} & \hat{a}_{12} & \dots & \hat{a}_{1n} \\ \hat{a}_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \hat{a}_{n-1n} \\ \hat{a}_{n1} & \dots & \hat{a}_{nn-1} & \frac{\mathrm{Tr}(A)}{n} \end{bmatrix} . \tag{2.4}$$

In order to see this, consider the matrix $A_0 = A - \frac{\mathrm{Tr}(A)}{n} I$. The trace of A_0 is zero and belongs to the field of values $\mathcal{F}(A_0)$, defined as (see [50])

$$\mathcal{F}(A_0) = \{x^* A_0 x, x^* x = 1\} .$$

By definition, $\mathcal{F}(A_0)$ contains the convex hull of the eigenvalues and thus

$$0 = \frac{\mathrm{Tr}(A_0)}{n} \in \mathcal{F}(A_0) .$$

Furthermore, for any unitary matrix U, $\mathcal{F}(U^* A_0 U) = \mathcal{F}(A_0)$ [50]. As $0 \in \mathcal{F}(A_0)$, there always exists a unitary matrix U such that $\hat{A} = U^* A_0 U + \frac{\mathrm{Tr}(A)}{n} I$ and is given by (2.4).

Let us consider the particular case where $\mathrm{Tr}(A) = 0$ in the following. Then let us define the similarity matrix T, for a small real parameter $\epsilon > 0$, as

$$T = \begin{bmatrix} 1 & & & \\ & \epsilon & & \\ & & \ddots & \\ & & & \epsilon^n \end{bmatrix} .$$

Now if we multiply $\hat{A}$ on the left and on the right by T and T^{-1} respectively we get a matrix $B \in \mathrm{orb}(A)$ which is written

$$B = \begin{bmatrix} 0 & \epsilon^{-1}\hat{a}_{12} & \dots & \epsilon^{-n}\hat{a}_{1n} \\ \epsilon\hat{a}_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & 0 & \epsilon^{-1}\hat{a}_{n-1n} \\ \epsilon^{n}\hat{a}_{n1} & \dots & \epsilon\hat{a}_{nn-1} & 0 \end{bmatrix} .$$

The limit for $\epsilon \to 0^+$ yields a matrix which is thus arbitrarily close to a marginally stable matrix. Therefore, when $\mathrm{Tr}(A) = 0$, the distance of $\mathrm{cl}(\mathrm{orb}(A))$ to $\mathrm{cl}(\mathbb{S}^n)$ is zero.

The result of example 13 extends to the discrete-time case.

It is also possible to consider other formulations of the problem than (2.3). For example, some authors have transformed the problem into one where one needs to determine the optimal *reflection coefficients*[1] [71, 75] which is particularly well-suited for discrete-time polynomials.

Let us now proceed to a more in-depth description of our characterization of the stable set $\mathbb{S}^n$. The set $\mathbb{S}^n$ is described by a Linear Matrix Inequality which is the Lyapunov equation in continuous-time

$$XP + PX^* \prec 0$$

or the Stein equation in discrete-time

$$XPX^* - P \prec 0$$

both of which were introduced in Chapter 1. One can note that our description of the set of stable matrices makes the set $\mathbb{S}^n$ an *open* set. Even though, (2.3) makes clear that we will work on the closure of $\mathbb{S}^n$.

One could equally describe the same stable set using other operators. In fact, most works on related problems have used, when X is a matrix, the spectral abscissa $\alpha(X)$ or the spectral radius $\rho(X)$ to enforce stability. These spectral functions are nonsmooth but a smoothed version such as the one proposed in [97] can be adopted. The problem can then be formulated as

$$\begin{aligned} \min_X \quad & \frac{1}{2}\|A - X\|^2 \\ s.t. \quad & \alpha(X) \leq 0 \ . \end{aligned}$$

The use of the spectral abscissa does not bring a clear advantage in the resolution but as its mathematical properties are well studied, the spectral abscissa is a very useful tool notably to derive some necessary local optimality conditions. Such optimality conditions will be detailed in 2.3.2.

[1] Reflection coefficients will be discussed later in the chapter on polynomials.

2.1.2 Artificially moving the stability boundary

Some further transformations could be brought to the original formulation (2.3). One could wish to reject marginally stable solutions. Marginally stable solutions generate oscillatory behaviors that can evolve into an unstable mode under the slightest perturbation of the system.

Artificially moving the stability boundary is a way of achieving pole placement and is desirable for many reasons. The pole placement topic is well discussed in the literature and originates from the desire to obtain some specific performance for a system [27, 28]. For example, significantly moving the stability boundary to the left is equivalent to enforce a minimum time constant for the system, while constraining the poles in a section of the complex plane will enable us to control overshoot among other things. Despite its intrinsic interest, the control of the dynamics that underlie the response of the stabilized system is not one of the central questions that we investigate. We nonetheless make suggestion to include this kind of specific developments in our algorithms at the end of the chapter on polynomials.

Modification of the Lyapunov and Stein operators.

We might well find out that some of the poles of the solution X_* are unstable due to numerical errors. This is made possible by a solution that is too close to the boundary of $\mathbb{S}^n$.

We could also wish to limit the number of decimals needed to represent our solution X_*. This typically happens when we want to display our solution. In this case, the truncation of some decimals amounts to a perturbation of the system and can result in instability.

All these undesirable phenomena can be avoided by keeping the solution X_* at a distance ϵ_0 away from the boundary of the stable set in the space of coefficients.

Algebraic conditions can be written for the displacement of the stability boundary. It corresponds to the continuous- and discrete-time conditions on the poles of the system

$$\mathrm{Re}\left\{\lambda_i(X)\right\} < -\epsilon_0 I \ ,$$

and

$$|\lambda_i(X)| < 1 - \epsilon_0 \quad \forall i \ ,$$

respectively, where $\epsilon_0 \geq 0$. Therefore when the system is presented as a matrix, the following proposition is useful.

Definition 2.1. *Given $\epsilon_0 \in \mathbb{R}_+$, a matrix X is said to be ϵ_0-stable if and only if $\exists P \in \mathbb{K}^n_{++}$ such that the following inequalities are satisfied in continuous-time*

$$XP + PX^* + 2\epsilon_0 P \prec 0 \ , \tag{2.5}$$

and in discrete-time

$$XPX^* - (1-\epsilon_0)^2 P \prec 0 \ . \tag{2.6}$$

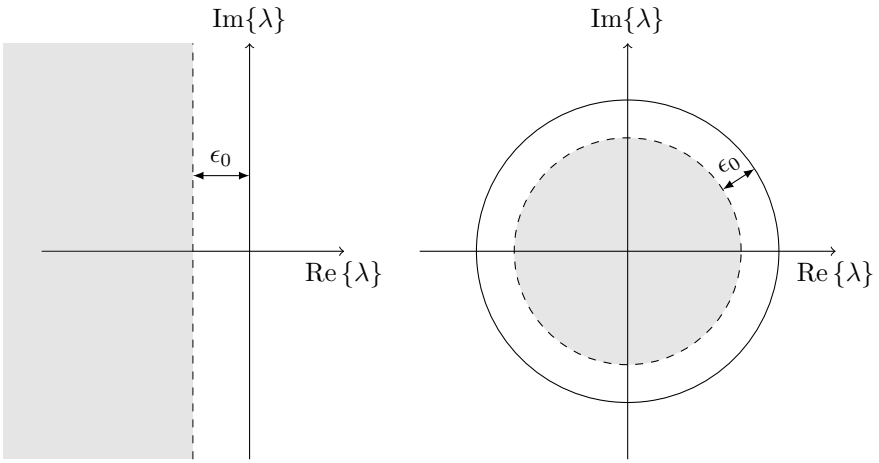


Figure 2.2: Regions of ϵ_0-stability in continuous- and discrete-time.

Equation (2.5) is obtained by shifting X by a quantity $-\epsilon_0 I$ in the original Lyapunov equation. Equation (2.6) is obtained by contracting X by a factor $1/(1-\epsilon_0)$ in the original Stein equation. The necessity and sufficiency proofs are presented in a broader context in [27].

The regions of stability are illustrated in Figure 2.2. This type of constraint was successfully implemented in our work.

Additional constraints for pole placement.

We have shown that it was possible to move artificially the stability boundary in order to dismiss marginally stable solutions. But as pointed out in his thesis by Tom D'haene [33], it can be relevant to constrain the phase of the poles of the system in an interval. As Lyapunov operators only constrain the modulus of the poles, adding this feature would require an additional constraint which would ideally be expressed as a function of the coefficients to avoid the cost of an explicit computation of the poles. As the phase $\varphi(\lambda_i)$ can be expressed as

$$\tan\varphi(\lambda_i) = -j\frac{\lambda_i - \bar{\lambda}_i}{\lambda_i + \bar{\lambda}_i} \ ,$$

we can define an additional constraint of the Lyapunov type

$$(1-\tan\alpha)XP - (1+\tan\alpha)PX^* \prec 0 \ . \tag{2.7}$$

Provided that $\alpha \in]-\frac{\pi}{2}, \frac{\pi}{2}[$ the condition (2.7) is well-defined. When $\alpha = -\frac{\pi}{2}$ or $\frac{\pi}{2}$ the condition is automatically satisfied. Two of these constraints are actually necessary to define an interval for the phase. As is shown in Figure 2.3, two distinct regions satisfy the two inequalities. In continuous-time the usual stability constraint is such that the feasible region is located in the left-half

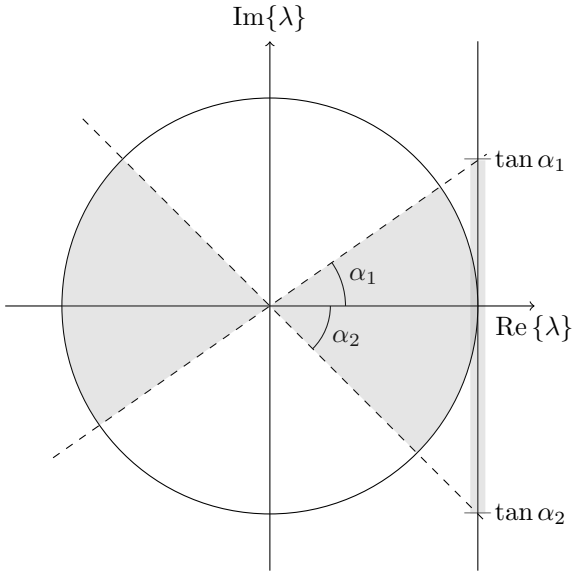


Figure 2.3: The interval of tangent values satisfying the combination of two inequalities of opposed sign of the type (2.7) for two angles α_1 and α_2 is easily pictured and describes two regions in gray inside the unit circle for the discrete-time case.

plane and thus everything goes without difficulties. Unfortunately, this is not the case in discrete-time but given that the inequality in (2.7) is strict, the two regions are separate. If a barrier function is applied on the constraint, solutions that are located in one region will not cross into the other one and the choice of the starting point will thus select the region of interest as well. Despite being tractable in theory, its inclusion in some form of efficient algorithm is not obvious since it modifies significantly the feasible set of the optimization problem. In the framework of this thesis we have worked with the usual stability constraint and did not try to add phase constraints of the type (2.7).

2.2 Perturbation theory

Authors in the literature have worked on many aspects to find bounds for delimiting the action of the perturbation, to develop analyses for determining the position of the perturbed eigenvalues and, related to this, to stabilize a matrix subject to some uncertainties.

Studying the trajectories followed by eigenvalues under the action of a perturbation is yet another way of illustrating the complexity of our problem. It underlines especially the non-linear behaviours that can occur when perturbing the coefficients of the matrix.

2.2.1 Perturbation bounds and theoretical global optimum

In some cases the global optimal solution of the nearest stable matrix can be computed analytically. This happens when A has a special structure and this special structure is purposely kept in the final solution. For example, consider the very trivial case where A and X are diagonal matrices. Then the space of eigenvalues and the space of coefficients are identical and, in the continuous-time case, the solution is achieved when A is projected on the cone of negative definite matrices in the sense of the 2-norm. In terms of eigenvalues this means that the unstable eigenvalues are projected on the imaginary axis and the stable eigenvalues are kept unchanged.

The subject of perturbation bound on the eigenvalues of matrices is quite well developed in the literature [9, 94]. We reproduce some of the results of Bhatia [9] that are linked to our problem. Let D_A and D_X be diagonal matrices of the eigenvalues of A and X. Then we define the distance between the eigenvalues of A and X as

$$d(D_A, D_X) = \min_{W \in \Pi} \|D_A - W D_X W^{-1}\|_F \ , \tag{2.8}$$

where Π is the set of permutation matrices.

Theorem 2.2 (Wielandt-Hoffman Theorem [9, 46, 49])**.** *Let A and X be two normal matrices that belong to $\mathbb{C}^{n\times n}$. Then*

$$d(D_A, D_X) \leq \|A - X\|_F \ . \tag{2.9}$$

Let us use an example to show the importance of this theorem.

Example 14 (Nearest stable normal matrix to an unstable normal matrix)**.** *Let the conditions of Theorem 2.2 be fulfilled. As both A and X are normal, we can decompose them as*

$$A = V D_A V^* \quad X = U D_X U^* \ .$$

While this decomposition is given for A, the choice of transformation U for X is free. We want to show that the nearest stable normal matrix X_* to A is also the solution of

$$\min_{X \in \mathbb{S}^n} d(D_A, D_X) \ .$$

In order to prove this, let us suppose that $\hat{X}_*$ is a solution for which the distance to A in the space of eigenvalues is optimal

$$d(A, \hat{X}_*) \leq d(A, X) \qquad \forall X \in \mathbb{S}^n, X \text{ normal} \ . \tag{2.10}$$

It is clear that $\hat{X}_*$ is not unique; any matrix $X_{**} = R\hat{X}_* R^*$ such that $RR^* = I$, $R \in \mathbb{C}^{n\times n}$ is a unitary matrix also satisfies (2.10). Therefore, there exists a unitary matrix U such that $X_* = U\hat{X}_* U^*$ and realises the equality $\|A - X_*\| = d(A, X_*)$. And we can easily conclude

$$\|A - X_*\| \leq d(A, X) \leq \|A - X\| \qquad \forall X \in \mathbb{S}^n, X \text{ normal} \ .$$

Thus in continuous-time, the nearest stable normal matrix X_* to A is the projection of A on the cone of the negative definite normal matrices equipped with the 2-norm. As all unitary, Hermitian and skew-Hermitian matrices are normal, inequality (2.9) also holds for them.

In our case, we seek to find the nearest stable matrix to A without keeping its particular structure and we cannot therefore determine beforehand where the eigenvalues of the global solution will lie. Still we can find bounds on the minimal distance that can be achieved. For a matrix X with no particular structure, we have in case A is normal and X is general [94]

$$d(D_A, D_X) \leq c\|A - X\|_F \ , \tag{2.11}$$

and $c = \sqrt{n}$ is the smallest possible constant. In case A is Hermitian, the constant c improves to $\sqrt{2}$. The fact that c is greater than 1 indicates that the distance between the eigenvalues X and the eigenvalues of A can be greater than the norm of the difference $A - X$. As expected, the bounds are even trickier when either of the two matrices A and X are general. In this case, the book of Bhatia [9] provides us with the following bound

$$d(D_A, D_X) \leq n(2M)^{1-\frac{1}{n}}\|A - X\|_F^{\frac{1}{n}} \ , \tag{2.12}$$

where $M = \max(\|A\|_F, \|X\|_F)$. Let us illustrate the consequences of these inequalities on our problem.

Example 15. *Consider a discrete-time system whose dynamic is represented by the following matrix of size n*

$$A = \begin{bmatrix} & & & b \\ a & & & \\ & \ddots & & \\ & & a & \end{bmatrix} .$$

where $a, b \in \mathbb{C}$. The eigenvalues of A are distributed evenly on a circle centered at the origin. We suppose that $|a|$ and $|b|$ are greater than one, making A unstable. We would like to determine the smallest perturbation Δb of b such that X is discretely stable. If $|b| = |a|$, then A is a normal matrix and the closest stable normal matrix is given by

$$X_* = \begin{bmatrix} & & & \frac{b}{|b|} \\ \frac{a}{|a|} & & & \\ & \ddots & & \\ & & \frac{a}{|a|} & \end{bmatrix} .$$

and the objective function equals

$$\|A - X_*\|_F = (|b| - 1)\sqrt{n} \ . \tag{2.13}$$

Now if the normality of X is not required, the smallest perturbation Δb of b such that the system becomes stable must satisfy $|b + \Delta b||a|^{n-1} = 1$. The solution Δb is such that

$$b + \Delta b = \frac{e^{j\varphi}}{|a|^{n-1}}$$

where the angle φ can be chosen to achieve the equality

$$|\Delta b| = |b| - \frac{1}{|a|^{n-1}} \ .$$

In other words, we have

$$\|A - X_*\|_F = |b| - \frac{1}{|a|^{n-1}} \ . \tag{2.14}$$

The comparison of (2.13) and (2.14) shows that the value of the objective function can be much smaller if the structure does not need to be maintained.

2.2.2 Perturbation of eigenvalues

Consider a matrix $X \in \mathbb{C}^{n\times n}$ whose eigenvalues are not in any specific region of the complex plane and thus X can either be stable, marginally stable or unstable. Also suppose that X is given as an analytic function of a vector of parameters p and let us set an initial vector of parameters $p_0 \in \mathbb{C}^m$, and a corresponding matrix $X_0 = X(p_0)$. We will analyze how the eigenvalues of X behave under the action of a perturbation Δp. The question is not new and some authors have brought answers to this meaningful question. The main result that we expose here was obtained by Lidskii [64] back in the 60's, and further reported by Moro et al [70]. Other authors contributed to the topic and [89, 98, 60, 61, 54] are just a few references.

Since we are working with analytic perturbations of X, it can be shown [54] that the perturbed matrix $X(p + \Delta p)$ with $\Delta p = \epsilon\hat{e}, \hat{e} \in \mathbb{C}^m$, $\|\hat{e}\| = 1, \epsilon \in \mathbb{R}$, admits a Taylor series given by

$$X(p_0 + \Delta p) = X_0 + \epsilon X_1 + \epsilon^2 X_2 + \cdots \ , \tag{2.15}$$

where the matrices $X_1, X_2, \ldots$ are obtained as a combination of partial derivatives $X(p)$, i.e. an algebraic computation gives

$$X_1 = \sum_{i=1}^{m} \left.\frac{\partial X(p)}{\partial p_i}\right|_{p=p_0} \hat{e}_i \ . \tag{2.16}$$

Series expansions for the eigenvalues and the left and right eigenvectors can also be computed. Each eigenvalues admits a so-called *Puiseux series*[83, 84], i.e. a series in fractional powers of ϵ, which is given, for an original unperturbed eigenvalue λ_0 of X_0, by

$$\lambda(\epsilon) = \lambda_0 + \sum_{i=1}^{\infty} \epsilon^{\beta_i}\mu_i \ , \tag{2.17}$$

where β_i is a fractional power.

Our interest goes to the leading term of this expansion since the behavior of the eigenvalue $\lambda(\epsilon)$ is predominantly defined by the lower order term as $\epsilon \to 0$.

The computation of the coefficients β_i and μ_i is most often cumbersome. It requires the full knowledge of the Jordan structure of the matrix and/or the knowledge of the left and right eigenvectors associated to the eigenvalue which is analysed. The analysis of the general treatment that applies to any given matrix $X \in \mathbb{C}^{n\times n}$ goes well beyond the scope of this thesis. We thus follow our line of work by developing the elements of theory we think are either necessary for our future developments or give an insight into the problem. Our goal is essentially to bring forward the relation between the dynamics followed by the perturbed eigenvalues and the structure of the matrix.

Two works will be at the center of our attention. The first one is the article of Moro, Burke and Overton [70] where the authors show how the so-called *Newton diagram* can be conveniently used to determine all the powers β_i in the expansion (2.17). The theory we present is intentionally simplified compared to the available content of [70] since only the presented elements will be used later on.

The second work central to our developments is the one of Seyranian and Mailybaev [89] where they illustrate the principles of interactions for the special case of a double eigenvalue λ_0 perturbed by a simple parameter of varying magnitude. Our intention here is not to redo the work from those authors but rather to gather at a single location some of the key results they obtained in their papers with a special focus on those that will be useful to us.

Newton diagram for determining the essential behavior of the perturbed eigenvalue

Let $X(p)$ be in Jordan form. Then for a given perturbation $p_\epsilon = p_0 + \epsilon\hat{e}$, the eigenvalues of $X(p_\epsilon)$ can be computed using the characteristic polynomial of $X(p_\epsilon)$

$$c(\lambda, \epsilon) = \det(\lambda I - X(p_\epsilon))$$

Without loss of generality suppose that X_0 only has one eigenvalue λ_0 which is 0, otherwise we can replace λ by $\lambda - \lambda_0$ in the characteristic polynomial and repeat the process for each eigenvalue. We then have

$$c(\lambda, \epsilon) = \det(\lambda I - X(p_\epsilon)) = \lambda^n + \alpha_1(\epsilon)\lambda^{n-1} + \cdots + \alpha_{n-1}(\epsilon)\lambda + \alpha_n(\epsilon) = 0\,, \quad (2.18)$$

where each coefficient $\alpha_k(\epsilon)$ is a polynomial

$$\alpha_k(\epsilon) = \hat{\alpha}_k \epsilon^{a_k} + \cdots\,, \quad (2.19)$$

where $a_k \in \mathbb{N}$. Following (2.19), $\alpha_k(\epsilon)$ is an analytic series convergent in a neighborhood of 0 which satisfies the identity

$$\alpha_k(0) = 0\,.$$

Therefore following [54] the expression $\lambda(\epsilon)$ for the roots of $c(\lambda, \epsilon)$ can be written as a convergent Puiseux series in ϵ but we are mainly interested in the leading

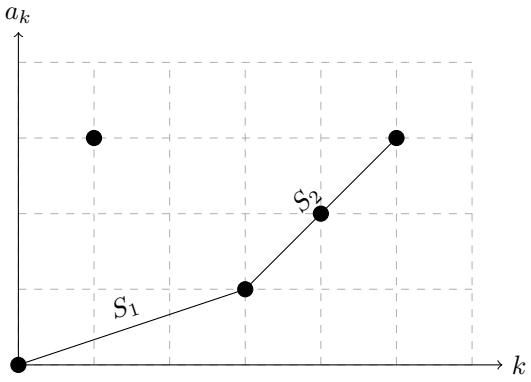


Figure 2.4: Newton diagram of $c(\lambda,\epsilon) = \lambda^5+(3\epsilon^3)\lambda^4+(\epsilon+2\epsilon^2)\lambda^2-(\epsilon^2-\epsilon^3)\lambda+\epsilon^3$

term of $\lambda(\epsilon)$, i.e.

$$\lambda(\epsilon) = \mu\epsilon^{\beta} \ . \tag{2.20}$$

The leading coefficient μ and exponent β can be found using a *Newton diagram*. Newton diagrams are found under various names [22, 59, 60] and are usually described as in [29] but we follow here the more handy description from [70]. The abscissa axis will be the index k of the coefficients $\alpha_k(\epsilon)$, and the ordinate axis will feature the smallest exponent a_k of ϵ in (2.19). If $\alpha_k(\epsilon)$ is identically zero, we take $a_k = \infty$.

It can be proved [70, 22, 60] that the number of roots of a certain power equals the length of the projection for each segment on the x-axis. Moreover, the leading power β of (2.20) is given by the slopes of each segment of the diagram and the leading coefficient μ is found as a non-trivial solution to

$$\sum_{(k,a_k)\in S_i} \mu^{n-k}\hat{\alpha}_k = 0 \ . \tag{2.21}$$

Example 16. *Consider the characteristic polynomial*

$$c(\lambda,\epsilon) = \lambda^5 + (3\epsilon^3)\lambda^4 + (\epsilon+2\epsilon^2)\lambda^2 - (\epsilon^2-\epsilon^3)\lambda + \epsilon^3 \ ,$$

whose Newton diagram is pictured in Figure 2.4. *The first segment S_1 yields the following result. Its projected length on the $k-$axis is 3, thus there are 3 roots with an expansion in fractional power $\lambda_i(\epsilon) = \mu_i\epsilon^{\frac{1}{3}}$, $i = 1,2,3$. We can determine the value of μ from the expression of $c(\lambda,\epsilon)$. The nodes on the segment S_1 correspond to the terms λ^5 and $\epsilon\lambda^2$ of $c(\lambda,\epsilon)$. Therefore, the leading coefficients μ_i are the three complex nonzero roots of the equation*

$$\mu^5 + \mu^2 = 0 \ .$$

We deduce that three out of the five roots of $c(\lambda,\epsilon)$ are given by

$$\lambda_i(\epsilon) = e^{j\frac{2i\pi}{3}}\epsilon^{\frac{1}{3}}, \ i = 1,2,3 \ .$$

The same reasoning applies to the line segment S_2 and we find two roots of power $\frac{1}{2}$. The terms of $c(\lambda,\epsilon)$ that are associated to a node on S_2 are $\epsilon\lambda^2$, $-\epsilon^2\lambda$ and ϵ^3 which implies that the leading coefficients satisfies

$$\mu^2 - \mu + 1 = 0 \ .$$

Therefore,

$$\lambda_4(\epsilon) = \frac{1+j\sqrt{3}}{2}\epsilon^{\frac{1}{2}}$$
$$\lambda_5(\epsilon) = \frac{1-j\sqrt{3}}{2}\epsilon^{\frac{1}{2}} \ .$$

The Newton diagram thus provides an easy way to determine the behavior of the eigenvalues of $X(p_0)$ under the action of a *given* perturbation. Whenever the structure of the perturbation is not available, estimations of the orders of magnitude that might occur for the leading terms of the Puiseux expansions are needed.

The question can be reformulated as determining all the possible Newton diagrams that can be obtained knowing the structure of X. Rather than providing the full developments of the theory in [70], we will limit ourselves to an insight on a particular case that we will use later on. In that perspective, suppose we have the following additional assumptions : X depends linearly on p, all the eigenvalues of X are zero and the Jordan form J of X is (logically) known. Then

$$\det(\lambda I - X(p_\epsilon)) = \det(\lambda I - J - \epsilon B) \ ,$$

for any perturbation matrix B of compatible size. If all eigenvalues of X are semisimple , meaning that the Jordan blocks associated to the eigenvalues are all of size ones, then J contains only diagonal elements and the characteristic polynomial of $J + \epsilon B$ is

$$c(\lambda,\epsilon) = \det(\lambda I - \epsilon B) = \lambda^n + \hat{\alpha}_1\epsilon^{n-1}\lambda^{n-1} + \hat{\alpha}_2\epsilon^{n-2}\lambda^{n-2} + \cdots \ . \qquad (2.22)$$

The coefficients of the characteristic polynomial have expressions of the form

$$\alpha_k(\epsilon) = \mathcal{O}(\epsilon^{n-k}) \ .$$

This is due to the structure of J and is thus verified independently of the structure of B. Since the coefficients α_k are all proportional to ϵ^{n-k}, the Newton diagram associated with (2.22) has only one segment of slope 1 which contains all the nodes of the diagram.

This means that all the roots admits a first order Taylor expansion of the type

$$\lambda_i(\epsilon) = \mu_i\epsilon \ .$$

This nice behavior does not happens if the Jordan blocks associated to X are not of size 1. Indeed, let us now see what happens if $J \in \mathbb{C}^{n\times n}$ is nonderogatory, i.e., when there is only one Jordan block associated to each

eigenvalue and let us consider as before that we have only one eigenvalue which is 0. Then,

$$J = \begin{bmatrix} 0 & 1 & & \\ & \ddots & & \\ & & & 1 \\ & & & 0 \end{bmatrix} ,$$

and the characteristic polynomial $c(\lambda, \epsilon) = \det(\lambda I - J - \epsilon B)$ still is

$$c(\lambda, \epsilon) = \lambda^n + \hat{\alpha}_1 \epsilon^{n-1} \lambda^{n-1} + \hat{\alpha}_2 \epsilon^{n-2} \lambda^{n-2} + \cdots .$$

The difference with the semisimple case occurs at the level of coefficients $\alpha_k(\epsilon)$ which now feature powers of ϵ that are lower than $n - k$, i.e.

$$\alpha_k(\epsilon) = \mathcal{O}(\epsilon^q) \qquad q < n - k .$$

Therefore the Newton diagram associated to the characteristic polynomial contains lines that have slopes typically smaller than 1. The consequence is that the eigenvalues of X now admits an expansion in fractional powers. This means that the eigenvalues are much more sensitive to perturbation than in the semisimple case. We now give the most simple example possible to illustrate this behavior.

Example 17. *Let X be a nonderogatory Jordan form of size n with 0 with an unique 0 eigenvalue*

$$X = \begin{bmatrix} 0 & 1 & & \\ & \ddots & & \\ & & & 1 \\ & & & 0 \end{bmatrix}$$

The goal is to estimate the worst possible perturbation of X. Therefore consider the perturbation matrix ϵB

$$\epsilon B = \begin{bmatrix} & & \\ & & \\ \epsilon & & \end{bmatrix} .$$

The characteristic polynomial $c(\lambda, \epsilon) = \det(\lambda I - J - \epsilon B)$ writes

$$c(\lambda, \epsilon) = \lambda^n + \epsilon .$$

The Newton diagram associated to it is represented in Figure 2.5 *and shows that the segment S has a slope $1/n$ which means that the behavior of the perturbed eigenvalues of X are governed by the law*

$$\lambda_i(\epsilon) = e^{j\frac{2i\pi}{n}} \epsilon^{\frac{1}{n}} \quad i = 1, \ldots, n.$$

This example illustrates the nontrivial role played by the block structure of the Jordan form of X in the behavior of the perturbed eigenvalues.

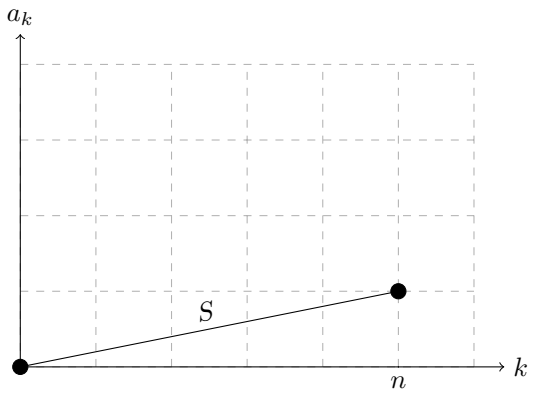


Figure 2.5: Newton diagram for the characteristic polynomial $\lambda^n + \epsilon$

Analytic expressions of the coefficients in the expansion

When the left and right eigenvectors of the eigenvalue λ_0 are known and are given by u_0 and v_0 respectively, we can compute an expression for the μ_i's in the expansion (2.17) which we rewrite here

$$\lambda(\epsilon) = \lambda_0 + \sum_{i=1}^{\infty} \mu_i \epsilon^{\beta_i}.$$

Indeed, both eigenvectors admit an expansion in fractional powers of ϵ and closed form expressions of μ_i can be obtained.

Let us state two examples that we will reuse later on.

Example 18 (Expansion of a simple eigenvalue)**.** *Let us consider λ_0 to be a simple eigenvalue of a matrix $X_0 = X(p_0) \in \mathbb{R}^{n\times n}$, $p_0 \in \mathbb{R}^m$ with associated left and right eigenvectors v_0 and u_0 in $\mathbb{R}^n$. Under a perturbation of $X(p_0 + \epsilon\hat{e})$ where $\hat{e} \in \mathbb{R}^m$ is normalized, the Taylor series of $X(\epsilon)$ is given by*

$$X(\epsilon) = X_0 + \epsilon X_1 + o(\epsilon) \ .$$

The Taylor series of the eigenvalues and the eigenvectors are

$$\begin{aligned} \lambda(\epsilon) &= \lambda_0 + \epsilon\mu_1 + o(\epsilon) \ , \\ u(\epsilon) &= u_0 + \epsilon w_1 + o(\epsilon) \\ v(\epsilon) &= v_0 + \epsilon y_1 + o(\epsilon) \ , \end{aligned}$$

We can determine the expressions of μ_1 by using the definition of eigenvalue

$$Xu = \lambda u \qquad v^T X = \lambda v^T \ .$$

Therefore we get in the one hand

$$v^T X u = (v_0^T X_0 u_0) + (v_0^T X_0 w_1 + y_1 X_0 u_0 + v_0^T X_1 u_0)\epsilon + o(\epsilon) \ ,$$

and in the second hand

$$\lambda v^T u = (v_0^T \lambda_0 u_0) + (v_0^T \lambda_0 w_1 + y_1 \lambda_0 u_0 + v_0^T \mu_1 u_0)\epsilon + o(\epsilon) \ .$$

Identifying terms of equal power of ϵ, we get the only non-trivial relation

$$\mu_1 = \frac{v_0^T X_1 u_0}{v_0^T u_0} \ . \tag{2.23}$$

Example 19 (Expansion of a double nonderogatory eigenvalue)**.** *As in the preceding example consider an eigenvalue $\lambda_0 \in \mathbb{R}$ of a matrix $X_0 = X(p_0) \in \mathbb{R}^{n\times n}$ except that λ_0 now is a nonderogatory eigenvalue of multiplicity 2. In this case, there are two associated generalized right-eigenvectors u_0 and u_1 in $\mathbb{R}^n$ that satisfy*

$$\begin{aligned} X_0 u_0 &= \lambda_0 u_0 & v_0^T X_0 &= \lambda_0 v_0^T \\ X_0 u_1 &= \lambda_0 u_1 + u_0 & v_1^T X_0 &= \lambda_0 v_1^T + v_0^T \end{aligned}$$

As in [89], we impose a unique choice of eigenvectors by making the additional hypothesis that

$$v_0^T u_0 = 0 \qquad v_0^T u_1 = 1 \qquad v_1^T u_1 = 0 \ .$$

As λ_0 is now a double nonderogatory eigenvalue, it admits a Puiseux series in ϵ

$$\lambda(\epsilon) = \lambda_0 + \epsilon^{\frac{1}{2}} \mu_1 + \epsilon \mu_2 + o(\epsilon) \ .$$

Likewise, its associated eigenvectors admit Puiseux series with the same power of ϵ and following the same reasoning as in the previous example, expressions for μ_1 and μ_2 can be deduced. They write [89]

$$\mu_1 = \pm \left(v_0^T X_1 u_0\right)^{\frac{1}{2}} \quad \mu_2 = \frac{\left(v_0^T X_1 u_1 + v_1^T X_1 u_0\right)}{2} \ . \tag{2.24}$$

Interactions of eigenvalues

The paper of Moro et al. [70] that we have discussed so far looked at how it was possible to find the leading term of the equations that drive the behavior of the perturbed eigenvalues. In particular, it identifies a simple technique to visualize the possible behavior of the perturbed eigenvalues based on the Jordan structure of the original unperturbed matrix X_0.

The central question which we now try to answer is to determine what happens if a single element of the perturbation Δp_i of $\Delta p = \epsilon \hat{e}$ is modified in magnitude. In his article Seyranian [89] tackles this question using the theory of Lidskii, Vishik and Luysternik [64, 98] and some of his own works.

The work of Seyranian comes as a complement to the work developed by Moro et al in [70] as it looks into the question of how the position of the perturbed eigenvalues evolves along with the magnitude of the perturbation.

Let us consider a real matrix $X(p) \in \mathbb{R}^{n\times n}$ which is a function of a vector $p \in \mathbb{R}^m$ with $m < n^2$ and a perturbation vector depending on $\epsilon \in \mathbb{R}$ following the equality

$$\Delta p = \epsilon \hat{e} \tag{2.25}$$

where $\|\hat{e}\| = 1$, and $\epsilon \in \mathbb{R}$, ϵ small.

Let us further suppose that $X(p_0)$ admits an eigenvalue λ_0 of algebraic multiplicity 2.

Therefore, following the work presented just before, the eigenvalue expansion can be of two types, namely λ_0 is either a semisimple eigenvalue, i.e. it has two associated Jordan blocks of size one, or a nonderogatory eigenvalue in which case it has only one Jordan block of size 2 associated with it.

When λ_0 is a double nonderogatory eigenvalue we have that

$$\lambda(\epsilon) = \lambda_0 + \mu_1 \epsilon^{\frac{1}{2}} + \mu_2 \epsilon + o(\epsilon) \ .$$

We can decompose μ_1 and μ_2 into

$$\mu_1 = \sum_{i=1}^{m} \mu_{1,i} \hat{e}_i \ , \qquad \mu_2 = \sum_{i=1}^{m} \mu_{2,i} \hat{e}_i \ ,$$

where $\mu_{1,i}$ and $\mu_{2,i}$ are given by (2.24) and (2.16)

$$\mu_{1,i} = \left(v_0^T \frac{\partial X(p)}{\partial p_i} u_0 \right) \Big|_{p_i = p_{0,i}} \quad \mu_{2,i} = \frac{1}{2} \left(v_0^T \frac{\partial X(p)}{\partial p_i} u_1 + v_1^T \frac{\partial X(p)}{\partial p_i} u_0 \right) \Big|_{p_i = p_{0,i}} \ ,$$

where we recall that v_0, v_1, u_0 and u_1 are the original left and right generalized eigenvectors associated with λ_0 We have thus

$$\lambda(\epsilon) = \lambda_0 \pm (\sum^{m} \mu_{1,i} \hat{e}_i)^{\frac{1}{2}} \epsilon^{\frac{1}{2}} + (\sum^{m} \mu_{2,i} \hat{e}_i) \epsilon_i + o(\epsilon) \ .$$

Finally, using the notation $\Delta p_i = \epsilon \hat{e}_i$ we write

$$\lambda(\epsilon) = \lambda_0 \pm (\sum^{m} \mu_{1,i} \Delta p_i)^{\frac{1}{2}} + (\sum^{m} \mu_{2,i} \Delta p_i) + o(\epsilon) \ . \tag{2.26}$$

In case λ_0 is semisimple, there are two independent left eigenvectors v_1 and v_2 and two independent right eigenvectors u_1 and u_2 associated with λ_0. As λ_0 is semisimple, the eigenvalue expansion is

$$\lambda(\epsilon) = \lambda_0 + \sum \mu_{1,i} \Delta p_i \tag{2.27}$$

where the expression for $\mu_{1,i}$ is dependent on the two pairs of eigenvectors associated with λ_0. We do not elaborate on the expression of μ_1 for a semisimple λ_0 here, see [89] for more details.

In both cases, given that we fix all elements of the perturbation but one, say Δp_1, both expansions can be put into the form

$$\lambda(\Delta p_1) = \lambda_0 + r(\Delta p_1) + jk(\Delta p_1) + o(\Delta p_1) \ , \tag{2.28}$$

where $r(\Delta p_1)$ and $k(\Delta p_1)$ are functions $\mathbb{R} \to \mathbb{R}$. By establishing a one-to-one correspondence between (2.28) and (2.27) or (2.26), a pattern for the behavior of the eigenvalues emerges.

Indeed, once the functions $r(\Delta p_1)$ and $k(\Delta p_1)$ are known, one can plot in the complex plane the followed by $\lambda(\Delta p_1)$. As we consider an eigenvalue λ_0

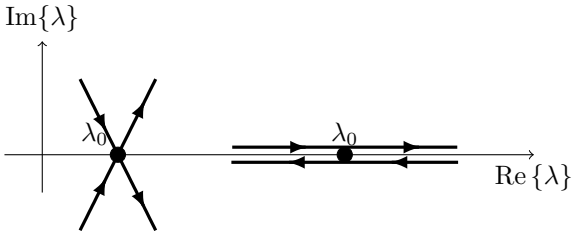


Figure 2.6: Weak interactions imply that the perturbed eigenvalues crosses without interacting when only one perturbation parameter is nonzero. Hence there are two cases : complex conjugate (left) or real (right). [89]

---

of multiplicity 2, two different paths are each time given by (2.28) and those paths intersect at a value $\Delta p_1 = 0$ and thus we can say that the eigenvalues interact.

The developments of Seyranian [89] put into light two types of interaction, *strong* and *weak*

- Strong interactions occur when λ_0 is nonderogatory. Indeed, the two eigenvectors associated with λ_0 coalesce as $\Delta p_1 \to 0$,
- Weak interactions occur when λ_0 is semisimple and thus it has two linearly independent eigenvectors.

Strong and weak interactions each have their particularities. Strong interactions are due to nondifferentiable expansions at $\Delta p_1 = 0$ and therefore the speed is infinite at that point. Weak interactions are characterized by a noninteraction when the elements Δp_i $i = 2, \ldots, n$ of the perturbation are zero, in which case the trajectory displayed by each perturbed eigenvalue is a straight line as shown in Figure 2.6. Still, when the elements Δp_i $i = 2, \ldots, n$ are fixed but nonzero, more complicated trajectories can appear, see [89] for examples.

Despite an easy conceptualization, the computation leading to the specific patterns of interaction are cumbersome. We give the most simple example of this by continuing the computations that we started at the beginning of this section for a nonderogatory eigenvalue λ_0 with multiplicity 2.

Example 20 (Strong interactions of a nonderogatory eigenvalue [89])**.** *Following (2.26) and (2.28), we have*

$$r + jk = \pm \left(\mu_{1,1}\Delta p_1 + \sum_{i=2}^{m} \mu_{1,i}\Delta p_i) \right)^{\frac{1}{2}} + \mu_{2,1}\Delta p_1 + \sum_{i=2}^{m} \mu_{2,i}\Delta p_i \ . \qquad (2.29)$$

As all the elements of the perturbation are fixed except Δp_1 we can pose

$$\alpha = \sum_{i=2}^{m} \mu_{1,i}\Delta p_i \qquad \beta = \sum_{i=2}^{m} \mu_{2,i}\Delta p_i \ ,$$

in which case (2.29) is equivalent to

$$r + jk = \pm \left(\mu_{1,1}(\Delta p_1 + \frac{\alpha}{\mu_{1,1}}) \right)^{\frac{1}{2}} + \mu_{2,1}(\Delta p_1 + \frac{\alpha}{\mu_{1,1}}) - \alpha \frac{\mu_{2,1}}{\mu_{1,1}} + \beta \ . \quad (2.30)$$

Two cases appear depending on the sign of

$$\mu_{1,1}(\Delta p_1 + \frac{\alpha}{\mu_{1,1}}) \ . \quad (2.31)$$

Either (2.31) is nonnegative in which case all the terms in (2.30) are real as well and thus $k = 0$ and

$$r = \pm \left(\mu_{1,1}(\Delta p_1 + \frac{\alpha}{\mu_{1,1}}) \right)^{\frac{1}{2}} + \mu_{1,2}(\Delta p_1 + \frac{\alpha}{\mu_{1,1}}) - \alpha \frac{\mu_{1,2}}{\mu_{1,1}} + \beta \ , \quad (2.32)$$

or (2.31) is negative and it gives birth to a purely imaginary term and

$$r = \mu_{1,2}(\Delta p_1 + \frac{\alpha}{\mu_{1,1}}) - \alpha \frac{\mu_{1,2}}{\mu_{1,1}} + \beta \ ,$$
$$k = \pm \left(\mu_{1,1}(\Delta p_1 + \frac{\alpha}{\mu_{1,1}}) \right)^{\frac{1}{2}} \ .$$

Eliminating the term $(\Delta p_1 + \frac{\alpha}{\mu_{1,1}})$ from both equations we get the equation of a parabola

$$r + \frac{\mu_{1,2}^2}{\mu_{1,1}} k^2 = \gamma \ , \quad (2.33)$$

where γ contains terms depending on $\Delta p_2, \ldots, \Delta p_m$ only.

We now have everything we need to picture the behavior of λ_0 when Δp_1 is increased continuously. First suppose that $\Delta p_2, \ldots, \Delta p_m$ are all zero which in turn implies that α, β, γ are zero. Then we see in Figure 2.7 *that starting from a negative value of Δp_1, the two perturbed eigenvalues computed from the expression (2.28) of $\lambda(\epsilon)$ evolve in the complex plane on the parabola described by (2.33), meet at λ_0 for $\epsilon_1 = 0$, then diverge in opposite directions along the real axis following (2.32). When $\gamma \neq 0$, then the intersection point is shifted an amount equal to γ along the real axis, see* Figure 2.7.

This type of trajectories appears frequently in our examples, especially when multiple eigenvalues appear in our solutions. We give such an example that occurs during the stabilization of a polynomial of degree 2 *in* Figure 2.8.

It is worth noting that the same developments can be made for a complex matrix. In Figure 2.9 we show the strong interactions that can occur when a double complex eigenvalue is perturbed.

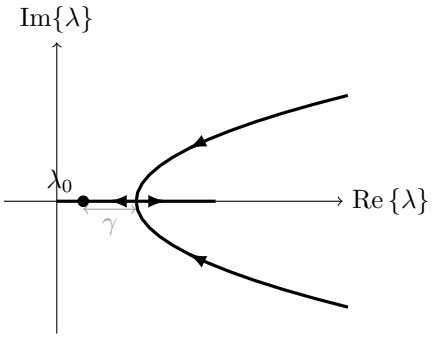


Figure 2.7: Pattern of strong interactions taking place when perturbing a double nonderogatory real eigenvalue of a real matrix by increasing only one of the parameter, represented here for $\mu_{1,1} > 0$. The speed of the eigenvalues increases to infinity as they cross each other. The direction of the arrows is opposite when $\mu_{1,1} < 0$. [89]

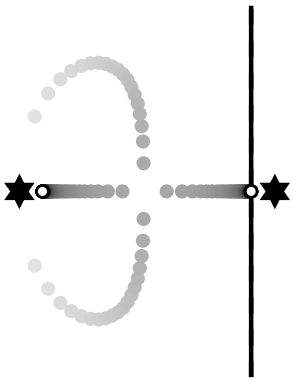


Figure 2.8: Illustration of strong real interaction occuring in the stabilization of the polynomial $a(\lambda) = \lambda^2 + 0.9\lambda - 0.1$ starting from the polynomial $p(\lambda) = (\lambda + 1)^2$. The solution reached is $x(\lambda) = \lambda^2 + 0.9\lambda$. The collision of the two complex conjugate roots and then the divergence along the real axis follow the same pattern as the one in Figure 2.7.

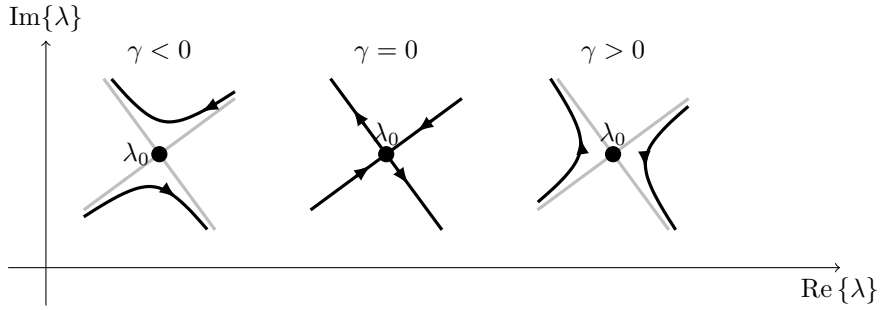


Figure 2.9: Possible patterns of strong interactions taking place when perturbing a double complex eigenvalue of a real matrix by increasing only one of the parameters in the perturbation. The two eigenvalues tend to converge towards the original eigenvalue location before diverging. The parameter γ denotes the contribution of the other elements of the perturbation. [89]

2.3 Singularities and optimality conditions for the set of stable systems

The main difficulty when working with nonconvex sets arises when we want to address the question of optimality of a candidate solution. The question is not easy and even though we concentrate on the more restrictive set of stable systems, very few authors have tried to address the questions that we explore in this part of the manuscript.

Our intention here is to offer a broader perspective on "How the set of stable systems looks" and the results that we present follow two directions. The first direction gives a clear view of the type of geometries that we face in the set of stable systems. The set of stable systems is well-known to be nonconvex but it is rather difficult to envision the type of singularities that our algorithms have to deal with. The literature on the topic is scarce and such a discussion in this thesis is rightly justified. The second direction explores what is currently known about optimality conditions for nonconvex sets. It is quite clear that in the general case no necessary and sufficient condition can be obtained. Still, one could hope for better results for some particular candidate solution of particular problems especially when a particular structure is added to the problem through the use of affine constraints.

In addition to the type of singularities, one could wonder the extent to which the stable set is nonconvex. A meaningful answer to this question can be found in the work of Saydy, Tits and Abed [87] among others. They looked at the question of the stability of the convex hull of two matrices or two polynomials. Their work showed that the segment in $\mathbb{R}^{n\times n}$ formed by convex hull of two matrices defined as

$$A(r) = (1-r)A_0 + rA_1$$

where $A_0, A_1 \in \mathbb{R}^{n\times n}$ and $r \in [0,1]$ could have as many as n intersections with the boundary of the stable set.

2.3.1 Singularities of the boundary of the stable set

The boundary of the set of stable systems is nonsmooth and various types of singularities appear. We reproduce here the results of Mailybaev and Seyranian who dedicated a series of articles to the topic of singularities of the stable set [67, 66, 68]. Their results were based on the work carried by various authors some time before [62, 64].

Let $X(p) \in \mathbb{R}^{n\times n}$ be a matrix which depends smoothly on the vector of parameters $p \in \mathbb{R}^m$. Then the boundary of the stability domain is made of smooth surfaces corresponding to a simple null eigenvalue of $X(p)$ or a pair of complex conjugate eigenvalues with zero real part.

Singularities appear whenever two or more of those smooth surfaces transversally intersect each other at λ_0. Singularities can be separated into classes since singularities can be reduced to a normal form via a local diffeomorphism. In [62], Levantoskii proves that for any finite number of parameters, there is a finite number of classes of singularities. He also underlines that those singu-

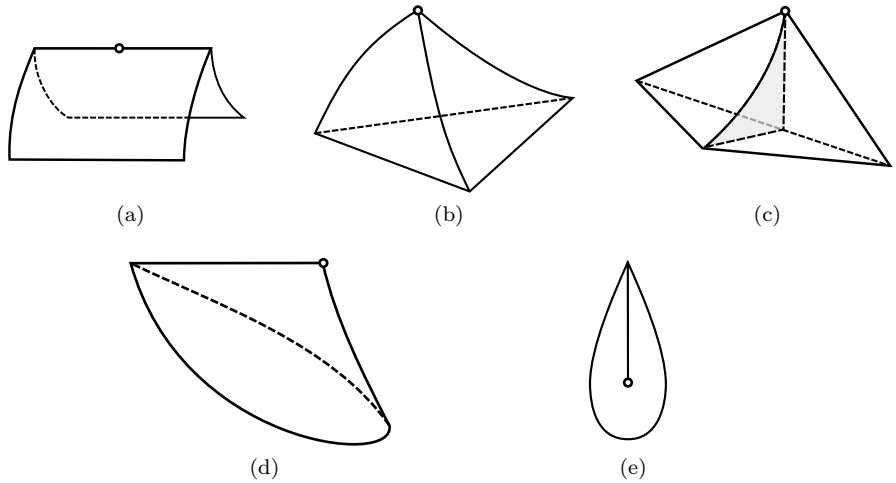


Figure 2.10: Set of possible singularities appearing on the boundary of the stable set when a matrix is dependent on 3 parameters. (a) dihedral angle, (b) trihedral angle, (c) break of an edge;(d) and (e) 3D and top view of a deadlock of an edge. [67, 68]

larities can be computed using the Routh-Hurwitz inequalities but that the complexity of their computation grows factorially.

The computation of these singularities has only been carried in a limited number of publications and for a limited number of settings. All available singularities have been computed in the real case. Levantoskii [62] and Arnold [3] have listed (and represented) the possible singularities for real matrices and polynomials up to $m = 4$.

We represent here the results reported by Mailybaev and Seyranian [67, 68] for a continuous-time real matrix dependent on $m = 3$ parameters. In that case, there is 4 types of singularities of the stable set. These singularities are depicted in Figure 2.10.

When X_0 has multiple eigenvalues on the boundary of the stable set, the Jordan structure of X_0 is related to the type of singularities. The link between the Jordan structure of a matrix and the singularities of the stable set is well established and is detailed in all the works related to the topic. Any two given matrices X_1 and X_2 belong to the same surface if they have the same Jordan structure. We list the various possibilities for a continuous-time real matrix dependent on $m = 3$ parameters in Table 2.1 where the notation $J(0^2)$ means that the singularity is linked to a Jordan form with a block of size 2 for a 0 eigenvalue and $J(0, \pm jw)$ means that a Jordan form contains a simple 0 eigenvalue and a simple pair of conjugate eigenvalues.

In conclusion, due to the geometries that they imply, the singularities of the stable set in the space of the coefficients provides a challenging setting for the search of the nearest stable matrix. As an example, consider the 2D-cut

Singularity	Jordan structures associated to it
Dihedral angle	$J(0^2)$, $J(0, \pm jw)$ or $J(\pm jw_1, \pm jw_2)$
trihedral angle	$J(0^2, \pm jw)$, $J(0, \pm jw_1, \pm jw_2)$ or $J(\pm jw_1, \pm jw_2, \pm jw_3)$
deadlock of an edge	$J((\pm jw)^2)$
break of an edge	$J(0^3)$

Table 2.1: Jordan structures associated with each type of singularities.[67]

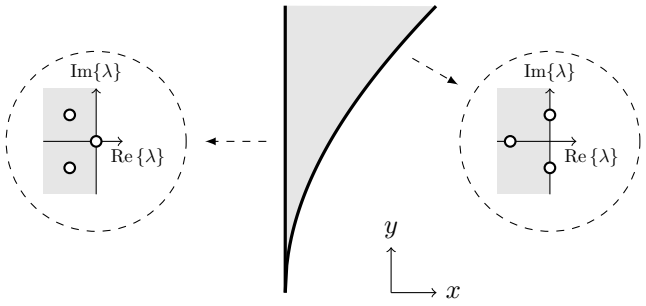


Figure 2.11: Visualization of a singularity of the stable set of real polynomials of degree 3 at the point λ^3 in the direction $x = -\lambda^2 + \lambda + 2$ and $y = \lambda^2 + \lambda$. The borders correspond to specific root configurations, the vertical one is due to polynomials with a root at the origin and two complex conjugate roots with negative real part, the curved border corresponds to polynomials with two purely imaginary conjugate roots and a negative real root.

of a break-of-an-edge singularity depicted in Figure 2.11. It seems clear that if the solution of the problem is located at the singularity, reaching it can be troublesome if our iterates are contained in the interior of the set. Indeed, the developments in [67] have underlined that for some singularities any cone which originates from the singularity and is contained in the stable set also has nonempty interior.

2.3.2 Optimality conditions for the nearest stable system

This section is devoted to the identification of suitable optimality conditions to our problem : the minimization of a convex function over a nonconvex set. We present results contained in the literature and will reproduce here essential developments that add value to this thesis. We start by enumerating some global optimality results to some closely linked problems that are studied in the literature, we then introduce the basic principles of variational analysis applied to the spectral abscissa. This last point is useful for describing a necessary condition for optimality. Finally a sufficient condition for local optimality is given, namely X_* must be a sharp local minimum.

Optimality results in the literature

Led by the pioneering work of Burke, Lewis and Overton [18, 19], authors in the literature have favored an approach to the stability of matrices and polynomials in terms of spectral abscissa and spectral radius [21, 97] rather than in terms of Lyapunov operators. Albeit different in appearance those studies are intrinsically linked to our problem and will put forward the difficulties of obtaining optimality results for this kind of problem.

The only global optimality result on a problem linked to stability appears in [14] where the authors reported results on the minimization of the root radius $\rho(x)$ and on the root abscissa $\alpha(x)$ of a polynomial $x(\lambda)$, in discrete-time and continuous-time respectively, subject to a single affine constraint on the coefficients of the polynomial.

We are unable to state global optimality conditions from which we could characterize the solution to the nearest stable system. But local optimality conditions that are either necessary or sufficient for optimization problems involving the abscissa map are at the heart of the work laid out in a series of articles by Burke and Overton [24, 25] and Burke, Lewis and Overton [18, 19, 21]. Their work is based on the use of a variational analysis tool given by the subgradient of the abscissa map that we present next.

Outside of these references there are virtually no other results. The global optimality result in [14] cannot be transposed or used in our work whereas the results of Burke and Overton can be applied on an individual basis to some instances of our minimization problem.

Local optimality conditions and subgradients

We have presented local optimality conditions for the minimization of a real-valued convex function $f(X)$ over a closed feasible set $\mathcal{C} \subset E$ where E is a given vector space and $\mathcal{C}$ is described by functional constraints in the introductory chapter of this manuscript, see Section 1.3. In particular, if the feasible set $\mathcal{C}$ is convex, a necessary and sufficient condition for the optimality of $\hat{X} \in \mathcal{C}$ is given by

$$\langle \nabla f(\hat{X}), X - \hat{X} \rangle \geq 0 \qquad \forall X \in \mathcal{C} \ .$$

This last condition can be reformulated in terms of the normal cone $N_{\mathcal{C}}$ to $\mathcal{C}$ as

$$0 \in \nabla f(\hat{X}) + N_{\mathcal{C}}(\hat{X}) \ , \tag{2.34}$$

where 0 denotes the zero matrix. As (2.34) is expressed in terms of the normal cone, the condition (2.34) holds for nonsmooth, nonconvex sets as well with the restriction that (2.34) is now necessary but not sufficient for the local optimality of $\hat{X}$.

Subgradients will be useful to describe the optimality conditions (2.34) applied to a feasible set $\mathcal{C}$ defined by

$$\mathcal{C} = \{X | g(X) \leq 0\}$$

with $g(X)$ being nonsmooth.

There is several types of subgradients. A full review of their many characteristics and properties being outside the scope of this thesis, we will not elaborate on them and will refer the reader to [86] for a full study. The definition that we give here applies to functions that are *regular* (or *subdifferentially regular*). As this concept complicates the development we do not discuss its definition and assume our functions to be *regular* throughout. Specifically, the spectral abscissa is subdifferentially regular at $X \in \mathbb{C}^{n\times n}$ if X only has nonderogatory *active* eigenvalues in the following sense,

Definition 2.3. *An eigenvalue $\lambda_i(X)$ is said to be active if* $\mathrm{Re}\{\lambda_i(X)\} = \alpha(X)$.

We refer once again the interested reader to [24, 86] for more advanced details on this topic.

Definition 2.4 (subgradient)**.** *A matrix $Y \in \mathbb{C}^{n\times n}$ is said to be a subgradient of a function $g(X) : \mathbb{C}^{n\times n} \to \mathbb{R}$ at $\hat{X}$ if*

$$g(X) \geq g(\hat{X}) + \langle Y, X - \hat{X}\rangle + o(\|X - \hat{X}\|) \quad \forall X \in \mathbb{C}^{n\times n} , \tag{2.35}$$

which we denote $Y \in \partial g(\hat{X})$ where $\partial g(\hat{X})$ denotes the set of all subgradients of g at $\hat{X}$.

We now define the *horizon subgradient* of a function g, which we define under the assumption that g is continuous.

Definition 2.5 (Horizon subgradient)**.** *Let $g : E \to \mathbb{R}$ be a nonsmooth function over a vector space E and let $\partial g(X)$ be the set of all subgradients of g at X. The horizon subgradient of $g(X)$, denoted $\partial^\infty g(X)$, is defined by*

$$\begin{aligned}\partial^\infty g(X) &= \{W | \exists Y : Y + tW \in \partial g(X) \quad \forall t \in \mathbb{R}_+\} \text{ if } \partial g(X) \text{ is nonempty,}\\ &= \{0\} \quad \text{otherwise.}\end{aligned} \tag{2.36}$$

Basically, under the regularity assumption the horizon subgradient is the recession cone of $\partial g(X)$. The recession cone corresponds to the set of directions of all the rays contained in the cone. By definition, $0 \in \partial^\infty g(X)$. It should be noted that if g is $\mathcal{C}^1$, $\partial g(X) = \{\nabla g(X)\}$, and $\partial^\infty g(X) = \{0\}$.

Let us recall a last definition which will come useful for the next theorem.

Definition 2.6. *Let $\Omega \subset E$ be a closed subset of E. By* $\mathrm{pos}\,\Omega$ *we denote the cone defined by*

$$\mathrm{pos}\,\Omega = \{tZ : Z \in \Omega, t \in \mathbb{R}_+\} .$$

Necessary local optimality condition for the nearest stable system

Let us apply the concepts stated above to our problem. In a string of articles [19, 18, 24, 25] and later [21], Burke and Overton have established the variational properties of the spectral abscissa for matrices and polynomials; most importantly they have shown in [24] when all active eigenvalues are nonderogatory, the spectral abscissa is *subdifferentially regular* which explains our choice

of definition of subgradients (2.35) and horizon subgradients (2.36) which would have been otherwise incorrect. The results presented here are taken from their work and the reference book on variational analysis of Rockafellar and Wets [86] that we already used extensively until this point.

Inspired by the work developed in [24], where the authors developed necessary local optimality conditions for so-called *semistable programming*, we now derive necessary optimality conditions for the nearest stable matrix problem in continuous-time. Following the same approach as in [24], we rewrite it as

$$\begin{aligned} \min_{X\in\mathbb{C}^{n\times n}} \quad & \frac{1}{2}\|A-X\|^2 \\ s.t. \quad & \alpha(X)\leq 0 \end{aligned} \tag{2.37}$$

The equivalence of (2.37) with the original formulation (2.3) is obvious. The following theorem ensues.

Theorem 2.7 (first order necessary optimality condition)**.** *If $\hat{X}\in\mathbb{C}^{n\times n}$ is a solution of (2.37) then*

$$A-\hat{X}\in \operatorname{pos}\partial\alpha(\hat{X})\cup\partial^{\infty}\alpha(\hat{X})\ . \tag{2.38}$$

Proof. The proof is based on elements from [86]. Let us define the feasible set $\mathcal{X}=\{X\in\mathbb{C}^{n\times n}\ :\ \alpha(X)\leq 0\}$, then (2.37) becomes

$$\min_{X\in\mathcal{X}}\ \frac{1}{2}\|A-X\|^2\ , \tag{2.39}$$

Then the necessary first order optimality condition (2.34) applies. Therefore, if $\hat{X}$ is a solution of (2.37) we get together from (2.34) and (2.39) that

$$0\in\hat{X}-A+N_{\mathcal{X}}(\hat{X})\ . \tag{2.40}$$

By Theorem 10.3 in [86] the normal cone $N_{\mathcal{X}}(\hat{X})$ is

$$N_{\mathcal{X}}(\hat{X})=\operatorname{pos}\partial\alpha(\hat{X})\cup\partial^{\infty}\alpha(\hat{X})\ . \tag{2.41}$$

Equations (2.40) and (2.41) leads to (2.38). □

Characterization of a point that satisfies the necessary optimality condition

The necessary optimality condition which we have stated implies that

$$A-\hat{X}\in \operatorname{pos}\partial\alpha(X)\cup\partial^{\infty}\alpha(X)\ . \tag{2.42}$$

but we still need to give an expression for an acceptable Y which we do now based on the work contained in [24]. We underline the fact that we only present results for the spectral abscissa and without any justifications. The work contained in [24] is much richer and contains partial results for an extension to the discrete-time.

Let us consider the set $\mathcal{M}$ which contains the indices of the active eigenvalues. Active eigenvalues were defined in definition 2.3.

Theorem 2.8 (Set of subgradients and horizon subgradients of $\alpha(X)$ [24])**.** *Given a matrix X in Jordan form whose active eigenvalues are nonderogatory, the set of subgradients and horizon subgradients at $\alpha(X)$ is given by the matrices $Y \in \mathbb{C}^{n\times n}$ such that*

$$Y = \begin{bmatrix} W^{(1)} & & \\ & \ddots & \\ & & W^{(p)} \end{bmatrix} \tag{2.43}$$

where each block $W^{(i)}$ associated to an eigenvalue λ_i is zero if i does not belong to the active set $\mathcal{M}$. Whenever $i \in \mathcal{M}$ we have that $W^{(i)}$ is a lower triangular Toeplitz matrix

$$W^{(i)} = \begin{bmatrix} \theta_1^{(i)} & & & \\ \theta_2^{(i)} & \ddots & & \\ \vdots & \ddots & \ddots & \\ \theta_{m^{(i)}}^{(i)} & \dots & \theta_2^{(i)} & \theta_1^{(i)} \end{bmatrix} \tag{2.44}$$

where $m^{(i)}$ is the algebraic multiplicity, $\theta^{(i)} \in \mathbb{C}^{m^{(i)}}$. The elements $\theta_1^{(i)}$ must satisfy

$$\theta_1^{(i)} \in \mathbb{R}, \quad \theta_1^{(i)} \geq 0, \quad \sum_{i\in\mathcal{M}} m^{(i)}\theta_1^{(i)} = 1 \tag{2.45}$$

when $Y \in \partial\alpha(X)$, or must satisfy

$$\theta_1^{(i)} = 0 \ , \tag{2.46}$$

when $Y \in \partial^\infty\alpha(X)$ instead. The elements $\theta_2^{(i)}$ must satisfy

$$\mathrm{Re}\{\theta_2^{(i)}\} \geq 0 \ , \tag{2.47}$$

in both cases.

Sketch of proof. A necessary condition for Y to be a subgradient of any spectral function at X is that Y^* commute with X. If X is in Jordan form, then the only matrices that commute with it have a block structure with triangular Toeplitz blocks. The justification for the conditions (2.45),(2.46) and (2.47) is in [24]. □

In order to completely describe our solution, note that the normal cone (2.41) can now be explained taking into account the *pos* modifier. The pos modifier cancels the effects of the constraint

$$\sum_{i\in\mathcal{M}} m^{(i)}\theta_1^{(i)} = 1$$

in (2.45). In other words, a matrix Y is contained in the set (2.41) if it satisfies the conditions (2.43), (2.44), $\theta_1^{(i)} \in \mathbb{R}$, $\theta_1^{(i)} \geq 0$ and $\mathrm{Re}\{\theta_2^{(i)}\} \geq 0$.

Sufficient optimality condition

A sufficient optimality condition was also stated in the literature using the notion of *sharp* minimum.

Definition 2.9 (Sharp minima [19, 18])**.** *For any Euclidean set E, a point $\hat{x} \in E$ is a sharp minimizer of a function $f : E \to \mathbb{R}$ if and only if there exists a $\delta > 0$ such that*

$$f(\hat{x} + z) - f(\hat{x}) \geq \delta \|z\| \qquad \forall \text{ small } z \in E \ .$$

Proposition 1 (Sharp minima [19, 18])**.** *A point $\hat{x} \in E$ is a sharp minimizer of a function $f : E \to \mathbb{R}$, f being subdifferentially regular, if and only if*

$$0 \in \operatorname{int} \partial f(\hat{x}) \ .$$

When it can be applied, Proposition 1 proves to be a very powerful tool in showing local optimality of nonsmooth functions. For example, Burke, Lewis and Overton [18] used it to characterize the optimality of the solution in some problems. In particular, they proved that the solution $x \in \mathbb{C}^{n-1}$ to the minimization problem

$$\begin{aligned} &\min_{x \in \mathbb{C}^{n-1}} \ \alpha(X(x)) \\ &\quad s.t. \\ &\qquad X(x) = \begin{bmatrix} -x_1 & 1 & & \\ x_1 & & \ddots & \\ \vdots & & & 1 \\ x_{n-1} & & & \end{bmatrix} \end{aligned}$$

is attained when all the element x_i of x are zero. Unfortunately, the sufficiency condition given in Proposition 1 applies mainly when affine constraints are applied on the coefficients such as in the example we have just provided. On the contrary, we do not have affine constraints applied on the coefficients of the matrix in the nearest stable problem.

2.3.3 Benchmark example

We now illustrate the content of the two previous sections, and introduce an example that will be a benchmark for continuous-time algorithms throughout this thesis. Take the following unstable matrix

$$A = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ \nu_0 & & & 0 \end{bmatrix} , \tag{2.48}$$

where $\nu_0 \in \mathbb{R}$. For small $|\nu_0|$, the matrix A can be viewed as a perturbation of the matrix X defined by

$$X_* = \begin{bmatrix} 0 & 1 & & \\ & \ddots & & \\ & & 1 & \\ & & 0 & \end{bmatrix} . \tag{2.49}$$

Since X_* is a nonderogatory matrix, we have seen in Example 17, p.47 that the perturbed matrix A has eigenvalues distributed following the equation

$$\lambda_i(A) = e^{\frac{2i\pi}{n}} \nu_0^{\frac{1}{n}} \quad .$$

Therefore the eigenvalues of A are equidistant points on the circle centered at the origin of the complex plane. The matrix A is thus an unstable matrix. Moreover, following our section on the singularities of the stable set and due to the fact that X_* is a single (nonderogatory) Jordan block, the singularity is a n-dimensional "break of the edge" singularity. Intuitively, we can expect that X_* is actually a local (and sometimes global) optimal solution to the nearest stable system problem in continuous-time

$$\min_{X \in \mathbb{S}^n} \frac{1}{2} \|A - X\|^2 . \tag{2.50}$$

What is unusual with this example is the range of values of ν_0 for which X_* seems to be a local optimal solution. We give the following conjecture regarding this observation.

Conjecture 2.10. *Provided $n > 2$ and $|\nu_0| < 1$, $\nu_0 \in \mathbb{R}$, X_* as given in (2.49) is a local optimal solution to the problem (2.50).*

Before discussing the results that led to this conjecture, let us stress that we think that for values of $|\nu_0| < \gamma_0$, $\gamma_0 \approx 0.5$, the matrix X_* is actually a, possibly unique, global optimal solution. But since it is difficult to fix the exact value of γ_0 for which X_* is indeed the global optimal solution, we prefer discussing the local optimality of X_*. Let us now discuss elements that led to the Conjecture 2.10.

Local optimality conditions

Let us show that X_* satisfies the necessary local optimality conditions that were detailed in Theorem 2.7 and 2.8. Let us recall their implications here. We must show that the gradient of the objective function (2.50) belongs to the cone generated from the set of subgradients and horizon subgradients of the spectral abscissa. This is in turn verified if the gradient of the objective has a Toeplitz structure whose elements satisfy the algebraic conditions (2.45)-(2.47) in Theorem 2.8.

Let us check that those conditions are indeed verified. First of all, X_* contains only one nonderogatory eigenvalue which is active. Then, as the matrix

X_* is already in Jordan form we can apply the necessary optimality conditions (2.38) to our problem. We have

$$\begin{bmatrix} & \\ \nu_0 & \end{bmatrix} = A - X_* \ .$$

Therefore, we can check visually that since the elements of the main diagonal of $A - X_*$ are all 0, $A - X_*$ satisfies the condition (2.46) which states that those elements can be zero. Moreover, since, for $n > 2$, all the elements on the first subdiagonal of $A - X_*$ are all zeros, $A - X_*$ also satisfies the condition (2.47) which requires that those elements be nonnegative. Hence, we conclude that our solution X_* satisfies the necessary optimality condition (2.38). Note that this necessary optimality condition holds for any value of ν_0 whether real or complex, provided that $n > 2$.

Limits of the conjecture

Now that we have shown that our solution X_* satisfies the necessary optimality conditions, we are able to discuss the reasons that led us to restrict the conjecture to a specific range of values of ν_0.

Let us first note that the case $\nu_0 = 0$ does not need to be treated since then A is already *marginally* stable and thus belongs to the closure of the stable set. We choose to limit our conjecture to real values of ν_0 on the one hand and to $|\nu_0| < 1$ on the other hand. We discuss both of those constraints in the following.

For values of ν_0 such that $|\nu_0| > 1$, ν_0 is larger than any other element of the matrix A and thus better solutions are easily found. For $|\nu_0| = 1$, the matrix A (2.48) becomes a normal matrix. The proposed solution X_* (2.49) achieves an objective value of

$$\|A - X_*\| = 1$$

but other solutions that achieve the same objective value can be easily computed. Indeed, any matrix $\hat{X}$ which is a circulant permutation of X_* is also stable and achieves the same objective value.

Since A is a normal matrix, a natural candidate solution is the normal matrix B_{ν_0} computed as the projection of A onto the cone of the positive definite matrices (in the 2-norm) for a fixed value of ν_0. As both matrices are normal and the Frobenius norm is a unitarily invariant norm, we have, in view of example 14, that

$$\|A - B_{\nu_0}\| = d(\lambda(A), \lambda(B_{\nu_0})) \ ,$$

where $d(\lambda(A), \lambda(B_{\nu_0}))$ denotes the minimal distance obtained by permutation of the order of the eigenvalues of A and B_{ν_0}. For $n = 3$, B_{ν_0} is closer than X_* to A if $\nu_0 = -1$ and at the same distance if $\nu_0 = 1$. Indeed, if $\nu_0 = -1$ the

eigenvalues of A are

$$\lambda(A) = \begin{bmatrix} -1 \\ \frac{1}{2} + j\frac{\sqrt{3}}{2} \\ \frac{1}{2} - j\frac{\sqrt{3}}{2} \end{bmatrix} ,$$

so $\|A - B_{-1}\| = \frac{\sqrt{2}}{2}$. Now if $n = 3$ and $\nu_0 = 1$, the eigenvalues are mirrored with respect to the imaginary axis and therefore $\|A - B_1\| = 1$. For $n = 4$ both distances are equal to 1 and for $n = 5$ the normal matrix $B_{|\nu_0|=1}$ is further away from A than X_*. Note that we have not proved that B_{ν_0} or $\hat{X}$ satisfies the necessary optimality condition stated in Theorem 2.7. However this is difficult in general since it requires the computation of the Jordan structure of both matrices.

A strong indicator of local optimality is obtained through numerical methods. We have carried several experiments to see if X_*, B_{ν_0} and $\hat{X}$ were indeed local minimizers of the problem (2.50). In order to look for this, we solved multiple instances of the problem from starting points that were perturbations of the points X_*, B_{ν_0} and $\hat{X}$. Numerical algorithms have consistently returned the expected solutions. Therefore, we can say that numerical experiments support our conjecture of the presence of a local minimum at X_*.

For values of $0 < |\nu_0| < 1$, the candidate solution X_* still satisfies the necessary local optimality conditions. Numerical experiments also tend to indicate that X_* is a local optimal solution. Nonetheless, the same experiments showed that we cannot confirm that X_* is a *global* optimum for any values of ν_0, as the next example shows.

Example 21. *Take* $\nu_0 = -0.75$. *Then* $\|A - X_*\| = 0.75$ *but the matrix* $\hat{X}$ *given by*

$$\hat{X} = \begin{bmatrix} -0.2811 & 0.8860 & 0.1022 \\ -0.1717 & -0.3077 & 0.8860 \\ -0.5184 & -0.1717 & -0.2811 \end{bmatrix}$$

achieves an objective value of $\|A - \hat{X}\| = 0.6338$. *The matrix* $\hat{X}$ *does not have any particular structure and has eigenvalues given by* -0.87, $\pm j0.7805$. *For* $n = 3$ *better solutions than* X_* *could be found for values of* $|\nu_0| > 0.5$.

We restricted our analysis to the case $\nu_0 \in \mathbb{R}$ but as pointed out by Michael Overton [78] one could consider ν_0 as belonging to $\mathbb{C}$. While X_* still satisfies the local optimality conditions for any ν_0, numerical experiments were inconclusive. Therefore we did not extend our conjecture to the complex case.

The case $n = 2$

We have emphasized that X_* as defined in (2.49) is optimal for $n > 2$. We will now study why this is not always true for $n = 2$ and give alternative solutions when possible. For $n = 2$, A and X_* are given by

$$A = \begin{bmatrix} & 1 \\ \nu_0 & \end{bmatrix} \qquad X_* = \begin{bmatrix} & 1 \\ 0 & \end{bmatrix} . \tag{2.51}$$

In order to satisfy the necessary optimality condition described in Theorem 2.7, we must also verify that $A - X_*$ satisfies the condition (2.47). This imposes that

$$\mathrm{Re}\{\nu_0\} \geq 0 \ .$$

Let us consider the case where $\nu_0 \in \mathbb{R}$ for now. X_* is only optimal for $\nu_0 \geq 0$. This was expected and illustrates the fact that if $\nu_0 < 0$, the eigenvalues of A are

$$\lambda = \pm j\sqrt{\nu_0} \ .$$

Thus, the eigenvalues of A are already marginally stable and the solution to the nearest matrix problem (2.50) is given by the matrix A itself.

Let us now go further and relax our constraint on ν_0 and allow $\nu_0 \in \mathbb{C}$, $|\nu_0| < 1$. In that case, the necessary optimality conditions for X_* still holds as long as $\mathrm{Re}\{\nu_0\} \geq 0$. One could thus think that X_* might be a *global* optimal solution for small magnitude of ν_0. Unfortunately various experiments have shown that this is not true as the following example suggests.

Example 22. *Let ν_0 be given by $\nu_0 = \frac{2}{3} + j\frac{1}{2}$ and where the solution X_* was obtained numerically using the Hanso method which we present later on in Section 3.4.*

$$A = \begin{bmatrix} & 1 \\ \frac{2}{3} + j\frac{1}{2} & \end{bmatrix} \quad \hat{X}_* = \begin{bmatrix} -0.2218 & 0.9467 - j0.0355 \\ 0.0276 - j0.0740 & -0.2218 \end{bmatrix} \ . \tag{2.52}$$

The distance to A is given by

$$\|A - \hat{X}_*\|_F = 0.8321 \ ,$$

while we have

$$\|A - X_*\|_F = |\nu| = 0.8333 \ ,$$

Still, as in the real case, X_* might still be a local optimal solution to the problem. A very strong indicator that this is true is obtained by repeating the same experiment with various starting points taken as small random perturbations of X_*. It was observed that for one half of the 12 runs we carried the solution converged to X_*, and to $\hat{X}_*$ given by (2.52) for the other half. This example illustrates perfectly the *local* implications of the optimality condition stated in Theorem 2.7.

Conclusions on chapter 2

The contribution of this chapter is two folds. First, we brought together in one place various elements of theory that can be found in the literature related to the set of stable systems. Second, we provided a meaningful example of an unstable matrix to which the theoretic elements could be applied, in particular the necessary optimality conditions of [24].

The nearest stable system is a difficult problem due to the properties of its domain. The nonconvexity and nonsmoothness of the stable set make the

search for any guarantee of global optimality of the solution hopeless. We highlighted the kind of nonconvexity we could face and how often they appear.

Singularities appear because the eigenvalues and roots are nonlinear functions of their coefficients. Under the action of some structured or unstructured perturbation on their coefficients, the eigenvalues/roots of a system follow trajectories that can be predicted. We have provided bounds on the perturbations, described expressions and techniques to find the leading terms of expansions describing those trajectories and have illustrated some typical interactions that occur between eigenvalues.

We have reproduced optimality results associated with closely related problems. Most results in the literature imply local optimality of the solution but this does not apply to the kind of problems we are dealing with. We presented necessary local optimality conditions in one hand, and sufficient local optimality conditions in the other hand. The necessary optimality condition applies to our problem but the sufficient optimality condition that we presented can only be useful when the constraints are affine.

Taken all together the results presented in this chapter give some insight on the complexity underlying the problem and help us visualize the type of problem we face and the solutions that can arise due to the specific geometry of the set. All aforementioned results were applied to the unstable companion matrix

$$A = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ \nu_0 & & & 0 \end{bmatrix}$$

and we discussed a number of possible cases and solutions.

Chapter 3

Methods for matrices

This chapter explores innovative methods for solving the nearest stable matrix problems both in continuous- and in discrete-time which we write as

$$\begin{array}{rl} \inf\limits_{X,P} & \dfrac{1}{2}\|X-A\|_F^2 \\ \text{s.t} & \mathcal{A}_P(X) \prec 0 \ , \\ & P = P^* \succ 0 \ . \end{array} \tag{3.1}$$

where the notation $\mathcal{A}_P(X)$ introduced in Chapter 1 denotes the Lyapunov operator

$$\mathcal{A}_P(X) = XP + PX^* \tag{3.2}$$

in continuous-time or the Stein operator

$$\mathcal{A}_P(X) = XPX^* - P \tag{3.3}$$

in discrete-time. The notations (3.2) and (3.3) enable us to avoid the dissociation of the developments in continuous- and in discrete-time case which will ease the burden arising from the equations.

Most existing methods are inefficient for solving the formulation (3.1). The nonconvex nature of the stable set makes it difficult to prove convergence to a minimum. In particular, one might wish to apply a path-following schemes on an augmented formulation of the algorithm using a self-concordant barrier function. In general, path following scheme applied to a minimization problem with a nonconvex feasible set are not guaranteed to converge since bifurcations can occur in the central path in which case the Hessian of the barrier becomes indefinite and any convergence guarantee is lost. Alternative schemes are thus sought.

The essential idea behind the first method that we explore is that it is possible to locally approximate the nonconvex set of the stable matrices using a very simple smooth convex object : the Dikin ellipsoid. Dikin ellipsoids have very nice mathematical properties inherited from the mathematical object that defines them : a self-concordant function. We present here the matrix version of the algorithm and will present its polynomial counterpart in the next chapter.

The second method we describe concentrates on the continuous-time formulation of the problem and follows an entirely different approach compared to the first method. It is based on an adequate reformulation of the problem which has better mathematical properties than the original formulation (3.1). More classic methods can then be applied to find a suitable solution.

The chapter is organized as follows :

Section 3.1 introduces a first iterative method for finding both the nearest stable matrix both in continuous-time and in discrete-time. It explores the concepts behind the construction of the Dikin ellipsoid. The section concludes with a short analysis on the success and the failures of the method.

Section 3.2 starts from the results obtained in the first section and suggests an augmented version of the algorithm to correct the weaknesses of the first version of the method. Some time is taken to make a convergence analysis of the resulting method.

Section 3.3 presents a method for the resolution of continuous-time instances of the nearest stable matrix problem. Several reformulations of the problem are presented and a method is proposed to solve one of the obtained reformulations. A convergence proof of the method is obtained.

Section 3.4 closes this chapter by a comparison of the performances of the various algorithms presented in the chapter and other well-known algorithms.

3.1 Nearest Stable system Using Successive Convex Approximations

The ideas presented in this section were developed independently from other results found in the literature which are related to ours. In particular it is worth mentioning the related ideas of Florian Jarre [52, 51] and Yinyu Ye [99].

The section is divided in three parts; the first part details the construction of the Dikin ellipsoid; we then show how the nearest stable system problem can be solved on the ellipsoid and finally state an iterative algorithm in a third part. The section concludes with two very short numerical examples that illustrate the successes and the failures of the method, followed by a discussion of the causes of these failures.

3.1.1 Dikin Ellipsoid for the stability region

We have already mentioned several times that the constraints of (3.1) define a highly nonconvex domain. Therefore, a first step toward the resolution of (3.1) is achieved by a convexification of the domain of the problem. Several types of approximations are available; a good example is provided by LMI approximations. LMI's are able to fit well the stability boundary [85] but they do not admit an iterative refinement (this is a one shot procedure). The solution that we present here has the advantages of being convex, easy to describe, and approaching quite well the stability boundary.

Our approach is the following : for a given P that satisfies the constraints of (3.1), there are many possible stable candidates X. As a consequence, for $P \succ 0$ fixed, we can define

$$\mathbb{S}_P \stackrel{\text{def}}{=} \{X \in \mathbb{C}^{n\times n} \ : \ -\mathcal{A}_P(X) \succ 0\} \subset \mathbb{S}^n, \tag{3.4}$$

and we have

$$\bigcup_{P\succ 0} \mathbb{S}_P = \mathbb{S}^n.$$

The reader is referred to Figure 3.1 for understanding the inclusion relations between the various sets of stable matrices we have and will define here.

Let

$$f(Y) = \begin{cases} -\ln\det Y, & \text{if } Y \succeq 0, \\ +\infty & \text{otherwise .} \end{cases}$$

If $P \succ 0$ then the set $\mathbb{S}_P$ has nonempty interior, and the function $b_P(X) \ : \ \mathbb{C}^{n\times n} \to \mathbb{R} \cup \{+\infty\}$:

$$b_P(X) \stackrel{\text{def}}{=} f(-\mathcal{A}_P(X)) \tag{3.5}$$

is a self-concordant barrier for $\mathbb{S}_P$. An essential result of the theory of interior point methods for optimization states, loosely speaking, that the open unit ellipsoid centered at a point in the domain of a self-concordant function, with shape defined by the Hessian of the barrier evaluated at that point, belongs to its domain [73]. Applying this to b_P, we obtain the following result :

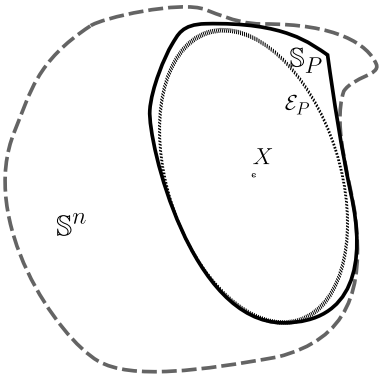


Figure 3.1: Sketch of the space of stable matrices $\mathbb{S}^n$, the set of stable matrices for a given $P : \mathbb{S}_P$, and its convex subset the Dikin ellipsoid $\mathcal{E}_P$ of center X.

Theorem 3.1 (Dikin Ellipsoid)**.** *Let $X \in \mathbb{S}^n$ and $P \in \mathbb{K}^n_{++}$ be given and be such that $\mathcal{A}_P(X) \prec 0$. For any $0 \leq \alpha < 1$, the ellipsoid*

$$\mathcal{E}(P, X; \alpha) \stackrel{def}{=} X + \{H \in \mathbb{C}^{n\times n} \ : \ \langle b''_P(X)H, H\rangle \leq \alpha\} \tag{3.6}$$

belongs to $\mathbb{S}_P$.

Proof. First note that the assumption on X is equivalent to stating that X lies in the domain of b_P. The inclusion follows from Theorem 4.1.5 in [73]. See also [35]. □

It is actually possible to consider other self-concordant barrier functions than the *log-det* barrier. But the *log-det* has the best barrier parameter [73].

The Dikin ellipsoid is quite a powerful tool in order to deal with our nonconvex domain. It yields ellipsoidal approximations of the stable domain on which the resolution of the optimization problem (3.1) is easy and straightforward. Still, the choice of suitable P and X is critical for obtaining a convex domain which is as close as possible to the original nonconvex set. We address the question of finding an initial X later on in the part dedicated to the search of starting points in Section 3.1.4 but one should have in mind that it can equally be provided by the user at the start of the algorithm. We will focus for now on an appropriate choice of P for a given X which amounts to optimizing the shape of our ellipsoidal approximation.

Optimizing the ellipsoid If X is a stable matrix, then by Theorem 1.62 there must exist $P \succ 0$ such that $\Psi_X(P)$ defined for a fixed X by

$$\Psi_X(P) = XP + PX^*$$

in continuous-time or

$$\Psi_X(P) = XPX^* - P$$

is negative definite. We now describe a procedure for computing such a P. Note that Theorem 1.62 also states that whenever $\Psi_X(P) \prec 0$ for a stable matrix X,

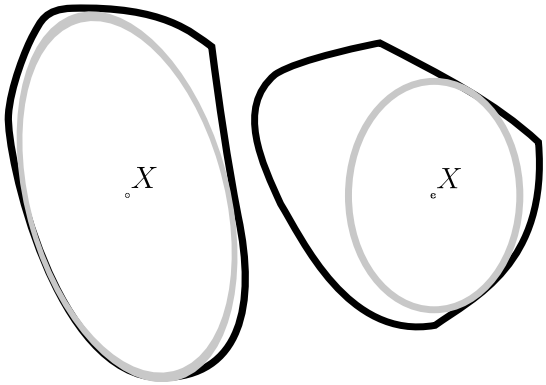


Figure 3.2: Sketch of two possible choices of P and their implication on the volume of both $\mathbb{S}_P$ (in black) and the Dikin Ellipsoid (in grey). The shape of the Dikin ellipsoid is only illustrative, there is no guarantee that its volume would be maximal inside $\mathbb{S}_P$.

---

then necessarily P is positive definite. Since the set of positive definite matrices P satisfying the condition $\Psi_X(P) \prec 0$ is a cone, we can limit our search to the bounded convex set

$$\mathbb{P}_X \stackrel{\text{def}}{=} \{P \in \mathbb{K}^n_{++} \ : \ \Psi_X(P) \prec 0, \quad \langle P, I\rangle = 1\}.$$

Any matrix P belonging to $\mathbb{P}_X$ satisfies our requirements. But for each P the Dikin ellipsoid associated to that choice is different both in volume and in orientation. Indeed, the set $\mathbb{S}_P$ changes for each choice of P and so does the Dikin ellipsoid as pictured in Figure 3.2. We will, however, choose P as a *central* point of this set. The motivation behind this choice is the following. By taking a P which lies on the boundary of $\mathbb{P}_X$, $\mathcal{A}_P(X)$ will be singular and hence X will be said to be marginally stable for this choice of P. Conversely, if we take P as a central point of the set of all possible P's, then our hopes are that X will somehow be a central point of the set $\mathbb{S}_P$. Hence, the volume of the implied Dikin ellipsoid will always be large but there is no proof that this is the best choice. Still, numerical experiments that were based on polynomial instances of the problem were carried out and tend to confirm that a central P yields better convergence in practice than random ones. We discuss further in Chapter 4 the material related to randomly generated stability certificates.

In order to compute the adequate P, we equip $\mathbb{P}_X$ with the self-concordant barrier

$$b_X(P) \stackrel{\text{def}}{=} f(-\Psi_X(P)), \qquad f(Y) = -\ln\det Y,$$

and choose $P = P_*$ to be the *analytic center* of the set $\mathbb{P}_X$:

$$P_* \stackrel{\text{def}}{=} \arg\min\{b_X(P) \ : \ P \in \mathbb{P}_X\}. \tag{3.7}$$

Theorem 3.2. *If $X \in \mathbb{S}^n$, then the solution of (3.7) is unique, and is given in the continuous-time case by the Lyapunov equations*

$$X^*Q^{-1} + Q^{-1}X + nI = 0, \tag{3.8}$$
$$XP + PX^* + Q = 0, \tag{3.9}$$

and in the discrete-time case by the Stein equations

$$X^*Q^{-1}X - Q^{-1} + nI = 0, \tag{3.10}$$
$$XPX^* - P + Q = 0. \tag{3.11}$$

Proof. Since b_X is a self-concordant barrier function and $\mathbb{P}_X$ is bounded, P_* must exist and is unique. This solution P_* must satisfy the first order necessary optimality conditions

$$b'_X(P) + \mu I = 0, \tag{3.12}$$
$$\langle P, I\rangle = 1, \tag{3.13}$$
$$\mu \in \mathbb{R},$$

which are obtained by computing the derivative of the Lagrangian of (3.7) with respect to P (3.12) and μ (3.13) respectively. We will first show that

$$\mu = n\ . \tag{3.14}$$

Indeed, observe that b_X is logarithmically homogeneous, i.e.

$$\begin{aligned} b_X(tP) &= -\ln\det(\Psi_X(tP)) \\ &= -\ln\det(t\Psi_X(P)) \\ &= -\ln(t^n \det\Psi_X(P)) \\ &= -b_X(P) - n\ln t, \qquad t > 0, \end{aligned} \tag{3.15}$$

and hence

$$\begin{aligned} -n \overset{(3.15)}{=} \tfrac{d}{dt} b_X(tP)|_{t=1} &\overset{\text{def}}{=} \langle b'_X(tP), P\rangle|_{t=1} \\ &\overset{(3.12)}{=} -\mu\langle I, P\rangle \\ &\overset{(3.13)}{=} -\mu. \end{aligned}$$

Note that the second equality on the first line is a simple application of the chain rule on $b_X(tP)$, see Section 1.2.2 for more explanation on the chain rule. Then, we can verify by inspection using our definition of adjoint (1.2) that the adjoint of Ψ_X is given by, in the continuous and discrete-time case respectively,

$$\Psi^*_X(Y) = X^*Y + YX, \tag{3.16}$$
$$\Psi^*_X(Y) = X^*YX - Y\ , \tag{3.17}$$

see Section 1.1.1 for details on the adjoint of an operator. If we now define

$$Q \overset{\text{def}}{=} -\Psi_X(P), \tag{3.18}$$

which is exactly (3.9) and (3.11), we can once more make use of the chain rule to obtain

$$b'_X(P) = -\Psi^*_X(f'(-\Psi_X(P))) \overset{(3.18)}{=} \Psi^*_X(Q^{-1}). \tag{3.19}$$

Finally, by substituting (3.14) and (3.19) into (3.12) we prove (3.8) and (3.10).

It remains to note that by Corollary 1.63, p.25, since X is stable, (3.8) (resp. (3.10)) has a unique solution in Q^{-1} and, in turn, (3.9) (resp.(3.11)) a unique solution in P. □

As a conclusion to this section, we give some insight on how to solve the systems of Theorem 3.2. An easy-to-implement way of solving the systems is to find the Cholesky decomposition of Q^{-1} by solving (3.8) or (3.10), invert it, construct Q and then find the Cholesky decomposition of P by solving (3.8) or (3.10). An implementation of this technique exists, see [42] and references therein. For some settings, this step is the limiting one in terms of complexity of the method. It is known that each Lyapunov equation requires $\mathcal{O}(n^3)$ operations to reach a solution. Though easy to use, this technique does not take advantage of the special structure of the equations, when available. This is especially true for companion matrices for which better complexity could be hoped for. We will talk more about this point later on in the chapter dedicated to polynomials.

3.1.2 Barrier Projection Method for general matrices

As the Dikin ellipsoid (3.6) is included in $\mathbb{S}_p$, itself included in $\mathbb{S}^n$, a nearest stable matrix can be found by solving the problem on successive approximations of the stable set described by the Dikin ellipsoid. Finding the nearest stable matrix in the Dikin ellipsoid amounts to solving

$$\begin{aligned} \min_H \ & \tfrac{1}{2}\|X + H - A\|_F^2 \\ & \langle b''_P(X)H, H\rangle \leq \alpha \ . \end{aligned} \tag{3.20}$$

The first order optimality conditions which are both necessary and sufficient are given by

$$X + H - A + \lambda\,(b''_P(X)[H]) = 0 \tag{3.21}$$

$$\frac{\lambda}{2}\,(\langle b''_P(X)[H], H\rangle - \alpha) = 0 \tag{3.22}$$

$$\lambda \geq 0 \ .$$

Note that in view of (3.21) and the fact that $X + H$ is stable and A is not, λ cannot be zero. Hence, the complementary slackness condition (3.22) imposes that the optimal solution for H will give a point on the boundary of the ellipsoid since

$$\langle b''_P(X)H, H\rangle = \alpha \ .$$

The remainder of the section describes the steps that one needs to implement in order to solve the optimality conditions (3.21)-(3.22). We confine

ourselves to a theoretic resolution of the equations and will give explicit expressions in the next section. The advantage of this approach is that the presented equations are as simple as they can be. The resolution is somewhat straightforward and the result obtained in the next section can be plugged into the solution that we present now.

Let us define the linear operator

$$\mathcal{B} \ : \ \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n} \ : \ H \to b''_P(X)[H] \ . \tag{3.23}$$

The operator $\mathcal{B}$ is positive definite and is thus diagonalizable. Let us consider that λ is known, then

$$H(\lambda) = (\mathcal{I} + \lambda\mathcal{B})^{-1}(A - X) \ .$$

Plugging this solution back into our first order optimality condition (3.22) we get

$$\langle \mathcal{B}(\mathcal{I} + \lambda\mathcal{B})^{-1}(A - X), (\mathcal{I} + \lambda\mathcal{B})^{-1}(A - X)\rangle = \alpha$$

which is an equation in the Lagrange multiplier λ. Following our hypothesis, $\mathcal{B}$ is diagonalizable,

$$\mathcal{B} = \mathcal{U}\mathcal{D}\mathcal{U}^*$$

where $\mathcal{U}$ is a unitary operator and $\mathcal{D}$ is a diagonal operator. Defining $\hat{A} - \hat{X} = \mathcal{U}^*(A - X)$ we obtain the equivalent equation

$$\langle \mathcal{D}(\mathcal{I} + \lambda\mathcal{D})^{-1}(\hat{A} - \hat{X}), (\mathcal{I} + \lambda\mathcal{D})^{-1}(\hat{A} - \hat{X})\rangle = \alpha \ . \tag{3.24}$$

As a diagonal operator can only perform element-wise multiplication by a constant, i.e.

$$\mathcal{D}(Y) = E \odot Y$$

for some given dense matrix E, we can define the matrix M as

$$M_{ij} = \frac{E_{ij}}{(1 + \lambda E_{ij})^2} \ .$$

Then (3.24) can be expressed as

$$\langle M \odot (\hat{A} - \hat{X}), (\hat{A} - \hat{X})\rangle = \alpha$$

or equivalently

$$\psi(\lambda) \stackrel{\text{def}}{=} \mathrm{Re}\left\{ \sum_{i,j} \frac{E_{ij}(\hat{A} - \hat{X})^2_{ij}}{(1 + \lambda E_{ij})^2} \right\} - \alpha = 0 \ . \tag{3.25}$$

Since $\mathcal{B}$ is positive definite and therefore E has positive elements, the poles of $\psi(\lambda)$ are all negative. Moreover the graph of $\psi(\lambda)$ shows that the function is positive for $\lambda = 0$, is strictly decreasing for $\lambda > 0$ and tends to $-\alpha$ as $\lambda \to \infty$. This enables us to conclude that there is one and only one positive root for

Algorithm 3.1 Barrier Projection Method

Choose a stable matrix X_0; **for** $k \geq 0$ **do**

- (a) Find P_k and Q_k by solving (3.8)-(3.11);
- (b) Solve (3.25) to get λ_k;
- (c) Determine H_k using (3.21);
- (d) $X_{k+1} = X_k + H_k$ is the projection of A onto the ellipsoid $\mathcal{E}(P_k, X_k; \alpha)$.

end

$\psi(\lambda)$. Based on the same argument we deduce that the shape of the function is convex for $\lambda > 0$ and we can thus use the Newton-Raphson iteration

$$\lambda_+ = \lambda - \frac{\psi(\lambda)}{\psi'(\lambda)} \ .$$

We now give the Barrier Projection Method in algorithm 3.1. In this algorithm, the difficulty surrounding the matrix case arises from the expression of $\mathcal{B}$ for the Hessian of the barrier function. Before developing detailed expressions for the matrix case, let us make a comment on a suitable choice of the value of α in algorithm 3.1. The value of α determine the radius of the ellipsoid inside $\mathbb{S}_P$. For $\alpha = 1$ the Dikin ellipsoid actually touches the boundary of $\mathbb{S}_P$ but it does not necessarily mean that it touches the boundary of $\mathbb{S}^n$. Taking α very close to 1 will ensure larger, and in most cases fewer, steps compared to smaller values of α.

Detailed expressions for matrices

Theorem 3.3. *The Hessian of the barrier function is given by*

$$\langle b''_P(X)H, H\rangle = 2\langle Q^{-1}\left(HP + PH^*\right)Q^{-1}P, H\rangle \tag{3.26}$$

in the continuous-time case and by

$$\begin{aligned}\langle b''_P(X)H, H\rangle = {} & 2\langle Q^{-1}\left(HPX^* + XPH^*\right)Q^{-1}XP, H\rangle \\ & + 2\langle Q^{-1}HP, H\rangle \ ,\end{aligned} \tag{3.27}$$

in the discrete-time case, and where Q is given in (3.18).

Proof. Let us consider the first and second order derivatives of $\mathcal{A}_P(X)$ denoted by $\mathcal{C}_P(H)$ and $\mathcal{D}_P(H)$ respectively. For the discrete-time setting, we have for the first order derivative

$$\begin{aligned}\mathcal{C}_P(H) &\stackrel{\text{def}}{=} D\mathcal{A}_P(X)[H] \\ &= XPH^* + HPX^* \ .\end{aligned} \tag{3.28}$$

The second order derivative is given by

$$\mathcal{D}_P(H) \stackrel{\text{def}}{=} D^2\mathcal{A}_P(X)[H,H] \tag{3.29}$$
$$= 2HPH^* \ . \tag{3.30}$$

For continuous-time systems, we find $\mathcal{C}_P(H) = \mathcal{A}_P(H)$ and $\mathcal{D}_P(H) = 0$. The adjoint of $\mathcal{C}_P(H)$ and $\mathcal{D}_P(H)$ are

$$\mathcal{C}_P^*(Y) = 2YXP \quad \text{and} \quad \mathcal{D}_P^*(Y) = 2YHP$$

in discrete-time, and

$$\mathcal{C}_P^*(Y) = 2YP$$

in continuous-time. See Section 1.1.1 for details on how to compute the adjoint of an operator. Since $f'(Y) = -Y^{-1}$, it can be shown that $f''(Y)H = Y^{-1}HY^{-1}$, which together with the chain rule (see Section 1.2.2), gives

$$\begin{aligned}
\langle b'_P(X), H\rangle &\stackrel{(3.18)}{=} \langle f'(Q), -\mathcal{C}_P(H)\rangle \\
&= \langle Q^{-1}, \mathcal{C}_P(H)\rangle \\
\langle b''_P(X)H, H\rangle &= \langle f''(Q)\mathcal{C}_P(H), \mathcal{C}_P(H)\rangle - \langle f'(Q), \mathcal{D}_P(H)\rangle \\
&= \langle Q^{-1}\mathcal{C}_P(H)Q^{-1}, \mathcal{C}_P(H)\rangle \\
&\quad + \langle Q^{-1}, \mathcal{D}_P(H)\rangle \\
&= \langle \mathcal{C}_P^*(Q^{-1}\mathcal{C}_P(H)Q^{-1}), H\rangle \\
&\quad + \langle \mathcal{D}_P^*(Q^{-1}), H\rangle \ .
\end{aligned} \tag{3.31}$$

At this point, all that remains to be done is to introduce (3.28) and (3.30) in the Hessian of the barrier (3.31) to obtain the desired result. □

Now that we have obtained the expressions for the Hessian of the self-concordant barrier function, we can write the first optimality condition (3.21) explicitly. We have

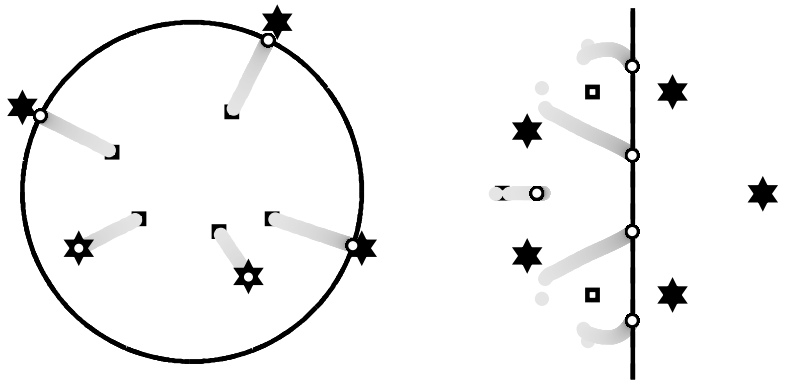
$$X + H - A + 2\lambda\left(Q^{-1}\left(HP + PH^*\right)Q^{-1}P\right) = 0 \ , \tag{3.32}$$

in the continuous-time case and

$$X + H - A + 2\lambda\left(Q^{-1}\left(HPX^* + XPH^*\right)Q^{-1}XP + Q^{-1}HP\right) = 0 \ , \tag{3.33}$$

in the discrete-time case. An efficient way for the resolutions of the equations (3.32) and (3.33) could not be found either for the continuous-time expression or for its discrete-time counterpart. Nonetheless, (3.28) and (3.30) are self-adjoint linear operators in H and we can obtain a matrix representation B of $\mathcal{B}$ by using Kronecker products. Then, (3.32) is vectorized,

$$\operatorname{vec}(H) = [I + \lambda B]^{-1}\operatorname{vec}(A - X) \ .$$



(a) (b)

Figure 3.3: Nearest stable matrix to (a) a normal matrix with eigenvalues given by (3.35); (b) the unstable matrix given by (2.48). Stars represent the position of the original eigenvalues; black squares, the eigenvalues of the starting point;grey scaled circles show the evolution of the iterates; white dots show the eigenvalues of the final solution.

Hence, λ is obtained by solving an equation similar to (3.25). If we label $h = \mathrm{vec}(H)$, $a - x = \mathrm{vec}(A - X)$ we get an equation in λ which is

$$\psi(\lambda) = \sum_i \frac{d_{B,i}(\hat{a} - \hat{x})_i^2}{(1 + \lambda d_{B,i})^2} - \alpha = 0 \ , \tag{3.34}$$

where $B = UD_BU^T$ and $\hat{a} - \hat{x} = U^T(a - x)$. It is solved using a Newton-Raphson scheme as mentioned earlier.

3.1.3 Performance and limitation of the method

The use of Kronecker products in our resolution for general matrices is not efficient and results in a complexity of $\mathcal{O}(n^5)$ using a Conjugate Gradient scheme for the resolution of (3.20). As a consequence this technique is only applicable to small systems. Still, one could investigate the improvements that might arise from the exploitation of the available structure. We investigated this for companion matrices for which the possible improvements could be particularly attractive. The result is an algorithm for polynomials that will be introduced in the next chapter.

We now provide two numerical examples that will illustrate the mixed results of the method in its current form. The first one is an easy example which will show that under favorable conditions the method behaves as expected. Our second example will be a run of our benchmark and will clearly demonstrate that our method does not always converge.

Stabilization in discrete-time Consider the normal matrix $A = U\Lambda U^*$ obtained from the diagonal matrix

$$\Lambda = \begin{bmatrix} 0.5+j & & & & & \\ & -1+\frac{1}{2}j & & & & \\ & & -\frac{2}{3}-\frac{1}{3}j & & & \\ & & & \frac{1}{3}-\frac{1}{2}j & & \\ & & & & 1-\frac{1}{3}j & \end{bmatrix}, \tag{3.35}$$

and some unitary matrix U. In the spectral space, since the eigenvalues are slightly unstable the expected optimal solution is the projection of the unstable eigenvalues of A on the unit circle which is the solution that we reached as represented in Figure 3.3a.

Stabilization in continuous-time Consider the matrix A given by (2.48) for $n = 5$, $\nu_0 = 0.1$:

$$A = \begin{bmatrix} & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \\ \nu_0 & & & & \end{bmatrix}.$$

The solution X_* that we reach achieves an objective value $\|A - X_*\| = 0.41$ far from the optimal value of 0.1. It should be emphasized that the solution obtained in this case is not even a local optimum. The reason of this failure is simple : the algorithm is not able to stay sufficiently away from the boundary even in some simple settings. Indeed, as the iterates near the boundary of the stable set, the size of the successive Dikin ellipsoids become smaller thereby reducing the step sizes. Let us take a simple case pictured in Figure 3.4. The iterates are clearly not converging to the local optimal solution even though the set has really nice local properties.

The observed limitations of the method call for a theoretical improvement of our method that would prevent the iterates from reaching the boundary of our admissible set too quickly. These improvements are the subject of our next section.

3.1.4 Starting point for general matrices

Constructing a starting point through a direct computation of the eigenvalues is an easy task. Three natural ways of creating a starting point are quickly described and some of them do not require the knowledge of the eigenvalues which makes them particularly attractive in terms of computation time.

Mirroring eigenvalues. With this technique one mirrors the unstable eigenvalues in the stable domain while keeping the stable eigenvalues untouched. This technique is particularly well suited for cases where the unstable eigenvalues are relatively close to the stability boundary. One can then hope that the eigenvalues of the solution are close to the original eigenvalues.

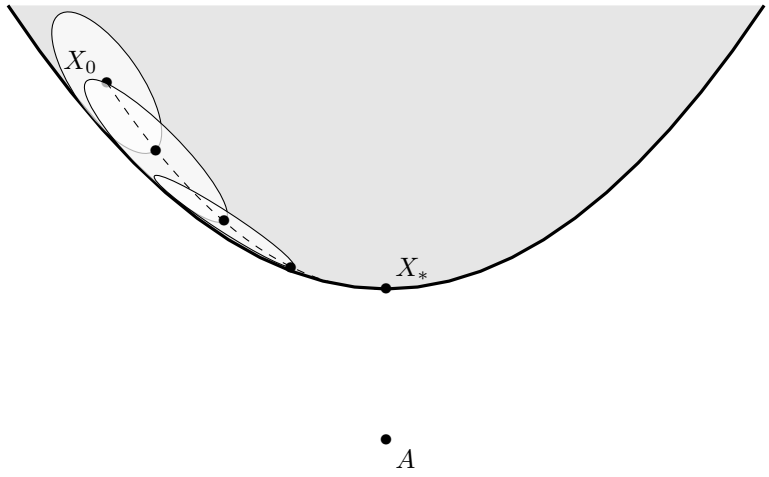


Figure 3.4: The behaviour of algorithm 3.1 does not always converge to a critical point even in some simple situations. Possible successive ellipsoidal approximations are represented and the solution gets quickly stuck on the boundary. The quality of the solution reached is highly dependent on the starting point.

Preserving the eigenvalues distribution. This technique is based on the translation (in continuous-time) or the contraction (in discrete-time) of all the eigenvalues of the matrix into the interior of the stable domain. The disadvantage of this technique is that it affects all the eigenvalues, unstable as well as stable. Its advantage is that if for some reason the computation of the eigenvalues is not an option, the technique is still applicable since it does not necessarily require the knowledge of the eigenvalues.

In discrete-time, a simple contraction of the eigenvalues suffices. As A is unstable it has at least one eigenvalue outside the unit disk. Hence its norm cannot be smaller than one. By taking

$$X_0 = \kappa \frac{A}{\|A\|_F} \ ,$$

we get a valid starting point as long as $0 < \kappa < 1$. If the value of the largest eigenvalue of A is known, one can replace $\|A\|$ by $\rho(A)$ in this last expression. One could work with the singular values of the matrix for the contraction as well.

In continuous-time, one only has to translate the matrix to do the job. Taking

$$X_0 = A - I(\|A\| + \kappa) \ ,$$

for some small positive κ will ensure a negative definite matrix with the same distribution of eigenvalues as A.

Preserving the skew-hermitian part of the matrix This method is probably the fastest one but is only applicable to continuous-time systems. The method is an application of Bendixson's lemma and implies that the eigenvalues of the starting point are all located on a vertical line at some user-defined distance from the stability boundary (in the spectral space). Given an unstable matrix A, we can compute the Hermitian and skew-Hermitian part of this matrix

$$A_H = \frac{A + A^*}{2} \quad A_S = \frac{A - A^*}{2} \ .$$

The skew-Hermitian part A_S has its eigenvalues on the imaginary axis. Hence a starting point is obtained by shifting the eigenvalues of A_S to the left. We thus have

$$X_0 = A_S + \delta I \ ,$$

where $\delta < 0$ is a user-defined parameter.

As a conclusion it is good to remind users that starting points that are obtained without a direct computation of the eigenvalues offer mixed results since their relative position with respect to the stability boundary is unknown.

3.2 Augmented method for the solution of the nearest stable system using successive convex approximations

The limitations observed for the method described in the previous section led us to consider an augmented version of our algorithm. Our goals are quite reasonable, we want to prevent the iterates from reaching the boundary of the stable set too quickly and we want the augmented version of our algorithm to provide a viable framework for some proof of convergence. Keeping the iterates away from the boundary while looking for the nearest stable matrix seems to be two contradictory goals since the solution that is sought will be on the boundary of the stable set unless the solution is the original matrix itself. But we will show that the first goal can be reached through the use of an auxiliary optimization problem which incorporates a barrier function. The second goal is then attained by showing that the solution of the auxiliary optimization problem converges to the desired solution for decreasing values of the barrier coefficient.

3.2.1 Construction of an auxiliary optimization problem

The objective function (3.1) is modified so as to incorporate a barrier term $\phi(.)$ in our objective function along with the constraint in $\langle P, I\rangle$ which bounds the feasible set of P. Note that this last constraint was already used in the previous section and does not change the optimal solution of the minimization problem.

Let $\mathcal{A}_P(X)$ be the Lyapunov operator in continuous- or discrete-time as before whose expression is

$$\begin{aligned} \mathcal{A}_P(X) &= XP + PX^* && \text{(continuous-time)} \\ \mathcal{A}_P(X) &= XPX^* - P \ . && \text{(discrete-time)} \end{aligned}$$

The inclusion of a barrier function in the formulation enables us to relax our constraints on the stable set. Indeed, the barrier function prevents the iterates from reaching the boundary of the stable set therefore the constraint $-\mathcal{A}_P(X) \succeq 0$ cannot be active. Furthermore, since $X \in \mathbb{S}^n$ and $-\mathcal{A}_P(X) \stackrel{\text{def}}{=} -\Psi_X(P) \succ 0$, Theorem 1.62 implies that P cannot be singular and we can relax the constraint $P \succ 0$.

The augmented version of the nearest stable matrix problem is

$$\begin{aligned} \min_{X,P} \quad & \frac{1}{2}\|X - A\|^2 + \mu\phi(-\mathcal{A}_P(X)) \\ s.t. \quad & -\mathcal{A}_P(X) \succeq 0 \\ & P \succeq 0 \ , \\ & \langle P, I \rangle = 1 \ . \end{aligned} \tag{3.36}$$

The self-concordant barrier $\phi(\cdot)$ could be any self-concordant function but the most natural choice is to use the *log-det* barrier that we have already used in order to build ellipsoidal convex approximations of our domain. With this choice (3.36) becomes

$$\begin{aligned} \min_{X,P} \quad & \frac{1}{2}\|X - A\|^2 + \mu b_P(X) \\ s.t. \quad & -\mathcal{A}_P(X) \succeq 0 \\ & P \succeq 0 \ , \\ & \langle P, I \rangle = 1 \ , \end{aligned} \tag{3.37}$$

where we recall that $b_P(X)$ is the self-concordant barrier defined by

$$b_P(X) = -\log\det(-\mathcal{A}_P(X)) \ .$$

We can apply our Barrier Projection Method to solve the minimization problem (3.37). Let us quickly review how the approach works. The Barrier Projection Method first builds a Dikin ellipsoid inside the stable set. The Dikin ellipsoid is described by (3.6) which we rewrite here for convenience

$$\mathcal{E}(X, P; \alpha) \stackrel{\text{def}}{=} X + \{H \in \mathbb{C}^{n\times n} \ : \ \langle b_P''(X)H, H\rangle \leq \alpha\} \ .$$

The choice of the Dikin ellipsoid fixes P. Indeed, the feasible set for P is described by

$$\mathbb{P}_X \stackrel{\text{def}}{=} \{P \in \mathbb{K}^n_{++} \ : \ \Psi_X(P) \prec 0, \quad \langle P, I\rangle = 1\},$$

and, we have explained in Section 3.1 that we take P as the analytic center of the set, i.e.

$$P = \arg\min_{\hat{P}} \{b_X(\hat{P}) \ : \ \hat{P} \in \mathbb{P}_X\}, \tag{3.38}$$

where $b_X(P)$ is the self-concordant barrier defined on the feasible set of P by

$$b_X(P) \stackrel{\text{def}}{=} f(-\Psi_X(P)), \qquad f(Y) = -\ln\det Y,$$

where $\Psi_X(P)$ is the continuous-time Lyapunov operator or the discrete-time Stein operator. The analytic center solution of (3.38) is also the solution of either the system given by the Lyapunov equations

$$X^*Q^{-1} + Q^{-1}X + nI = 0, \tag{3.39}$$
$$XP + PX^* + Q = 0, \tag{3.40}$$

in continuous-time or by the Stein equations

$$X^*Q^{-1}X - Q^{-1} + nI = 0, \tag{3.41}$$
$$XPX^* - P + Q = 0. \tag{3.42}$$

in discrete-time.

Once P is fixed, we can find the nearest stable matrix $X + H$ inside the Dikin ellipsoid by solving the convex optimization problem

$$\begin{array}{ll} \min\limits_{H} & \frac{1}{2}\|X + H - A\|^2 + \mu b_P(X + H) \\ s.t. & \langle b''_P(X)[H], H\rangle \leq \alpha. \end{array} \tag{3.43}$$

As it appears, the formulation (3.43) is very much similar to the formulation of the algorithm 3.1 we detailed for the BPM method. The only things we must change are the expressions of the various terms found in the first order optimality conditions of (3.43). Their general expression is given by

$$X + H - A + \mu b'_P(X + H) + \lambda b''_P(X)[H] = 0 \tag{3.44}$$
$$\frac{\lambda}{2}\left(\langle b''_P(X)[H], H\rangle - \alpha\right) = 0 \tag{3.45}$$
$$\lambda \geq 0\ .$$

In contrast with the previous section, developments for the continuous-time case and the discrete-time case have to be separate at this point.

3.2.2 Resolution of the auxiliary optimization problem

Continuous-time developments

The resolution of (3.44) and (3.45) requires the knowledge of the first and second order derivatives of $b_P(X)$ which we recall here.

Proposition 2. *The first and second order derivatives of $b_P(X)$ in continuous-time are given by*

$$b'_P(X) = 2Q^{-1}P\ , \tag{3.46}$$
$$b''_P(X)[H] = 2Q^{-1}(HP + PH^*)Q^{-1}P\ . \tag{3.47}$$

Furthermore, the first order approximation of $b'_P(X)$ in continuous-time is given by

$$b'_P(X + H) \approx b'_P(X) + b''_P(X)[H]\ . \tag{3.48}$$

Proof. The expressions (3.46) and (3.47) were proven in Section 3.1 in Theorem 3.3. We nonetheless reconstruct them here for the sake of clarity. The first and second order derivatives of $\mathcal{A}_P(X)$ are

$$\begin{aligned} D\mathcal{A}_P(X)[H] &= HP + PH^* \\ D^*\mathcal{A}_P(Y)[H] &= 2YP \\ D^{2,*}\mathcal{A}_P(X)[H,H] &= 0. \end{aligned}$$

The first and second order directional derivatives of $b_P(X)$ are thus

$$\begin{aligned} Db_P(X)[H] = \langle b'_P(X), H\rangle &= \langle D^*\mathcal{A}_P(X)[f'(-\mathcal{A}_P(X)], H\rangle \ , \\ &= \langle D^*\mathcal{A}_P(X)(Q^{-1}), H\rangle \end{aligned} \tag{3.49}$$

$$D^2b_P(X)[H,H] = \langle b''_P(X)[H], H\rangle = \langle D^*\mathcal{A}_P(X)[Q^{-1}D\mathcal{A}_P(X)[H]Q^{-1}], H\rangle \ . \tag{3.50}$$

The expression of $b'_P(X+H)$ is computed as the first order approximation of (3.49) which is

$$b'_P(X+H) = -D^*\mathcal{A}_P(X)\left[f'(-\mathcal{A}_P(X+H))\right] \ .$$

Since $\mathcal{A}_P(X)$ is a linear operator in continuous-time, $\mathcal{C}[H] = D\mathcal{A}_P(X)[H]$ is also a linear operator in H and we can write

$$\begin{aligned} b'_P(X+H) &= D^*\mathcal{A}_P(X)\left[(-\mathcal{A}_P(X) - \mathcal{A}_P(H))^{-1}\right] \\ &= D^*\mathcal{A}_P(X)\left[(Q - \mathcal{A}_P(H))^{-1}\right] \\ &= D^*\mathcal{A}_P(X)\left[Q^{-\frac{1}{2}}(I - Q^{-\frac{1}{2}}\mathcal{A}_P(H)Q^{-\frac{1}{2}})^{-1}Q^{-\frac{1}{2}}\right] \\ &\overset{(I-A)^{-1}\approx(I+A)}{=} D^*\mathcal{A}_P(X)\left[Q^{-1} + Q^{-1}\mathcal{A}_P(H)Q^{-1}\right] \ , \\ &= 2Q^{-1}P + 2Q^{-1}(HP + PH^*)Q^{-1})P \ , \end{aligned} \tag{3.51}$$

which proves (3.48). □

Adaptations for the discrete-time case

Dedicated developments are needed in discrete-time because Proposition 2 only holds for linear operators. Even though $\mathcal{A}_P(X)$ is quadratic in discrete-time, it can be transformed into a LMI for which all of the above is applicable. Let us define

$$\begin{aligned} M(X,P) \ &: \ \mathbb{C}^{n\times n} \times \mathbb{K}^n_+ \to \mathbb{C}^{2n\times 2n} \ : \ (X,P) \to \begin{pmatrix} P & XP \\ PX^* & P \end{pmatrix} \\ R(H,P) \ &: \ \mathbb{C}^{n\times n} \times \mathbb{K}^n_+ \to \mathbb{C}^{2n\times 2n} \ : \ (X,P) \to \begin{pmatrix} 0 & HP \\ PH^* & 0 \end{pmatrix} \ . \end{aligned} \tag{3.52}$$

We adapt the expression of our barrier function to

$$b_P(X) = \zeta(M(X,P)) \ , \tag{3.53}$$

where for a matrix $K \in \mathbb{K}^n_+$

$$\zeta(K) \; : \; \mathbb{C}^{2n\times 2n} \to \mathbb{R} \; : \; \zeta(K) = -\log\det(K) \; . \tag{3.54}$$

Then the optimal solution of the optimization problem given by

$$\begin{array}{rl} \min\limits_{X,P\in\mathbb{C}^{n\times n}} & \frac{1}{2}\|X-A\|^2 + \mu b_P(M(X,P)) \\ s.t. & M(X,P) \succeq 0 \; . \end{array} \tag{3.55}$$

is attained at a point (X_*, P_*) which is also solution of (3.37). Note that using the Schur complement, we can show that the condition $M(X,P) \succeq 0$ in (3.55) enforces both

$$P \succeq 0$$

and

$$P - XPX^* \succeq 0 \; .$$

The next developments follow the approach we have consistently adopted until now. We construct an iterative scheme, which consists in building successive Dikin ellipsoids which are convex approximations of the stable set of matrices described by $M(X,P) \succ 0$ and are strictly included in the stable set. This yields a convex optimization problem that looks exactly like (3.43) which we rewrite here for convenience

$$\begin{array}{ll} \min\limits_{H} & \frac{1}{2}\|X+H-A\|^2 + \mu b_P(X+H) \\ s.t. & \langle b'_P(X)[H], H\rangle \leq \alpha. \end{array} \tag{3.56}$$

Therefore our discrete-time reformulation is in accordance with the approach applied in continuous-time to (3.43). The only difference between (3.56) and (3.43) lies in the expressions taken by the gradient and the Hessian of $b_P(X)$. As the construction of the convex problem (3.56) fixes P through the construction of the Dikin ellipsoid and X is taken as the center of the ellipsoid, $H \in \mathbb{C}^{n\times n}$ is the only degree of freedom which is left in (3.56). Therefore, in accordance with our notations, we make implicit the dependency on P in our notations in the sequel and we use $M_P(X)$ and $R_P(H)$ instead of $M(X,P)$ and $R(H,P)$ whenever P is fixed. When $X \in \mathbb{S}^n$ is fixed as well, we further use $K = M(X) \in C^{2n\times 2n}$. We can now construct an expression for $b_P(X+H)$.

Proposition 3. *Given $X \in \mathbb{S}^n$, the gradient and the Hessian of $b_P(X)$ can be computed as*

$$b'_P(X) = -R^*_P(K^{-1}) \tag{3.57}$$

$$b''_P(X)[H] = R^*_P(K^{-1}R_P(H)K^{-1}), \qquad H \in \mathbb{C}^{n\times n} \; , \tag{3.58}$$

respectively. Furthermore we have

$$b'_P(X+H) \approx b'_P(X) + b''_P(X)[H] \; . \tag{3.59}$$

Proof. The first and second order derivatives of ζ as defined in (3.54) are, for $Y \in \mathbb{K}^{2n}_{++}$, $\zeta'(Y) = -Y^{-1}$ and $\zeta''(Y)[H] = Y^{-1}HY^{-1}$. We have

$$Db_P(X)[H] = \langle b'_P(X), H\rangle = \langle \zeta'(K), DM_P(X)[H]\rangle = \langle -K^{-1}, R_P(H)\rangle \ .$$

The second order derivative is obtained likewise. We have

$$\begin{aligned} D^2b_P(X)[H,H] = \langle b''_P(X)[H], H\rangle &= \langle \zeta''(K)[R_P(H)], R_P(H)\rangle \ , \\ &= \langle K^{-1}R_P(H)K^{-1}, R_P(H)\rangle \ . \end{aligned}$$

In the same way,

$$\begin{aligned} \langle b'_P(X+H), H\rangle &= \langle -(M_P(X+H))^{-1}, R_P(H)\rangle \\ &= \langle -(K+R_P(H))^{-1}, R_P(H)\rangle \\ &= \langle -K^{-\frac{1}{2}}(I+K^{-\frac{1}{2}}R_P(H)K^{-\frac{1}{2}})^{-1}K^{-\frac{1}{2}}, R_P(H)\rangle \\ &\approx \langle -K^{-\frac{1}{2}}(I-K^{-\frac{1}{2}}R_P(H)K^{-\frac{1}{2}})K^{-\frac{1}{2}}, R_P(H)\rangle \\ &= \langle -K^{-1}+K^{-1}R_P(H)K^{-1}, R_P(H)\rangle \end{aligned}$$

which concludes the demonstration. □

Proposition 4. *The gradient $b'_P(X)$ and the Hessian $b''_P(X)[H]$, are given by the expressions*

$$\begin{aligned} b'_P(X) &= 2\hat{K}_2P \ , \\ b''_P(X)[H] &= 2(\hat{K}_2PH^*\hat{K}_2 + \hat{K}_1HP\hat{K}_3)P \ , \end{aligned}$$

where

$$\begin{pmatrix} \hat{K}_1 & \hat{K}_2 \\ \hat{K}_2^* & \hat{K}_3 \end{pmatrix} = K^{-1} = \begin{pmatrix} Q^{-1} & -Q^{-1}X \\ -X^*Q^{-1} & P^{-1}+X^*Q^{-1}X \end{pmatrix} \ ,$$

and Q is defined as before as $Q = P - XPX^*$.

Proof. First of all, it can be checked by inspection that the given expression for K^{-1} is indeed the inverse of M and, consequently, that it is positive definite as well. Then, it is clear that the adjoint $R_P^*(\cdot)$ of $R_P(\cdot)$ must be such that $R_P^* : \mathbb{C}^{2n\times 2n} \to \mathbb{C}^{n\times n}$. If we now decompose the scalar product (3.57) we get

$$\begin{aligned} \langle K^{-1}, R_P(H)\rangle &= \langle K^{-1}, \begin{pmatrix} 0 & HP \\ PH^* & 0 \end{pmatrix}\rangle \\ &= 2\langle \hat{K}_2, HP\rangle \ . \end{aligned}$$

Using the definition of adjoint, this last result proves our expression of $b'_P(X)$. Now let us define

$$W = \begin{pmatrix} W_1 & W_2 \\ W_2^* & W_3 \end{pmatrix} = K^{-1}R_P(H)K^{-1} \ .$$

Note that our partition of W in blocks requires that the product $K^{-1}R_P(H)K^{-1}$ is Hermitian. As K^{-1} and $R_P(H)$ are both Hermitian, this condition is satisfied. Then it suffices to compute W_2 to conclude. □

Resolution of the first order optimality conditions

The use of the same notations in continuous- and in discrete-time for the barrier function $b_P(X)$ enables us to set a unified framework for the resolution of the first order optimality conditions (3.44) and (3.45). In both cases, using (3.48) in continuous-time and (3.59) in discrete-time, they finally become

$$X - A + \mu b'_P(X) + H + (\mu + \lambda) b''_P(X)[H] = 0 \tag{3.60}$$

$$\frac{\lambda}{2} \left(\langle b''_P(X)[H], H \rangle - \alpha \right) = 0 \tag{3.61}$$

$$\lambda \geq 0 \ .$$

The solution of (3.60) is dependent on the value of the Lagrange multiplier $\lambda \geq 0$. Let us suppose that $\lambda > 0$ for now. Then, the steps we have detailed to find λ in the previous section for the Barrier Projection Method can be applied to (3.60). Following (3.60), H can be expressed as a function of λ,

$$\operatorname{vec}(H(\lambda)) = \left(I + (\mu + \lambda) B \right)^{-1} w_\mu \ ,$$

where

$$w_\mu = \operatorname{vec}(A - X - \mu b'_P(X)) \ ,$$

and

$$B \in \mathbb{C}^{2n \times 2n} \ ,$$

is the matrix representation of the linear operator $b''_P(X)[H] \in \mathbb{C}^{n \times n}$ which is such that

$$B \operatorname{vec}(H) = \operatorname{vec}(b''_P(X)[H]) \ .$$

We can then plug $H(\lambda)$ in the complementarity condition (3.61) to obtain a univariate equation in λ

$$\psi_\mu(\lambda) = \sum_i \frac{d_{B,i} \hat{w}_{\mu,i}}{(1 + (\mu + \lambda) d_{B,i})^2} - \alpha = 0 \ , \tag{3.62}$$

where $U d_B U^T = B$ and $\hat{w}_\mu = U^T w_\mu$. From this point onward, the developments are analogous to those obtained in the previous section. Equation (3.62) is solved using a Newton scheme and yields the value of λ. Once λ is known, (3.60) amounts to a linear system of the form

$$\left(I + (\mu + \lambda) B \right) \operatorname{vec}(H) = w_\mu \ , \tag{3.63}$$

whose solution H_* is the solution of the convex problem (3.43). Finally and unlike the first version of the algorithm, the first order optimality condition (3.60) also admits a solution for the case $\lambda = 0$ which corresponds to a minimum inside the Dikin ellipsoid. This case is easily solved by setting $\lambda = 0$ in (3.60). The resulting equation is a linear system in H whose resolution follows the one of (3.63).

3.2.3 Algorithm and convergence proofs

The global optimal value of the auxiliary problem converges to true global optimal value

Let us demonstrate that the auxiliary optimization problem (3.37) given by

$$\begin{aligned} \min_{X,P} \quad & \frac{1}{2}\|X-A\|_F^2 + \mu b_P(X) \\ s.t. \quad & -\mathcal{A}_P(X) \succeq 0 \\ & P \succeq 0 \ , \\ & \langle P, I\rangle = 1 \ , \end{aligned} \tag{3.64}$$

can be used to solve the nearest stable system

$$\begin{aligned} \min_{X} \quad & \frac{1}{2}\|X-A\|_F^2 \\ s.t. \quad & X \in \mathrm{cl}(\mathbb{S}^n) \ . \end{aligned}$$

Our first statement follows the results of Theorem 1.3.2 in [73] which we rewrite here. Consider a vector space E and a subset $\mathcal{S} \subset E$ and suppose we have the functions $f : E \to \mathbb{R}$ and $\hat{g} : E \to \mathcal{S}$, $F : \mathcal{S} \to \mathbb{R}$ a self-concordant barrier function for $\mathcal{S}$ and a penalty coefficient $\mu \in \mathbb{R},\ \mu > 0$. Also suppose that we want to solve the optimization problem

$$\begin{aligned} \min_{x} \quad & f(x) + \mu F(\hat{g}(x)) \\ s.t. \quad & \hat{g}(x) \in \mathcal{S} \ . \end{aligned}$$

We introduce some useful notations : let f_* be the global optimal value of $f(x)$, let $\mathcal{Q}$ denote the feasible set, i.e. $\mathcal{Q} = \{x \mid \hat{g}(x) \in \mathcal{S}, x \in E\}$ and let us define $V_*(\mu) = \min_{x\in\mathcal{Q}} [f(x) + \mu F(\hat{g}(x))]$. We are now able to formulate our statement.

Theorem 3.4 ([73],Thm 1.3.2 (Adapted))**.** *If there exists a $\hat{\mu} > 0$ and a constant $\hat{F}$ such that*

$$f(x) + \hat{\mu} F(\hat{g}(x)) \geq \hat{F} \quad \forall x \in \mathrm{int}\,\mathcal{Q}, \tag{3.65}$$

then $\forall \mu \in [0, \hat{\mu}]$

$$\lim_{\mu\to 0} V_*(\mu) = f_* \ .$$

Proof. We will prove the two separate statements

1. $\lim_{\mu\to 0} V_*(\mu) \geq f_*$,
2. $\lim_{\mu\to 0} V_*(\mu) \leq f_*$.

Taken together these two statements prove the theorem. In order to prove our first statement, suppose that (3.65) is verified then

$$F(\hat{g}(x)) \geq \frac{1}{\hat{\mu}}(\hat{F} - f(x)) \qquad \forall x \in \mathrm{int}\,\mathcal{Q}.$$

For $\mu < \hat{\mu}$, we can thus write the inequalities

$$\begin{aligned} V_*(\mu) &\geq \min_{x \in \mathcal{Q}} \left\{ f(x) + \frac{\mu}{\hat{\mu}}(\hat{F} - f(x)) \right\} \\ &= \min_{x \in \mathcal{Q}} \left\{ (1 - \frac{\mu}{\hat{\mu}}) f(x) + \frac{\mu}{\hat{\mu}} \hat{F} \right\} \\ &= (1 - \frac{\mu}{\hat{\mu}}) f_* + \frac{\mu}{\hat{\mu}} \hat{F} \ . \end{aligned}$$

Taking the limit for $\mu \to 0$ yields the desired result. In order to prove that our second statement holds as well, we consider the inequality

$$V_*(\mu) \leq f(x) + \mu F(\hat{g}(x)) \qquad \forall x \in \operatorname{int} Q \ .$$

Therefore,

$$\lim_{\mu \to 0} V_*(\mu) \leq f(x) \qquad \forall x \in \operatorname{int} \mathcal{Q} \ ,$$

and thus

$$\lim_{\mu \to 0} V_*(\mu) \leq f_* \ .$$

□

Corollary 3.5. *The global solution of the auxiliary optimization problem (3.64) converges to the nearest stable matrix X_* to A when μ goes to* 0.

Proof. We are interested in applying Theorem 3.4 to the auxiliary optimization problem (3.64) obtained from our initial formulation (3.1) given by

$$\begin{aligned} \inf_{X,P} \quad & \frac{1}{2}\|X - A\|_F^2 \\ \text{s.t} \quad & \mathcal{A}_P(X) \prec 0 \\ & P = P^* \succ 0 \ . \end{aligned}$$

Let

$$f_\mu(X, P) = \ f(X) + \mu b_P(X) \tag{3.66}$$

be the objective function of (3.64) where

$$f(X) = \frac{1}{2}\|X - A\|_F^2. \tag{3.67}$$

It was observed in Section 3.1 that $b_P(X)$ is homogeneous in P and that the feasible set of P is convex. As a consequence, we can limit our search to a bounded convex set by adding the constraint

$$\langle P, I \rangle = 1 \ . \tag{3.68}$$

Though adding the constraint (3.68) changes the feasible set of the optimization problem, it does not change the optimal value of (3.1) nor its solution X_*, therefore the two problems are equivalent. With the additional constraint

(3.68), $b_P(X)$ is bounded in P. Then the only condition that must be verified to apply Theorem 3.4 on (3.66) is that

$$f_\mu(X,P) \geq \hat{F}$$

for some constant $\hat{F}$. Intuitively, it is clear that $b_P(X)$ alone is not bounded below but $f_\mu(X,P)$ is bounded in X (it is the sum of a quadratic function and a negatively signed logarithm). Let us now prove that this is indeed the case and that, as a consequence, $\hat{F}$ exists and is finite.

The lower bound $\hat{F}$ exists if the level sets of the function are bounded. In order to prove this last statement, we can see that for an appropriate constant C_0

$$\frac{1}{2}\|A\|_F^2 - \langle A, X\rangle + \frac{1}{2}\|X\|_F^2 - \mu \log\det(-XP - PX^*) = f_\mu(X,P) \leq C_0 \ .$$

Then we can compute the minimum of the function

$$\min_X \frac{1}{4}\|X\|_F^2 - \langle A, X\rangle - \mu \log\det(-XP - PX^*)$$

since it is convex and the first order optimality conditions are satisfied at some X and the minimum has some value $\underline{f}$. In order to prove that $\underline{f}$ is finite, let us bound the term which is dependent on the log det function. We have

$$\begin{aligned}
\log\det(-XP - PX^*) &= \sum_i \log\lambda_i(-XP - PX^*) \\
&\leq n \log\lambda_{max}(-XP - PX^*) \\
&\leq n \log\|XP + PX^*\| \\
&\leq n \log 2\|X\|\|P\| \ .
\end{aligned}$$

Since $P \in \mathbb{K}_+^n$ and $\langle P, I\rangle = 1$, we also have that $\|P\| \leq 1$. Therefore, we deduce that

$$\underline{f} \geq \min_X \frac{1}{4}\|X\|_F^2 - \langle A, X\rangle - n\mu \log 2\|X\| \ .$$

Finally, we conclude that

$$\|X\|_F \leq 2\sqrt{C_0 - \frac{1}{2}\|A\|_F^2 - \underline{f}} \ ,$$

and thus $\hat{F}$ exists and is finite (and the solution (X_*, P_*) exists and is finite as well). □

In practice, it was observed that the method does not converge well as μ goes to zero and that other techniques ought to be preferred. Our approach is to fix a minimal value for $\mu_K > 0$ for which we can also prove convergence to a stationary point of the auxiliary problem (3.64).

Algorithm 3.2 Augmented Barrier Projection Method (ABPM)

Choose a stable matrix X_0, a tolerance $\epsilon_{tol} > 0$ and $\mu_K > 0$;
Find P_0 and Q_0 by solving (3.39)-(3.42);
for $k \geq 0$ **do**

(a) Solve (3.62) to get λ_k;
(b) Determine H_k using (3.63);
(c) $X_{k+1} = X_k + H_k$ is the solution of (3.43);
(d) Find P_{k+1} and Q_{k+1} by solving (3.39)-(3.42);
(e) **If** $f_{\mu_K}(X_k, P_k) - f_{\mu_K}(X_{k+1}, P_{k+1}) < \epsilon_{tol}$ **then** return **end**
end

Algorithm and convergence proof

We now present the augmented version of the algorithm 3.1 in algorithm 3.2 before proving its convergence. An initial stable point X_0 for the algorithm 3.2 can be found by manipulating the eigenvalues of the matrix, but a simpler techniques is provided by Bendixson's lemma. See Section 3.1.4 for a discussion on available techniques to obtain a starting point.

Let us now state our second convergence result and let us analyse the convergence of the Augmented Barrier Projection Method, given by algorithm 3.2 applied to the formulation

$$\begin{array}{ll} \min\limits_{X,P} & \left[f_{1/\mu} = \dfrac{1}{2\mu}\|X - A\|_F^2 + b_P(X) + n(\langle P, I\rangle - 1) \right] \\ s.t. & -\mathcal{A}_P(X) \succeq 0 \\ & P \succeq 0 \ , \end{array} \tag{3.69}$$

where the whole objective function was divided by μ compared to the formulation (3.64) and the constraint $\langle P, I\rangle$ was lifted into the objective function. We have already proved in Theorem 3.2 that the Lagrange multiplier λ associated with the constraint $\langle P, I\rangle$ had an optimal value $\lambda = n$. Therefore, the two formulations are equivalent since dividing the objective function by μ does not change the solution itself. Moreover, in contrast with the formulation (3.64), the self-concordant properties of the objective function of (3.69) are preserved as $\mu \to 0$.

We have briefly introduced the importance of self-concordant functions in the first chapter. In particular, we mentioned the possibility of constructing an affine-invariant metric. We now introduce this metric and state important results associated with self-concordant functions. As this will require the introduction of new notations for the norms and in order to avoid confusion, we explicitly denote the Frobenius norm by $\|\cdot\|_F$ in the remainder of the section.

Definition 3.6 ([73]). *Let $F(x)$ denote a self-concordant function on* $\operatorname{dom} F$

and let F' and F'' denote the gradient and the Hessian of F . Then, we define

$$\begin{aligned}\|w\|_x &= \langle (F''(x))[w], w\rangle \\ \|w\|_{x,*} &= \langle (F''(x))^{-1}[w], w\rangle \ , \\ \omega(t) &= t - ln(1+t) = \frac{t^2}{2} - \frac{t^3}{3} + \cdots \ .\end{aligned}$$

Proposition 5 ([73, Theorem 4.1.7])**.** *For any $x, y \in \operatorname{dom} F$, we have*

$$F(y) \geq F(x) + \langle F'(x), y - x\rangle + \omega(\|y - x\|_x) \ .$$

Proposition 6 ([73, Theorem 4.1.10])**.** *Let $x_0 \in \operatorname{dom} F$ be given, and let the damped Newton step be defined by*

$$x_{k+1} = x_k - \frac{1}{1 + \|F'(x_k)\|_{x,*}} (F''(x_k))^{-1} F'(x_k), \ k \geq 0 \ .$$

Then one step performed by the damped Newton method achieves a monotonic decrease,

$$F(x_k) - F(x_{k+1}) \geq \omega(\|F'(x_k)\|_{x_k,*}) \ . \tag{3.70}$$

Moreover,

$$\|x_{k+1} - x_k\|_x < 1 \ .$$

We will now prove the convergence of our algorithm applied to (3.69) but the proof holds for (3.64) as well. We prove several results that are needed for the convergence of the algorithm.

Theorem 3.7. *Let P_+ be the analytic center generated by the log-det barrier function over the set*

$$\mathbb{P}_X = \{P \in \mathbb{K}^n_+ \mid \Psi_X(P) \prec 0, \ \langle P, I\rangle = 1\}, \ X \in \mathbb{S}^n \ .$$

Then

$$\nabla_P f_{1/\mu}(X, P_+) = 0 \ , \tag{3.71}$$

$$f_{1/\mu}(X, P) - f_{1/\mu}(X, P_+) \geq \omega(\|P - P_+\|^2_{P_+}) \ . \tag{3.72}$$

Proof. The analytic center is computed as the solution of the minimization problem

$$\min_{P \in \mathbb{P}_X} \ [b_X(P) = -\log\det(-\Psi_X(P))] \ . \tag{3.73}$$

The Lagrangian of the problem is

$$b_X(P) + \lambda(\langle P, I\rangle - 1) \ , \tag{3.74}$$

where $\lambda \in \mathbb{R}$ is a Lagrange multiplier. Note that (3.74) does not include the constraint $P \succeq 0$. Indeed, a positive definite P is automatically verified since X is stable and the barrier term guarantees that the Lyapunov constraint is positive definite. The first order optimality condition applied to (3.74) are thus

$$b'_X(P) + \lambda I = 0 \ . \tag{3.75}$$

Since this first order optimality condition corresponds exactly to the one obtained from (3.69) for P, the update P_+ is such that

$$\nabla_P f_{1/\mu}(X, P_+) = 0 \ .$$

This proves (3.71).

In order to prove (3.72), we can use Proposition 5 which yields

$$b_X(P) \geq b_X(P_+) + \langle b'_X(P_+), P - P_+ \rangle + \omega(\|P - P_+\|^2_{P_+}) \ . \tag{3.76}$$

The first order term of (3.76) is zero since following (3.75) we have that

$$\begin{aligned} \langle b'_X(P_+), P - P_+ \rangle &= -\lambda \langle I, P - P_+ \rangle \\ &= -\lambda(\langle I, P \rangle - \langle I, P_+ \rangle) \ , \end{aligned}$$

and $\langle P, I \rangle = 1, \ \forall P \in \mathbb{P}_X$. It suffices to note that $f_{1/\mu}(X, P) - f_{1/\mu}(X, P_+) = b_X(P) - b_X(P_+)$ to conclude. □

Let us now prove that a step in X achieves a sufficient decrease.

Theorem 3.8. *Let $X \in \mathbb{S}^n$, $P \in \mathbb{P}_X$ and $\mu > 0$ be given. Then for $H \in \mathbb{C}^{n \times n}$ equal to the solution of*

$$\begin{aligned} \min_H \quad & \frac{1}{2\mu} \|X + H - A\|^2_F + b_P(X + H) \\ s.t. \quad & \langle b''_P(X)[H], H \rangle \leq \alpha. \end{aligned} \tag{3.77}$$

the update $X_+ = X + H$ achieves a monotonic decrease, i.e.

$$f_{1/\mu}(X, P) - f_{1/\mu}(X_+, P) \geq \omega(\|\nabla_X f_{1/\mu}(X, P)\|^2_{X,*}) \ . \tag{3.78}$$

Proof. Let us prove that one step of the ABPM in X does at least as well as the damped Newton method. By Proposition 6, the update X_+ obtained by one step of the damped Newton method applied to $f_{1/\mu}(X, P)$ is contained in the Dikin ellipsoid defined from $f_{1/\mu}(X, P)$ which is

$$\langle \nabla^2_X f_{1/\mu}(X, P)[X_+ - X], X_+ - X \rangle < 1 \ .$$

If we decompose the Hessian $\nabla^2 f_{1/\mu}(X, P)$, we get

$$\langle \nabla^2_X f_{1/\mu}(X, P)[X_+ - X], X_+ - X \rangle = \langle (\tfrac{1}{\mu} I + b''_P(X))[X_+ - X], X_+ - X \rangle \ .$$

Therefore, for any $H \in \mathbb{C}^{n \times n}$ such that

$$\langle \nabla^2_X f_{1/\mu}(X, P)[H], H \rangle < 1 \ , \tag{3.79}$$

we have

$$\langle b''_P(X)[H], H \rangle < 1 \ . \tag{3.80}$$

This last implication shows that the steps computed from the minimization problem (3.77) are at least as large as the ones performed by the damped Newton method since the Dikin ellipsoid (3.79) defined by $f_{1/\mu}(X, P)$ is contained

in the Dikin ellipsoid (3.80) defined from the barrier function $b_P(X)$. Hence, each step performed by the minimization problem (3.77) achieves at least a monotonic decrease, i.e.

$$f_{1/\mu}(X,P) - f_{1/\mu}(X_+,P) \geq \omega(\|\nabla_X f_{1/\mu}(X,P)\|^2_{X,*}) \ .$$

□

We now have everything we need to prove convergence of our algorithm.

Theorem 3.9. *Given a matrix A, a stable matrix X_0 and parameters $\mu > 0$ and $0 < \alpha < 1$, the algorithm 3.2 converges to a stationary point of the optimization problem (3.69).*

Proof. By equation (3.72) in Theorem 3.7 we know that

$$f_{1/\mu}(X_{k+1},P_{k+1}) \leq f_{1/\mu}(X_{k+1},P_k) \ .$$

Therefore, from (3.78) in Theorem 3.8 we get

$$\begin{aligned} f_{1/\mu}(X_k,P_k) - f_{1/\mu}(X_{k+1},P_{k+1}) &\geq f_{1/\mu}(X_k,P_k) - f_{1/\mu}(X_{k+1},P_k) \\ &\geq \omega(\|\nabla_X f_{1/\mu}(X_k,P_k)\|^2_{X_k,*}) \\ &= \omega(\|\nabla f_{1/\mu}(X_k,P_k)\|^2_{X_k,*}) \end{aligned}$$

where the last equality is justified by the fact that the gradient $\nabla_P f_{1/\mu}(X_k,P_k)$ is zero as proved in Theorem 3.7. As a consequence and as the level sets are bounded, the Augmented Barrier Projection Method (ABPM) described by algorithm 3.2 converges monotonically to a point (X_*,P_*) which satisfies the condition

$$\langle \tfrac{1}{\mu} I + b''_{P_*}(X_*))^{-1} \nabla f_{1/\mu}(X_*,P_*), \nabla f_{1/\mu}(X_*,P_*) \rangle = 0 \ .$$

As $\mu > 0$, we conclude that (X_*,P_*) satisfies the first order optimality condition

$$\nabla f_{1/\mu}(X_*,P_*) = 0 \ .$$

□

Now that we have proved the convergence of our algorithm, we are interested in the distance to the optimal solution. For this purpose, we can switch back to our usual formulation of the minimization problem (3.64) which is

$$\begin{aligned} \min_{X,P} \quad & \frac{1}{2}\|X-A\|^2_F + \mu b_P(X) \\ s.t. \quad & -\mathcal{A}_P(X) \succeq 0 \\ & P \succeq 0 \ , \\ & \langle P, I \rangle = 1 \ . \end{aligned} \tag{3.81}$$

Theorem 3.10 (Distance to optimal value)**.** *Let $\mu > 0$ be fixed and let $f_{**} = \frac{1}{2}\|A - X_{**}\|_F^2$, $X_{**} \in \mathrm{cl}(\mathbb{S}^n)$ be the optimal value of (3.81) . Denote by ν the barrier parameter associated with the self-concordant barrier $b_P(X)$. If X_* is a minimizer of $f_{\mu_K}(X)$ given by (3.66) then if X_* and X_{**} have P_* as a common Lyapunov certificate*

$$0 < f_* - f_{**} \leq \mu_K \nu \ . \tag{3.82}$$

Proof. On the first hand, if X_* is a minimizer of $f_\mu(X)$ then by the necessary first order optimality condition for the convex set $\mathbb{S}_{P_*}$ we have

$$\langle \nabla_X f_\mu(X_*, P_*), X - X_* \rangle = \langle \nabla f(X_*) + \mu_K b'_{P_*}(X_*), X - X_* \rangle \geq 0 \ \forall X \in \mathbb{S}_{P_*} \ . \tag{3.83}$$

On the second hand, the Taylor series of f yields

$$f_{**} \geq f(X_*) + \langle \nabla f(X_*), X_{**} - X_* \rangle \ . \tag{3.84}$$

Plugging (3.83) in (3.84) gives

$$f_* - f_{**} \leq \langle \mu b'_{P_*}(X_*), X_{**} - X_* \rangle \ .$$

Finally, let us consider the domain of the barrier function

$$\mathbb{S}_{P_*} = \{X \in \mathbb{C}^{n\times n} \mid -\mathcal{A}_{P_*}(X) \succ 0\} \subset \mathbb{S}^n \ ,$$

then by [73, Theorem 4.2.4]

$$\langle b'_{P_*}(\hat{X}), X - \hat{X} \rangle \leq \nu \qquad \forall X \in \mathbb{S}_{P_*}, \ \hat{X} \in \mathbb{S}_{P_*} \ , \ \hat{X} \text{ fixed},$$

which concludes the demonstration. □

Note that Theorem 3.10 requires that X_{**} be in the set $\mathbb{S}_{P_*}$ obtained from the optimal P_*. In practice, it is quite unlikely that this is verifiable.

Strategies for μ

We have shown that the algorithm 3.2 and the convergence proof that ensued worked for a constant $\mu = \mu_K > 0$ chosen at the beginning of the algorithm. Actually, the proof still holds if we build a finite decreasing sequence $\mu_0, \ldots, \mu_K > 0$. Indeed, when the value μ_K is reached after K steps, the iterate X_K can be taken as the starting point in algorithm 3.2 and convergence follows. This question is important since the adopted strategy will influence the number of steps required before convergence.

The remainder of this section is dedicated to the construction of the sequence of barrier coefficients μ_j and to the appropriate timing to change them. We start by discussing an appropriate value for μ_K before developing these points.

A suitable choice of μ_K satisfies a number of criteria. It should not be too small to avoid small steps and it should balance the contribution of the barrier term in the objective function. A clever choice is to link the value of μ_K to the conditioning of the problem. For example, one can take

$$\mu_K = \epsilon_\mu \epsilon_{tol} \frac{\|A - X_0\|_F^2}{b_P(X_0)} \ , \tag{3.85}$$

where ϵ_{tol} is the tolerance used for the stopping criterion and ϵ_μ is some small constant. We usually take $\epsilon_\mu = 10^{-3}$. The solution (3.85) avoids the pitfall that would follow from a fixed predefined value of μ_K.

Once μ_K is chosen we can construct the sequence of coefficients μ_j. The strategy that we suggest is borrowed from convex optimization theory. In convex optimization a good value for μ_0 is given by

$$\mu_0 = \min(\frac{1}{\sqrt{n\lambda_{\min}^2(X_0)}}, \sqrt{n\lambda_{\min}^2(X_0)}) \ .$$

The sequence μ_j is then generated by repeating the instruction

$$\mu_{j+1} = \frac{\mu_j}{\beta} \ , \tag{3.86}$$

as much as needed, where $\beta = 1 + \frac{1}{\sqrt{n}}$.

Finally, let us discuss the appropriate moment to operate the switch from μ_j to μ_{j+1}. Again several choices can be considered and none of them can be dismissed but some will yield larger steps and thus faster convergence. The easiest choice is to decrease μ_j at each iteration using (3.86) but this will most likely favour steps in the direction of the boundary of the stable set and many small steps are thus to be expected to reach the optimum at the end. We made the choice to rather decrease the value of μ_j to μ_{j+1} only when the iterate X_k satisfies the solution

$$f_{\mu_j}(X_k) - f_{\mu_j}(X_{k+1}) < \epsilon_{tol}\gamma_\mu$$

where γ_μ is typically of the order of 10^1 when $\epsilon_{tol} = 10^{-6}$. In case ϵ_{tol} is really small one might prefer larger values of γ_μ to avoid unnecessary small steps before decreasing μ_j. When this technique is used, one has to prevent the algorithm from stopping before the final value μ_K is reached.

3.2.4 Numerical example

In order to validate the above theory we carry out the same experiment we tried in the last section. We consider the benchmark matrix A which is given by (2.48) which we rewrite here for convenience and for $n = 5$

$$A = \begin{bmatrix} 0 & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \\ \nu_0 & & & & 0 \end{bmatrix}$$

with $\nu_0 = 0.1$. The conjectured solution is given by X_* which corresponds to A everywhere except for the lower left corner of X_* which is 0 instead of ν_0 as detailed in Section 2.3.3. Therefore the optimal objective value is given by $\|A - X_*\|_F = \nu_0$. The algorithm now reaches a solution X_f, depicted in Figure 3.5, for which

$$\|A - X_f\|_F = 0.1173,$$

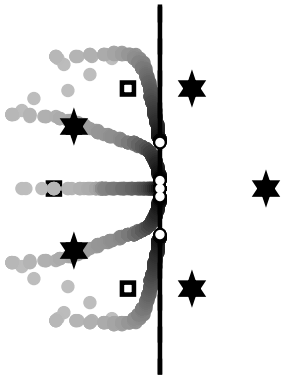


Figure 3.5: Eigenvalues of the nearest stable matrix to the unstable matrix given by (2.48). Stars represent the position of the original eigenvalues; black squares, the eigenvalues of the starting point; grey scaled circles show the evolution of the iterates; white dots show the eigenvalues of the final solution.

while the solution reached by the Barrier Projection Method in the previous section was at a distance 0.41 from A. The gain achieved by the augmented version of the algorithm is thus really substantial.

3.3 Gradient method for the nearest continuous-time stable matrix

The essential idea explored in this part of the manuscript is to use a reformulation of the nearest stable system in which the nonconvexity of the problem is shifted from the constraint into the objective function. Dealing with an optimization problem with nonconvex objective function and convex domain will be much easier in this particular case.

Because the continuous-time Lyapunov constraint is linear both in X and P, the continuous-time case offers a much more suitable framework than the discrete-time case. Therefore, the work presented here concentrates on continuous-time matrices.

The work described here starts from the most general form of the nearest stable matrix problem using Lyapunov constraints which was already presented in the two first chapters. Let the unstable matrix $A \in \mathbb{C}^{n\times n}$ be given. Then the problems is stated as follows

Problem P1

$$\begin{aligned} \inf_{X,P} \quad & \|A - X\|_F^2 \\ s.t \quad & XP + PX^* \prec 0 \\ & P = P^* \succ 0 \ , \end{aligned} \tag{3.87}$$

which is an optimization problem with convex objective function and nonconvex constraints.

In this section, we suggest formulations of the problem where the feasible sets are closed and for which marginally stable solutions are thus acceptable.

The work is organized as follows. We first explore our reformulation of (3.87) in Section 3.3.1 in which the nonconvexity of the Lyapunov constraint is moved to the objective function and a smoothing term is added. This last addition enables us to prove the existence of a solution to our reformulation. We then describe the gradient method used to solve the problem and its convergence properties in Section 3.3.2, followed by a description of the algorithm in Section 3.3.3. Finally, numerical experiments are carried out in Section 3.3.4 to show the effectiveness of the approach.

3.3.1 Reformulation of the nearest stable matrix problem

The matrix P in P1 (3.87) being positive definite, we can take $B = P^{-1}$ and $Y = -XP$ in our reformulation P1 (3.87) to obtain
Problem P2

$$\inf_{Y,B,\eta} \quad \|A + YB\|_F^2 \tag{3.88a}$$

$$Y + Y^* = 2\eta I \tag{3.88b}$$

$$B \succ 0 \ , \tag{3.88c}$$

$$\eta > 0 \ , \ \eta \in \mathbb{R} \ , \tag{3.88d}$$

where we made use of corollary 1.63 to transform the Lyapunov constraint in (3.87) into the equality (3.88b). The choice of a right-hand term equal to $2\eta I$ is purely arbitrary but it offers the advantage of being easy to handle since the solution in Y reduces to $Y = \eta I + jH$, where $H \in \mathbb{C}^{n\times n}$ is a Hermitian matrix. One could also choose to fix η to a constant value. This possibility was investigated first but proved to be less efficient than the suggested variation in practice.

Several reasons call for a transformation of the problem into a more suitable form: two of them are that we are currently looking for an infimum and that the variables are all possibly unbounded in the present formulation. We have decomposed the transformation in three stages. The first stage displays a first reformulation and shows that the solution of the problem is bounded; the second stage is designed to keep only two, easy to deal with, variables in the problem; the last stage consists in adding a constraint such that the domain is closed.

Boundedness Take $Y = \eta I + jH$ as a general solution of the optimization problem P2 (3.88) as suggested. Then let us consider the following reformulation of our problem P2 augmented with Tikhonov regularization terms in H, B and η

Problem P3

$$\min_{H,B,\eta} \quad \left[\|A+\eta B+jHB\|^2+\epsilon(\|H\|^2+\|B\|^2+\eta^2)=\psi_\epsilon(B,H,\eta)\right] \tag{3.89a}$$

$$B \succeq 0 \tag{3.89b}$$

$$H = H^* \ , \tag{3.89c}$$

$$\eta \geq 0 \ . \tag{3.89d}$$

Let us prove that the level sets of $\psi_\epsilon(B,H,\eta)$ are bounded. Bounded level sets will imply that the solution to (3.89) exists. Consider an appropriate constant C_0 and the set

$$\mathcal{C} = \{(B,H,\eta) | \psi_\epsilon(B,H,\eta) \leq C_0\} \ .$$

We need to prove that $\mathcal{C}$ is bounded, i.e. that for any $(B,H,\eta) \in \mathcal{C}$, $\|H\|$, $\|B\|$ and η are bounded. Bounded level sets will show that $\|HB\|$ and $\|\eta B\|$ are bounded as well and thus η and $\|B\|$ cannot be zero, justifying the relaxations (3.89b) of (3.88c) and (3.89d) of (3.88d).

From the objective function (3.89a) we get

$$\epsilon(\|H\|^2+\|B\|^2+\eta^2) \leq \psi_\epsilon(B,H,\eta) \leq C_0 \ .$$

Thus we have

$$\|H\| \leq \frac{1}{\sqrt{\epsilon}} C_0 \ , \tag{3.90}$$

$$\|B\| \leq \frac{1}{\sqrt{\epsilon}} C_0 \ , \tag{3.91}$$

$$\eta \leq \frac{1}{\sqrt{\epsilon}} C_0 \ . \tag{3.92}$$

The level sets of the function are thus bounded and the solution exists. Indeed, from (3.89) we deduce

$$\psi_\epsilon(B,H,\eta) - \mu \log\det(B) \geq \epsilon(\|H\|^2+\|B\|^2+\eta^2) - \mu \log\det(B) \ . \tag{3.93}$$

The inequality (3.93) makes it clear that the minimum of the right-hand term in (3.93) is attained at $(B_*(\mu), H_*, \eta_*) = (\sqrt{\frac{\mu}{2\epsilon}} I, 0, 0)$ which stays finite as $\mu \to 0$.

As an additional comment, note that when a fixed $\eta = 1$ is taken in (3.88) then the regularization term in $\epsilon\|B\|^2$ is unnecessary since $\|B\|$ is nonetheless bounded. Indeed, when we remove the regularization term $\epsilon\|B\|^2$ and the dependence in η from (3.89a) we get the expression

$$\begin{aligned} \min \quad & \zeta_\epsilon(B,H) = \|A+B\|^2 - 2\langle jA, HB\rangle + \|HB\|^2 + \epsilon\|H\|^2 \\ s.t \quad & H = H^* \\ & B \succeq 0 \ . \end{aligned} \tag{3.94}$$

We observe that (3.94) is minimal when the product HB is equal to $-jA$ which yields the bound

$$C_0 \geq \zeta_\epsilon(B,H) \geq \|A+B\|^2 - \|A\|^2 \ ,$$

which in turn yields

$$\|A+B\| \leq \sqrt{C_0 + \|A\|^2}\ .$$

Eventually, by making use of the identity $\|Y+Z\| \geq \|Y\| - \|Z\|$ we get

$$\|B\| \leq \sqrt{C_0 + \|A\|^2} + \|A\|\ . \tag{3.95}$$

Note that we could also prove that $\|HB\|$ is bounded with the formulation (3.94), but this will be of no use for our further developments.

The introduction of Tikhonov regularization terms into the formulation P3 (3.89) makes the level sets bounded and thus transforms the search for an infimum in P1 (3.87) and P2 (3.88) into a search for a minimum in P3 (3.89).

It is well-known that the use of the Tikhonov regularization guarantees the convergence of the global optimal solution $\psi_{\epsilon,*}$ of P3 (3.89) to the global infimum of P1 (3.88) for a decreasing sequence $\epsilon \to 0$. For the convenience of the reader, we make such a convergence proof available here.

Theorem 3.11. *Let $x \in E$ where E is some vector space and let the function $f(x) : E \to \mathbb{R}$ and let $Q \subset E$ and consider the minimization problem*

$$\min_{x \in Q} \qquad f_\epsilon(x) = f(x) + \epsilon\|x\|^2, \tag{3.96}$$

where ϵ is some real parameter which satisfies $\epsilon > 0$. Let us denote the global optimal solution of (3.96) by $f_{\epsilon,*}$ and also denote by f_* the global infimum of $f(x)$. Then

$$\lim_{\epsilon \to 0} f_{\epsilon,*} = f_*\ .$$

Proof. There are two cases, either f_* is finite or $f_* = -\infty$. Both can be proved from at most two statements. Indeed, if f_* exists and is finite we must prove the following statements.

1. $\lim_{\epsilon \to 0} f_{\epsilon,*} = \hat{f}_* \geq f_*$;
2. $\hat{f}_* \leq f_*$.

The second statement suffices to prove the case where $f_* = -\infty$. The first statement must hold since $\lim_{\epsilon \to 0} f_{\epsilon,*}$ is a decreasing sequence and $f_{\epsilon,*}$ is bounded below, therefore there exists a limit point $\hat{f}_*$. Moreover, for any $\epsilon > 0$, $f_{\epsilon,*} \geq f_*$.

Our second statement holds as well since

$$f_{\epsilon,*} \leq f(\hat{x}) + \epsilon\|\hat{x}\|, \qquad \forall \hat{x} \in Q$$

and thus

$$\lim_{\epsilon \to 0} f_{\epsilon,*} \leq f(\hat{x}) \qquad \forall \hat{x} \in Q\ .$$

□

Note that the proof of Theorem 3.11 does not require that a solution to $\inf_{x \in Q} f(x)$ actually exists, it only requires that $f(x)$ is such that a minimization problem can be stated.

Eliminating H Our goal is to transform the objective function (3.89a) so that we can write the optimality conditions in H more easily. We can thus transform our objective function given by (3.89a) into

$$\psi_\epsilon(B, H, \eta) = \|A + \eta B\|^2 + 2\langle A + \eta B, jHB\rangle + \|HB\|^2 + \epsilon(\|H\|^2 + \|B\|^2 + \eta^2) \ .$$

We can also modify the scalar product $\langle A + \eta B, jHB\rangle$ to isolate H. Taking into account that H is Hermitian we find by (1.8) and the fact that $\forall Z \in \mathbb{C}^{n\times n},\ \langle Z, H\rangle = \langle \frac{Z+Z^*}{2}, H\rangle$ that

$$2\langle A + \eta B, jHB\rangle = \langle -j(AB + \eta B^2) + j(BA^* + \eta B^2), H\rangle$$

which in turn yields

$$\psi_\epsilon(B, H, \eta) = \|A + \eta B\|^2 + \langle j(BA^* - AB), H\rangle + \langle B^2 + \epsilon I, H^2\rangle + \epsilon(\|B\|^2 + \eta^2)$$

The problem P3 given in (3.89) is thus equivalent to
Problem P4

$$\begin{aligned} \min_{H,B,\eta} \quad & \|A + \eta B\|^2 + \langle j[BA^* - AB], H\rangle + \langle B^2 + \epsilon I, H^2\rangle + \epsilon\|B\|^2 + \epsilon\eta^2 \quad (3.97)\\ & B \succeq 0 \\ & H = H^* \ , \\ & \eta \geq 0 \ . \end{aligned}$$

All the terms that appear in P4 are Hermitian. The first order optimality condition in H is

$$(B^2 + \epsilon I)H + H(B^2 + \epsilon I) = -j[BA^* - AB] \ .$$

At this point, we make use of the decomposition $B = U\Lambda U^*$ and define $\tilde{H} = U^*HU$ and $\tilde{A} = U^*AU$ to obtain

$$(\Lambda^2 + \epsilon I)\tilde{H} + \tilde{H}(\Lambda^2 + \epsilon I) = j[\tilde{A}\Lambda - \Lambda\tilde{A}^*] \ . \quad (3.98)$$

The solution $\tilde{H}$ is given by

$$\tilde{H}_\epsilon(U, \Lambda) = W_\epsilon(\Lambda) \odot \tilde{C}(U, \Lambda) \ , \quad (3.99)$$

where we have

$$\tilde{C}(U, \Lambda) = j[\tilde{A}\Lambda - \Lambda\tilde{A}^*] \ , \quad (3.100)$$

$$W_\epsilon(\Lambda)_{ij} = \tfrac{1}{\lambda_i^2 + \lambda_j^2 + 2\epsilon} \ , \quad (3.101)$$

and the subscript ϵ denotes a dependence on the parameter $\epsilon > 0$. It is worth noting that by (3.98)

$$\langle \Lambda^2 + \epsilon I, \tilde{H}_\epsilon^2\rangle = \frac{1}{2}\langle j[\tilde{A}\Lambda - \Lambda\tilde{A}^*], \tilde{H}_\epsilon\rangle = \frac{1}{2}\langle \tilde{C}, \tilde{H}_\epsilon\rangle \ . \quad (3.102)$$

Using the decomposition $B = U\Lambda U^*$, we can rewrite the objective function (3.97) as

$$\|A + \eta U\Lambda U^*\|^2 + \langle C, H\rangle + \langle U\Lambda U^* + \epsilon I, H\rangle + \epsilon\|B\|^2 + \epsilon\eta^2 \ .$$

As multiplication by a unitary matrix on the left and on the right inside a scalar product does not change the value of the scalar product, this last expression amounts to

$$\|\tilde{A} + \eta\Lambda\|^2 + \langle\tilde{C}, \tilde{H}\rangle + \langle\Lambda + \epsilon I, \tilde{H}\rangle + \epsilon\|\Lambda\|^2 + \epsilon\eta^2 \ . \tag{3.103}$$

Finally, plugging the expression for the solution $\tilde{H}_\epsilon$ given by (3.99) and the result (3.102) into (3.103), we get a fifth reformulation which is equivalent to P4.

Problem P5

$$\begin{array}{ll} \min\limits_{U,\Lambda,\eta} & \|\tilde{A} + \eta\Lambda\|^2 - \dfrac{1}{2}\langle\tilde{C}, \tilde{H}_\epsilon\rangle + \epsilon\|\Lambda\|^2 + \epsilon\eta^2 \\ s.t. & UU^* = I \\ & \Lambda \succeq 0 \ , \\ & \eta \geq 0 \ , \end{array} \tag{3.104}$$

where the dependence of the objective function on U is made implicit and $\tilde{H}_\epsilon$ and $\tilde{C}$ are given by (3.99) and (3.100) respectively.

Eliminating η It is clear that a closed form minimization in η is also possible. Indeed, all the suggested formulations until now, included P5 above, are quadratic functions of η. Therefore, from (3.104) we can write

$$\eta_* = \left[-\frac{\langle\tilde{A}, \Lambda\rangle}{\|\Lambda\|^2 + \epsilon}\right]_+ \ . \tag{3.105}$$

Plugging (3.105) in P5 (3.104) yields a sixth and final formulation

Problem P6

$$\begin{array}{ll} \min\limits_{U,\Lambda} & \|A\|^2 - \dfrac{\langle AU, U\Lambda\rangle_-^2}{\|\Lambda\|^2 + \epsilon} - \dfrac{1}{2}\langle C(U,\Lambda), H_\epsilon(U,\Lambda)\rangle + \epsilon\|\Lambda\|^2 \\ s.t. & UU^* = I \\ & \Lambda \succeq 0 \ , \end{array} \tag{3.106}$$

where $H_\epsilon = U\tilde{H}_\epsilon U^*$ and $C = U\tilde{C}U^*$.

Other possible reformulations Starting from the very first formulation of the nearest stable matrix problem given by (3.87), one can derive many formulations. In particular, the suppression of the variable η yields an optimization problem similar to P5 (3.104). In practice, a gradient method applied to P5 with $\eta = 1$, fixed, has proved to be less efficient than the formulation P6 given by (3.106).

Also, instead of eliminating H from the formulation P3 (3.89), one could write $B = U\Lambda U^*$ and eliminate Λ instead. The expression of $\Lambda(H, U)$ yields a simpler optimization problem but this approach only suppresses n variables from the original problem whereas eliminating H suppresses n^2 variables from the optimization.

An interesting variation of (3.89) is obtained if we use η as a parameter and fix it to a constant value (say $\eta = 1$). Then the only regularization term needed in P3 (3.89) is the term in $\|H\|^2$ since it can be proved that $\|B\|$ is bounded by (3.95). The formulation then becomes interesting if we constrain ourselves to real arithmetic. Indeed, we can augment the objective function with a barrier term that includes the constraint (3.89b). Restricting ourselves to real arithmetic then changes the problem (3.94) into

$$\begin{aligned} \min_{B, L_H} \quad & \|A + B + (L_H - L_H^T)B\|^2 + \epsilon\|L_H\|^2 - \mu \log\det(B) \ , \\ s.t. \quad & B = B^* \\ & L_H \text{ is a lower triangular matrix with zero diagonal,} \\ & B, L_H \in \mathbb{R}^{n\times n} \ . \end{aligned} \tag{3.107}$$

3.3.2 First order optimization scheme and convergence

We aim at solving a problem of the type

$$\min_{(U,\Lambda)\in\mathcal{B}\times\mathcal{D}} f(U, \Lambda) \ ,$$

where

$$\begin{aligned} \mathcal{B} &= \left\{U \in \mathbb{C}^{n\times n} \ : \ UU^* = I\right\} \ , \\ \mathcal{D} &= \left\{\Lambda \in \mathbb{R}^{n\times n} \ : \ \Lambda \text{ is diagonal, } \Lambda \succeq 0\right\} \ , \end{aligned}$$

and $f(U, \Lambda)$ is

$$f(U, \Lambda) = \|A\|^2 - \frac{\langle AU, U\Lambda\rangle^2}{\|\Lambda\|^2 + \epsilon} - \frac{1}{2}\langle W_\epsilon \odot C, C\rangle + \epsilon\|\Lambda\|^2 \ , \tag{3.108}$$

where the dependence of W_ϵ and C given by (3.101) and (3.100) on U and Λ is made implicit.

Before discussing the gradient method that we will apply to our minimization problem, let us recall our definitions of Lipschitz gradient that we introduced in the first chapter and their implications. We first introduce a norm which allows the definition of two different Lipschitz constants.

Definition 3.12. *Let $Z = (X, Y) \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$. We denote by $\|Z\|$ and $\|Z\|_*$ the norm and the dual norm of Z defined by*

$$\begin{aligned} \|Z\| \ &= \sqrt{\|X\|_F^2 + \kappa\|Y\|_F^2} \ , \\ \|Z\|_* &= \sqrt{\|X\|_F^2 + \frac{1}{\kappa}\|Y\|_F^2} \ . \end{aligned} \tag{3.109}$$

Proof. We can rewrite the norm $\|Z\|$ as

$$\|Z\| = \|(\|X\|_F, \sqrt{\kappa}\|Y\|_F)\|_2 \ .$$

The expression of $\|Z\|_*$ is then obtained as a consequence of Theorem 5.4.16 in [50]. □

Definition 3.13. *A function $f(Z) = h(X,Y) : \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n} \to \mathbb{R}$ has a Lipschitz continuous gradient on a convex set $\mathcal{Q}$ if there exists a positive constant L such that for any two points $Z = (X,Y)$ and $\hat{Z} = (\hat{X},\hat{Y}) \in \mathcal{Q}$*

$$\|\nabla f(Z) - \nabla f(\hat{Z})\|_* \leq L\|Z - \hat{Z}\| \ . \tag{3.110}$$

The use of a special norm and dual norm on $\mathcal{Q}$ in this definition is needed for the definition of two different Lipschitz constants as defined in the next properties. Indeed, following our definitions we can define two Lipschitz constants $L_1 = L$ and $L_2 = \kappa L$.

Property 3.14. *For a real-valued scalar function $h(X,Y)$ with Lipschitz continuous gradient, given any (X,Y) and $(\hat{X},\hat{Y}) \in \mathcal{Q}$ the following inequality holds :*

$$\begin{aligned} h(X,Y) \leq h(\hat{X},\hat{Y}) &+ \langle \nabla_X h(\hat{X},\hat{Y}), X - \hat{X}\rangle \\ &+ \langle \nabla_Y h(\hat{X},\hat{Y}), Y - \hat{Y}\rangle + \frac{L_1}{2}\|X - \hat{X}\|_F^2 + \frac{L_2}{2}\|Y - \hat{Y}\|_F^2 \ . \end{aligned}$$

Proof. This is a direct consequence of Lemma 1.2.3 in [73] applied to our definition of norms (3.109). However, for the convenience of the reader, we rewrite it here for our function $h(X,Y)$. Let $Z = (X,Y)$ and $\hat{Z} = (\hat{X},\hat{Y})$. Then $\forall Z,\ \hat{Z} \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$ we have

$$\begin{aligned} h(Z) &= h(\hat{Z}) + \int_0^1 \langle \nabla h(\hat{Z} + \tau(Z - \hat{Z})), Z - \hat{Z}\rangle \mathrm{d}\tau \\ &= h(\hat{Z}) + \langle \nabla h(\hat{Z}), Z - \hat{Z}\rangle \\ &\quad + \int_0^1 \langle \nabla h(\hat{Z} + \tau(Z - \hat{Z})) - \nabla h(\hat{Z})), Z - \hat{Z}\rangle \mathrm{d}\tau \ . \end{aligned}$$

Therefore, we can build an upper bound for the term $h(Z) - h(\hat{Z}) - \langle \nabla h(\hat{Z}), Z - \hat{Z}\rangle$ as follows.

$$\begin{aligned} &|h(Z) - h(\hat{Z}) - \langle \nabla h(\hat{Z}), Z - \hat{Z}\rangle| \\ &\qquad = \left| \int_0^1 \langle \nabla h(\hat{Z} + \tau(Z - \hat{Z})) - \nabla h(\hat{Z})), Z - \hat{Z}\rangle \mathrm{d}\tau \right| \\ &\qquad \leq \int_0^1 \left| \langle \nabla h(\hat{Z} + \tau(Z - \hat{Z})) - \nabla h(\hat{Z})), Z - \hat{Z}\rangle \right| \mathrm{d}\tau \ . \end{aligned}$$

Using our definitions of norm and dual norm (3.109), we get

$$\begin{aligned}
&\leq \int_0^1 \left\|\nabla h(\hat{Z} + \tau(Z - \hat{Z})) - \nabla h(\hat{Z}))\right\|_* \left\|Z - \hat{Z}\right\| \mathrm{d}\tau \\
&\leq \int_0^1 \tau L \left\|Z - \hat{Z}\right\|^2 \mathrm{d}\tau \\
&= \frac{L}{2}\left(\|X - \hat{X}\|_F^2 + \kappa\|Y - \hat{Y}\|_F^2\right)
\end{aligned}$$

The conclusion follows by taking $L_1 = L$ and $L_2 = \kappa L$. □

Now that we have defined our two Lipschitz constants L_1 and L_2, the notation $\|\cdot\|$ will refer exclusively to the Frobenius norm.

Let us discuss the domain of the function. The following definition applies to $\mathcal{B}$ with $k = n$:

Definition 3.15. *A set $\mathcal{C}$ is an equinormed set if*

$$\|Y\| = k \qquad \forall Y \in \mathcal{C} \ .$$

By definition an equinormed set cannot be convex but it is nonetheless compact. Given that $\mathcal{B}$ is not convex, we can take the convex hull $\operatorname{co}\mathcal{B}$ and apply the gradient mapping method developed in [73] and described in a slightly different but compatible way for equinormed sets in [34]. The gradient mapping can be applied on optimization problems with simple sets. We recall its construction and main properties here.

Definition 3.16. *Consider the sets $\mathcal{Q}_1$ and $\mathcal{Q}_2$, a function $h : \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n} \to \mathbb{R} : (X, Y) \to h(X, Y)$ with a Lipschitz continuous gradient with constants L_1 and L_2. Take $l_1 \geq L_1$, $l_2 \geq L_2$ and any $(\hat{X}, \hat{Y}) \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$. Then the gradient mappings $g_{\mathcal{Q}_1}(\hat{X}; l_1)$ and $g_{\mathcal{Q}_2}(\hat{Y}; l_2)$ of h at $(\hat{X}, \hat{Y})$ are defined as*

$$\begin{aligned}
g_{\mathcal{Q}_1}(\hat{X}; l_1) &= l_1(\hat{X} - X_{\mathcal{Q}}) \\
g_{\mathcal{Q}_2}(\hat{Y}\,; l_2) &= l_2(\hat{Y}\ - Y_{\mathcal{Q}}\,)
\end{aligned} \tag{3.111}$$

where $X_{\mathcal{Q}}$ and $Y_{\mathcal{Q}}$ are given by

$$\begin{aligned}
(X_{\mathcal{Q}}, Y_{\mathcal{Q}}) = \arg\min_{(X,Y)\in\mathcal{Q}=\mathcal{Q}_1\times\mathcal{Q}_2} \big[\phi(X, Y) &\stackrel{def}{=} h(\hat{X}, \hat{Y}) + \langle\nabla_X h(\hat{X}, \hat{Y}), X - \hat{X}\rangle \\
&+ \langle\nabla_Y h(\hat{X}, \hat{Y}), Y - \hat{Y}\rangle \\
&+ \frac{l_1}{2}\|X - \hat{X}\|^2 + \frac{l_2}{2}\|Y - \hat{Y}\|^2\big].
\end{aligned} \tag{3.112}$$

If the sets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are simple enough, we can solve (3.112) in closed form, and we can then apply the usual gradient method to our function using this definition. The following properties are derived from the definition of gradient mapping. Similar properties are found in [73] for functions with one argument.

Property 3.17. *Given two convex sets $\mathcal{Q}_1, \mathcal{Q}_2 \subseteq \mathbb{C}^{n\times n}$ and a function $h : \mathcal{Q}_1 \times \mathcal{Q}_2 \to \mathbb{R} : (X, Y) \to h(X, Y)$ with Lipschitz continuous gradient. Then for any $(\hat{X}, \hat{Y}) \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n}$ we have*

$$1. \quad \langle \nabla_X h(\hat{X}, \hat{Y}), X - X_{\mathcal{Q}} \rangle \geq \langle g_{\mathcal{Q}_1}, X - X_{\mathcal{Q}} \rangle \tag{3.113}$$

$$2. \quad \langle \nabla_Y h(\hat{X}, \hat{Y}), Y - Y_{\mathcal{Q}} \rangle \geq \langle g_{\mathcal{Q}_2}, Y - Y_{\mathcal{Q}} \rangle$$

$$3. \quad h(\hat{X}, \hat{Y}) - \phi(X_{\mathcal{Q}}, Y_{\mathcal{Q}}) \geq \frac{1}{2l_1}\|g_{\mathcal{Q}_1}\|^2 + \frac{1}{2l_2}\|g_{\mathcal{Q}_2}\|^2 \tag{3.114}$$

Proof. In order to prove the first two properties, note that the first order optimality condition at $(X_{\mathcal{Q}}, Y_{\mathcal{Q}})$ implies that

$$\begin{aligned}
\langle \nabla_X h(\hat{X}, \hat{Y}) - g_{\mathcal{Q}_1}, X - X_{\mathcal{Q}} \rangle &= \langle \nabla_X \phi(X_{\mathcal{Q}}, Y_{\mathcal{Q}}), X - X_{\mathcal{Q}} \rangle \geq 0 \\
\langle \nabla_Y h(\hat{X}, \hat{Y}) - g_{\mathcal{Q}_2}, Y - Y_{\mathcal{Q}} \rangle &= \langle \nabla_Y \phi(X_{\mathcal{Q}}, Y_{\mathcal{Q}}), Y - Y_{\mathcal{Q}} \rangle \geq 0 \, .
\end{aligned}$$

The proof of the third statement is given in [74] on which we apply our definition of norm (3.109),

$$\begin{aligned}
h(\hat{X}, \hat{Y}) - \phi(X_{\mathcal{Q}}, Y_{\mathcal{Q}}) &= \phi(\hat{X}, \hat{Y}) - \phi(X_{\mathcal{Q}}, Y_{\mathcal{Q}}) \\
&\geq \frac{l_1}{2}\|\hat{X} - X_{\mathcal{Q}}\|^2 + \frac{l_2}{2}\|\hat{Y} - Y_{\mathcal{Q}}\|^2 \\
&= \frac{1}{2l_1}\|g_{\mathcal{Q}_1}\|^2 + \frac{1}{2l_2}\|g_{\mathcal{Q}_2}\|^2 \, .
\end{aligned}$$

□

Property 3.18. *If $(l_1, l_2) \geq (L_1, L_2)$, we have*

$$h(X_{\mathcal{Q}}, Y_{\mathcal{Q}}) \leq \phi(X_{\mathcal{Q}}, Y_{\mathcal{Q}}) \, .$$

Proof. As h has a gradient which is Lipschitz continuous, it is possible to show that for (X, Y) and $(\hat{X}, \hat{Y}) \in \mathcal{Q}$, we have

$$\begin{aligned}
|h(X, Y) - h(\hat{X}, \hat{Y}) - \langle \nabla_X h(\hat{X}, \hat{Y}), X - \hat{X} \rangle - \langle \nabla_Y h(\hat{X}, \hat{Y}), Y - \hat{Y} \rangle| \\
\leq \frac{L_1}{2}\|X - \hat{X}\|^2 + \frac{L_2}{2}\|Y - \hat{Y}\|^2 \, ,
\end{aligned}$$

see Lemma 1.2.3 in [73] for a proof. We thus have

$$h(X, Y) \leq \phi(X, Y) \, .$$

□

These last properties are helpful to show that the gradient mapping provides us with a first order method.

Theorem 3.19. *Given two simple convex compact sets $\mathcal{Q}_1$ and $\mathcal{Q}_2 \subseteq \mathbb{C}^{n\times n}$ and a function $h : \mathcal{Q} = \mathcal{Q}_1 \times \mathcal{Q}_2 \to \mathbb{R}$ bounded below by h^* on $\mathcal{Q}$ and having a Lipschitz continuous gradient with constant L_1 and L_2. Then if $l_1 \geq L_1$, $l_2 \geq L_2$ and the solution to (3.112) is computable, the step*

$$(X_{k+1}, Y_{k+1}) = (X_{\mathcal{Q}}, Y_{\mathcal{Q}}) \, , \tag{3.115}$$

converges to a local optimal solution.

Proof. Equation (3.115) derives from the usual gradient step

$$(X_{k+1}, Y_{k+1}) = (X_k, Y_k) - (\frac{1}{l_1} g_{\mathcal{Q}_1}(X_k; l_1), \frac{1}{l_2} g_{\mathcal{Q}_2}(Y_k; l_2)) \ . \tag{3.116}$$

Let us now rewrite (3.114) in the case of our gradient step (3.116),

$$\begin{aligned} h(X_k, Y_k) - h(X_{k+1}, Y_{k+1}) &\geq h(X_k, Y_k) - \phi(X_Q, Y_Q) \\ &\geq \frac{1}{2l_1} \|g_{\mathcal{Q}_1}(X_k; l_1)\|^2 + \frac{1}{2l_2} \|g_{\mathcal{Q}_2}(Y_k; l_2)\|^2 \ . \end{aligned}$$

The function h being bounded below we get

$$h(X_0, Y_0) - h^* \geq \lim_{k \to \infty} \sum_{m=0}^{k} \left[\frac{1}{2l_1} \|g_{\mathcal{Q}_1}(X_m; l_1)\|^2 + \frac{1}{2l_2} \|g_{\mathcal{Q}_2}(Y_m; l_2)\|^2 \right] \ .$$

The compactness of $\mathcal{Q}_1$ and $\mathcal{Q}_2$ enables us to conclude

$$\begin{aligned} \|g_{\mathcal{Q}_1}(X_k; l_1)\|^2 &\to 0 \\ \|g_{\mathcal{Q}_2}(Y_k; l_2)\|^2 &\to 0 \ . \end{aligned}$$

□

The advantage of this algorithm is its low complexity. But this reduction in complexity only arises if the solution to (3.112) can be easily obtained, i.e. if the sets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are simple enough.

3.3.3 Explicit solutions for the gradient mappings

Now that we have described our optimization scheme and its convergence, let us look at the expression taken by the gradient mappings. By examination we see that everything is applicable with $\mathcal{Q}_1 = \operatorname{co} \mathcal{B}$ and $\mathcal{Q}_2 = \mathcal{D}$, since $\mathcal{B}$ is compact and $\mathcal{D}$ has compact level sets. Let us see how to get a closed form solution to the minimization problem (3.112).

Gradient mapping on $\mathcal{B}$ As $\mathcal{B}$ is an equinormed set, the following general result applies

Theorem 3.20. *If $\mathcal{Q}$ is an equinormed set, the expression of $X_{\mathcal{Q}}$ in (3.112) is obtained as the solution of a linear programming problem whose expression is*

$$X_{\mathcal{Q}} = P_{\mathcal{Q}}(l_1 \hat{X} - \nabla_X h(\hat{X}, \hat{Y})) \ ,$$

where $P_{\mathcal{Q}}(Z)$ is the retraction of Z on $\mathcal{Q}$.

Proof. Fixin $Y = \hat{Y}$, equation (3.112) simplifies to

$$\begin{aligned} X_{\mathcal{Q}} &= \arg\min_{X \in \mathcal{Q}} [\langle \nabla_X h(\hat{X}, \hat{Y}), X - \hat{X} \rangle + \frac{l_1}{2} \|X - \hat{X}\|^2] \\ &= \arg\min_{X \in \mathcal{Q}} [\langle \nabla_X h(\hat{X}, \hat{Y}) - l_1 \hat{X}, X \rangle - \langle \nabla_X h(\hat{X}, \hat{Y}), \hat{X} \rangle + l_1 n] \ , \end{aligned}$$

where the last equation is obtained by applying the definition of equinormed set on $\mathcal{Q}$. The minimum is reached in the direction $-(\nabla_X h(\hat{X}) - l_1 \hat{X})$. □

The retraction on $\mathcal{B}$ is not difficult to figure out. For a given matrix $Z \in \mathbb{C}^{n\times n}$, we have $P_{\mathcal{B}}(Z) = Z[Z^*Z]^{-\frac{1}{2}}$. Moreover, an additional simplification occurs if we compute the polar factor of Z

$$\begin{aligned} Z &= R\Sigma V^* \\ P_{\mathcal{Q}}(Z) &= RV^* \ . \end{aligned}$$

Such results were obtained in [34], and the gradient mapping in the variable U is now defined.

Gradient mapping on $\mathcal{D}$ The gradient mapping in Λ is obtained in a similar way. Let $\hat{\Lambda}$ be fixed, then the optimization of (3.112) reduces to

$$\begin{aligned} \min_{\Lambda} \ & \langle \underbrace{\nabla_\Lambda f(\hat{U},\hat{\Lambda}) - l_2\hat{\Lambda}}_{D_g}, \Lambda\rangle + \frac{l_2}{2}\|\Lambda\|^2 \\ s.t \ & \Lambda \succeq 0 \\ & \Lambda \text{ is diagonal.} \end{aligned} \tag{3.117}$$

Then the solution to (3.117) is

$$\Lambda = [-D_g/l_2]_+ \ . \tag{3.118}$$

The resolution of the gradient mappings in U and in Λ requires the computation of the gradient of the objective function (3.108) both in U and in Λ. The details for the computations of these gradients are given in Appendix 3.B.

Algorithm The results obtained enable us to express algorithm 3.3 for the resolution of (3.106).

In this algorithm, we used Property 3.18, p.107 to check the validity of our Lipschitz constant. This test originates from [74] and is an efficient alternative to the use of the definition (3.110) of Lipschitz continuous gradient.

Note that convergence test $\|\frac{g_{\mathcal{B}}}{l_1}\|^2 + \|\frac{g_{\mathcal{D}}}{l_2}\|^2 < \epsilon_{tol}$ is equivalent to checking the condition

$$l_1\|(U_i - U_{i+1})\|^2 + l_2\|(\Lambda_i - \Lambda_{i+1})\|^2 < \epsilon_{tol} \ .$$

The complexity of each iteration is $\mathcal{O}(n^3)$ due to the use of the SVD. Its convergence is monotonic.

Note that the scheme is separable (as the problem was) in the sense that the steps 1 to 3 on the one hand, and 4 to 6 on the other hand can be implemented separately. Hence if l_1 does not satisfy the Lipschitz condition, the step in Λ is still valid and can be kept. This approach does not result in big savings in computational time.

The nature of the information provided by the Lipschitz constants is intrinsically local. Hence the conditions $l_1 \geq L_1$ and $l_2 \geq L_2$ should be understood as local conditions and it is therefore appropriate to update the values of l_1 and/or

Algorithm 3.3 Gradient Mapping Method

Choose an initial U_0 and Λ_0 , take a small l_1 and l_2 and a tolerance ϵ_{tol} and let f, ϕ and $g_{\mathcal{Q}}$ be defined as in (3.108),(3.112) and (3.111) for some given set $\mathcal{Q}$ respectively. Then

1. $G = l_1 U_i - \nabla_U f(U_i, \Lambda_i)$;
2. Compute the SVD : $R\Sigma V^* = G$;
3. $U^+ = RV^*$;
4. Compute $D_g = \nabla_\Lambda f(U_i, \Lambda_i) - l_2\Lambda_i$;
5. $\Lambda^+ = [-D_g/l_2]_+$;
6. if $f(U^+, \Lambda^+) > \phi(U^+, \Lambda^+)$ then double the value of l_1 and l_2 and go to step (1);
7. if $f(U^+, \Lambda^+) < \phi(U^+, \Lambda^+)$ then $U_{i+1} = U^+$, $\Lambda_{i+1} = \Lambda^+$, if $\|\frac{g_{\mathcal{B}}}{l_1}\|^2 + \|\frac{g_{\mathcal{D}}}{l_2}\|^2 < \epsilon_{tol}$ then exit otherwise $i = i + 1$ and go to step (1).

l_2 only when necessary. The adopted approach suffices to guarantee convergence. For the same reasons, we also regularly decrease both Lipschitz constants. This yields improved step lengths if the function is sufficiently smooth and speeds up local convergence. In order to make sense, we also need to prove that the Lipschitz constant is bounded from above by a finite constant. We provide such a proof in Appendix 3.A.

Once the algorithm has converged, one is still interested in actually reconstructing the nearest stable matrix X_* to A. We can compute successively $\tilde{H}_{\epsilon,*}$ given by (3.99) and η_* given by (3.105) to obtain

$$X_* = -U_*(\eta_* I + j\tilde{H}_{\epsilon,*})\Lambda_* U_*^* \ .$$

In practice it is also often more convenient for the convergence of the algorithm 3.3 to adopt a path-following-like approach for the values of ϵ. We start from an initial value ϵ_0 and decrease it by a constant factor β_ϵ each time some convergence threshold is crossed until a minimal value ϵ_{low} is attained. Several choices are possible for the values of those parameters. Typically we take a

value ϵ_0 which is small compared to $\|\Lambda\|^2$. We adopt the values,

$$\begin{aligned} \epsilon_{tol} &= n^2\ 10^{-9} \\ \epsilon_0 &= \|\Lambda_0\|^2\ 10^{-3} \\ \beta_\epsilon &= \sqrt{n} \\ \epsilon_{low} &= 10^{-2}\epsilon_{tol} \end{aligned} \tag{3.119}$$

We conducted our numerical experiments with this choice of parameters.

3.3.4 Numerical example

We apply all of the above to the continuous-time benchmark introduced in Section 2.3.3.

The algorithm 3.3 was implemented using MATLAB®. We have initiated our algorithm with the parameters given in (3.119) and $l_1 = l_2 = 10^{-2}$. The starting point is obtained by processing the following steps : Fix a small $\delta > 0$,

1. Take $M = -\left(\frac{A+A^*}{2}\right)$,
2. $U_0^* D U_0 = M$,
3. $[D]_- = \delta$,
4. $\Lambda_0 = D/\operatorname{Tr}(D)$.

Our solution X_* is approximately

$$X_* = \begin{bmatrix} 0 & 0.9994 & 0 & 0 \\ -0.0120 & 0 & 1.0000 & 0 \\ 0 & -0.0132 & 0 & 1.0001 \\ 0.0001 & 0 & -0.0120 & 0 \end{bmatrix} ,$$

and the eigenvalues of X_* lie at $\pm j1.9\ 10^{-1}$, $-1.8\ 10^{-6}$ and $-2\ 10^{-12}$. Note that in order to present the matrix in a suitable way, we had to round the numbers after the 4^{th} decimal. The rounding process markedly affects the eigenvalues that we compute from the expression of X_*. We can avoid this caveat by enforcing a minimal distance to the stability boundary for the eigenvalues of X. This can be implemented through the Lyapunov operator, see [27]. The objective function reaches the value

$$\|A - X_*\|_F - \nu_0 = 2.4\,10^{-3} ,$$

computed in 2.5 seconds and 3016 iterations. The solution is shown in Figure 3.6.

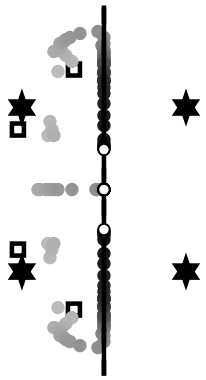


Figure 3.6: Trajectory of the eigenvalues of the successive solutions. Stars represent the eigenvalues of the original problem, squares the eigenvalues of the starting point, grey-scaled dots the eigenvalues of the intermediate solutions. The eigenvalues of X_* are given by the white dots.

3.4 Comparison of the performances

Two main methods were developed in this chapter, the Augmented Barrier Projection Method given by the algorithm 3.2 and the Gradient Mapping Method given by algorithm 3.3. Preliminary results show that the two methods achieve their respective goal in a fair amount of time for problems of small size.

The question we now want to answer is the following. How do our algorithms behave compared to other existing methods and for various sizes of the problems?

We chose to restrict our tests to continuous-time problems. Indeed, even though discrete-time problems offer an interesting variation of the problem, we prefer to focus on instances of the problem that all algorithms seen in this chapter can process. Our analysis will be extended to discrete-time instances of the problem in the chapter on polynomials where more additional discrete-time methods are available.

We first introduce the method with which we are going to compare our results, namely Hanso, before presenting the experiments. The discussion of our numerical results follows.

3.4.1 Hanso

Hanso is an optimization algorithm which includes a BFGS quasi-Newton method and gradient sampling methods. Hanso is primarily applicable to smooth convex problems but also works for nonsmooth or nonconvex problems, or both, in which case the main underlying hypothesis is that nonsmoothness occurs at a given point with probability zero and, in case it occurs, gradients at some nearby points suffice to provide acceptable search directions. The theory

is developed in [21, 63].

Because Hanso is not able to deal directly with constraints we add a penalty term to the objective function. The penalty term consists in the spectral abscissa function $\alpha(X)$ first encountered in the first chapter, Section 1.4.3.

The objective function used in Hanso is

$$\min_X \quad [f(X) = \|A - X\| + \mu\alpha(X)] \ , \tag{3.120}$$

whenever X is unstable and

$$\min_X \quad [f(X) = \|A - X\|] \ , \tag{3.121}$$

otherwise. One can note that we do not use the square norm in this case since numerical simulations tend to show that it is less efficient than the suggested formulation [78].

Being a first order method, Hanso requires the value and the gradient of the objective function evaluated at a point as inputs. The value of the objective function is given by (3.120) if $\alpha(X) > 0$ and by (3.121) otherwise. Accordingly, the gradients of (3.120) and (3.121) are, respectively,

$$\nabla f(X) = \frac{1}{\|A - X\|} X - A + \mu v u^* \ ,$$
$$\nabla f(X) = \frac{1}{\|A - X\|} X - A \ ,$$

where v (resp. u^*) is the right (resp. left) eigenvector associated with the active eigenvalue

$$\lambda_i(X) = \arg\max_{\lambda_j} \mathrm{Re}\,\{\lambda_j(X)\} \ .$$

See Section 1.4.3 for a justification of the expression of $\nabla\alpha(X)$.

Hanso has several advantages. It can work with stable as well as unstable matrices, it works from several starting points to retain the best solution only, and last but not least it displays acceptable convergence. On the other hand, its main weakness is that the solutions found are not necessarily stable. This might be corrected by taking some precautions such as displacing the stability boundary described by the spectral abscissa but this would in turn change the optimization problem.

3.4.2 Presentation of the experiments

We carried out two different types of experiments. We first tested the influence of the size of the problems on our algorithms. We then tried to evaluate how the performances of the algorithms were affected by the nonnormality of the matrices.

Two instances, one real and one complex, were generated for each type of experiment. As the presence of singularities on the boundary of the stable set are associated with active eigenvalues with multiplicity higher than one, a difficult instance is generated by the perturbation of such a matrix. This is not

a difficult case if the method reaches the point from the outside of the stable set but most of the algorithms we tested have iterates that are contained in the interior of the set. From this observation we consider the two following instances for our experiments

- Benchmark matrix,
- Random matrix.

Benchmark matrix Consider the benchmark we have already detailed in Chapter 2, Section 2.3.3. The unstable matrix is given by

$$A = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ \nu_0 & & & 0 \end{bmatrix} \tag{3.122}$$

For values of ν_0 such that $|\nu_0| < 1,\ \nu_0 \in \mathbb{R}$, a conjectured local optimal solution is given by the matrix

$$X_* = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ & & & 0 \end{bmatrix}$$

which has all its eigenvalues at zero. We took the value $\nu_0 = 0.1$ for the real instance of the problem and the value $\nu_0 = \frac{0.1+j0.3}{|0.1+j0.3|}$ for the complex instance. Since $|\nu_0| = 1$ in the complex case, several candidate solutions are easily available, one of which is obtained by projecting A onto the cone of negative definite matrices using the spectral norm. Whether all methods converge to the same solution has been an open question which will receive a partial answer.

Random matrix In contrast with the benchmark example, random matrices drawn from a Gaussian distribution of their coefficients should generally be easy to stabilize. Indeed, their eigenvalues have an uniform distribution on a disk centered in 0 and of radius approximately equal to the square root of the Frobenius norm of the matrix, and they do not have a particular structure. Because of these reasons, random matrices can be considered as easy examples.

The coefficients of real random matrices were generated from the Gaussian law $\mathcal{N}(0, 1)$. The real and the imaginary parts of the coefficients of complex random matrices were generated according to the law $\mathcal{N}(0, \frac{1}{2})$. Matrices were then scaled proportionally to the square root of their norm.

The resulting matrix A has approximately half of its eigenvalues that are unstable as shown in Figure 3.7.

In order to evaluate the impact of the nonnormality of the unstable matrix on the performance of the algorithms we conducted two additional experiments on matrices of fixed dimensions taken as $n = 8$.

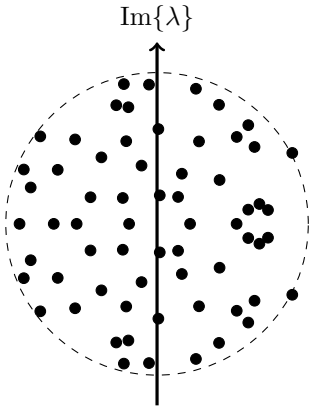


Figure 3.7: Typical eigenvalues distribution of a real random matrix of size $n = 64$ on both sides of the real axis. The unit circle is dashed.

Variable values of ν_0 The first additional experiment is created from a benchmark matrix of size $n = 8$ once more. For this numerical experiment, we created 10 benchmark matrices with the parameter ν_0 taken equidistantly in the the interval $[1, 10^{-2}]$. One will note that for $\nu_0 = 1$, the created benchmark matrix is in fact normal while for $\nu_0 = 10^{-2}$ the unstable matrix is very close to the conjectured solution located at a singularity of the stable set as discussed in chapter 2. The successive unstable matrices given in input to the algorithms are thus supposedly more difficult to solve for small values of ν_0.

Random nonnormal matrices The second experiment we created to measure the impact of the nonnormality on the performances of the algorithms consists in creating successive nonnormal random matrices A_i. Each nonnormal random matrix was created from two other random matrices R_i and T_i. The matrix R_i is a complex random matrix of size 8. The matrix $T_i \in \mathbb{R}^{8\times 8}$ is a random similarity matrix whose conditioning is taken as an input parameter and is generated using the *gallery* function of Matlab. We then instantiate the unstable matrix A_i using the expression

$$A_i = T_i R_i T_i^{-1} \ , \tag{3.123}$$

where i is the index of the experiment. Let $t(i)$ denote the conditioning of the matrix T_i. We carried out 13 experiments with values of $t(i)$ logarithmically spaced between $[1, 10^6]$. The results are reported in the next subsection.

Parametrization We applied three methods on these test cases

- Augmented Barrier Projection Method (ABPM),
- Gradient Mapping Method (GMM),

- Hanso (HANSO).

As each method has its own stopping criterion, a unified value for the tolerance/precision required for the solution could not be adopted. Several parametrizations of the methods were tested and we retained a different parametrization for each method which balances the time performance and the precision reached.

Hence, we have parametrized ABPM with a tolerance for the stopping criterion which is $\epsilon_{tol} = n^2\ 10^{-8}$. The parameters ϵ_μ and γ_μ related to the minimal value and the decrease of the barrier coefficient were taken as $\epsilon_\mu = n^2 10^{-10}$ and $\gamma_\mu = n^2 10$ so as to have a barrier coefficient which is lower than the prescribed tolerance for the stopping criterion. This is not required but it was observed to yield better results. The parametrization of GMM follows the same rule of thumb. The tolerance is fixed to $\epsilon_{tol} = n^2 10^{-9}$ and the minimal value of the smoothing parameter ϵ_{low} was fixed to 10^{-11}.

HANSO uses the default parametrization of the software. In short, Hanso either stops the BFGS phase when the algorithm converged with a tolerance of 10^{-4} for the smallest vector in the convex hull of the gradients at the current point but also at nearby points, or when the number of iterations reaches 1000 [63]. If this last case occurs, Hanso tries to improve the best point using gradient sampling techniques.

Starting points are obtained by mirroring the unstable eigenvalues of the matrix inside the stable domain. As two of the strengths of Hanso are its ability to work from several starting points and to work indistinctly with stable and unstable matrices, we also feed it with two other starting points : the first one is a perturbation of the unstable matrix A, the second one is a perturbation of the starting point chosen previously.

Performance indicators We denote the final solution reached by any of the algorithms by X_f. We can use several criteria depending on whether the theoretic solution is conjectured or not.

First, we will compare the time taken by each of the algorithms.

Then, when the global theoretic solution is known, we can measure the gap between the optimal objective value and the final objective value achieved by each algorithm. We thus define

$$\text{GAP} = \|X_f - A\|_F - \|X_* - A\|_F \ . \tag{3.124}$$

We can further look at how the average error on the coefficients is evolving with the size of the problem. Therefore, when the exact solution X_* is known our third criterion is

$$\text{MEAN ACCURACY} = \frac{1}{n}\|X_f - X_*\|_F \ . \tag{3.125}$$

One can check that (3.125) implies that MEAN ACCURACY takes the value ϵ when the average error on each coefficient of X_f is ϵ as well. As we formulated a conjecture on the solution for the continuous-time benchmark only, the two criteria (3.124) and (3.125) are used in that case.

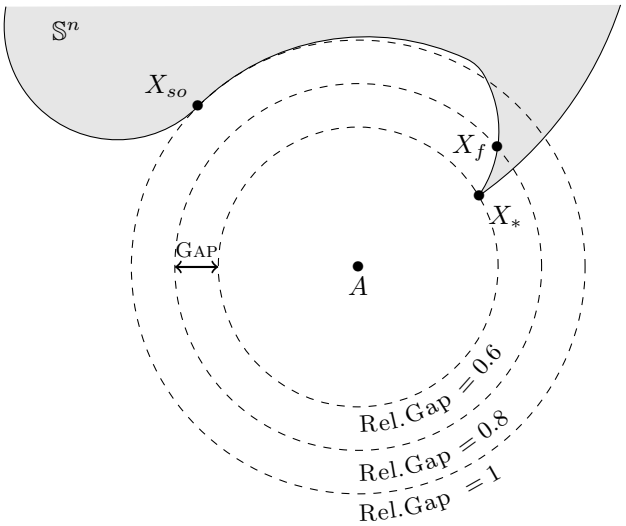


Figure 3.8: Difference of interpretation of the Gap and the Rel.Gap criteria. The relative gap is the ratio between the distance $\|X_{so} - A\|$ and the distance $\|X_f - A\|$ achieved by the solution returned by the algorithm. It is therefore an indicator of the improvement realised by X_f with respect to X_{so}. The gap is an absolute measurement which is the difference $\|X_f - A\| - \|X_* - A\|$. X_f is not represented as a minimum for the purpose of the illustration.

When processing random matrices, our possibilities are much more limited since the theoretic solution is not known. We therefore use a relative measure of the gap with respect to a suboptimal point X_{so}

$$\text{Rel. Gap} = \frac{\|X_f - A\|_F}{\|X_{so} - A\|_F} \ . \tag{3.126}$$

The choice of the suboptimal solution X_{so} should be carried with care. By default, X_{so} should be both easy to compute and achieve a *fair* suboptimal value. If X_{so} satisfies both conditions and the relative gap is greater than 1 then this means that a cheap solution which performs at least as well as X_f is readily available. A good choice for X_{so} highlights the differences of performance between the methods. As computing the relative gap is cheap once X_f is known, many different points can be tried to check the relevance of the relative gap criterion.

Note that the gap (3.124) and the relative gap (3.126) do not measure the same thing but the relative gap is somehow the best approximation we can obtain in the absence of the knowledge of the exact solution. The differences in the interpretation of the criteria are illustrated in Figure 3.8.

3.4.3 Numerical results

Performances related to a size increase Four unstable matrices of sizes $N = 2^i;\ i = 2, \ldots, 5$ were generated each time on which we applied the three algorithms ABPM, GMM and Hanso. The results for the benchmark are displayed in Table 3.1 and in Table 3.2 for the real and the complex case respectively, and illustrated in Figure 3.9. The results for real random matrices are given in Table 3.3 and those for complex random matrices are given in Table 3.4, both are illustrated in Figure 3.10.

In the real case, the results for the benchmark show that the methods tend to approach the expected solution given by

$$X_* = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ 0 & & & 0 \end{bmatrix} , \tag{3.127}$$

for small sizes and preserve the quality of the solution with larger sizes as Figure 3.9e tends to show. It was observed that the ability of Hanso to attain a really good precision for $n = 4$ is made possible by its ability to work with unstable matrices which both GMM and ABPM do not have. It appears very clearly from the differences in gap and accuracy with ABPM and GMM observed for $n = 4$ in Figure 3.9a and Figure 3.9e that this is an argument in favor of Hanso. This difference does not hold for easier cases as Figure 3.10a and 3.10b both suggest.

The large execution times of ABPM that are observable in Figure 3.9c are probably due to the worsening conditioning of the matrices used in the algorithm. It was indeed observed that the conditioning of the matrices which

are used in the construction of the ellipsoid was quickly deteriorating with the dimensionality of the problem. Still for smaller dimensions ABPM yields very good results at low cost and the quality of the results is preserved on the long run.

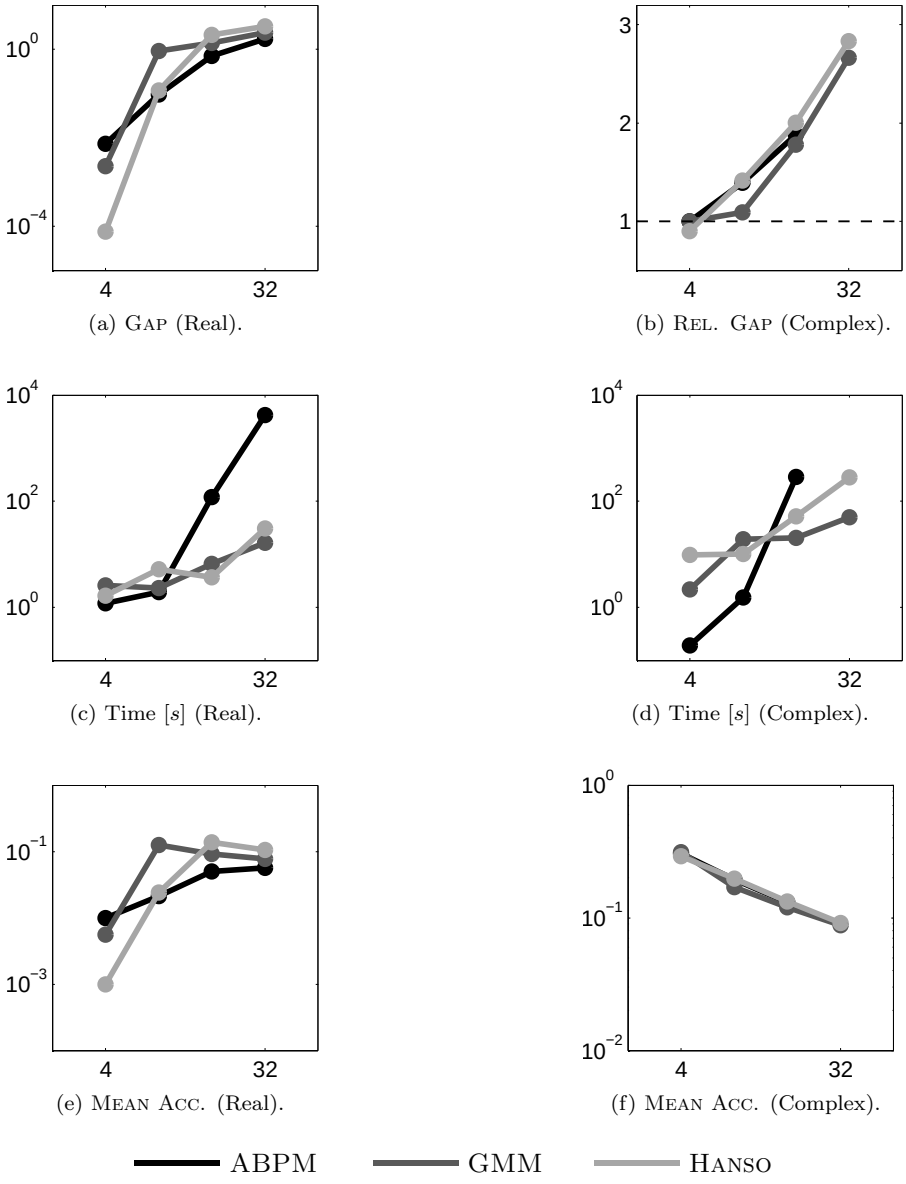
On average, the observed differences between the real and the complex case in Figure 3.9 are not really substantial. Still it is interesting to observe that for $n = 4$, HANSO manages to find a better solution than X_* given by (3.127). According to Figure 3.9f, this solution is nonetheless at a close distance to X_* which tends to prove that X_* probably is not a local minimum. The same graph nonetheless shows that the solutions that are reached are always close to X_* as n grows.

The results contained in Figure 3.10 for random matrices show that the performance of the algorithms is kept constant generally speaking except for HANSO which seems to be less efficient on this type of problem even for small problems. Indeed, the eigenvalues of the approximate minimizer were computed and it was observed that the solution reached was actually unstable when $n = 8$ while the other two methods reached stable solutions in that case. This loss of performance for the sizes $n = 16$ and $n = 32$ is probably due to the fact that HANSO switches to a limited memory version of the algorithm as soon as $n > 10$ (which gives 100 variables) and does not apply additional gradient sampling methods in this case.

The execution times do not vary greatly between the two experiments as is confirmed by a comparison between Figure 3.9c and Figure 3.10c.

Influence of the nonnormality on the performances We represent the gap, the execution times and the mean accuracy obtained for different values of ν_0 applied to the benchmark and with $n = 8$ in Figure 3.11.

In order to improve the relevance of our results for the random case, we performed repeated runs for each value of the conditioning of the similarity matrix T. The results are then averaged and the standard deviation is computed. We give a plot of the average execution times for problems of sizes $n = 4$ and $n = 8$ in Figure 3.12 and the corresponding average distances between the original unstable matrix A and the obtained solution X_f in Figure 3.13. In view of those figures, we can conclude that HANSO does not seem affected at all by the nonnormality of the unstable matrix while ABPM takes more time but reaches approximately the same precision and GMM looses both time and precision along with the nonnormality of the unstable matrix.

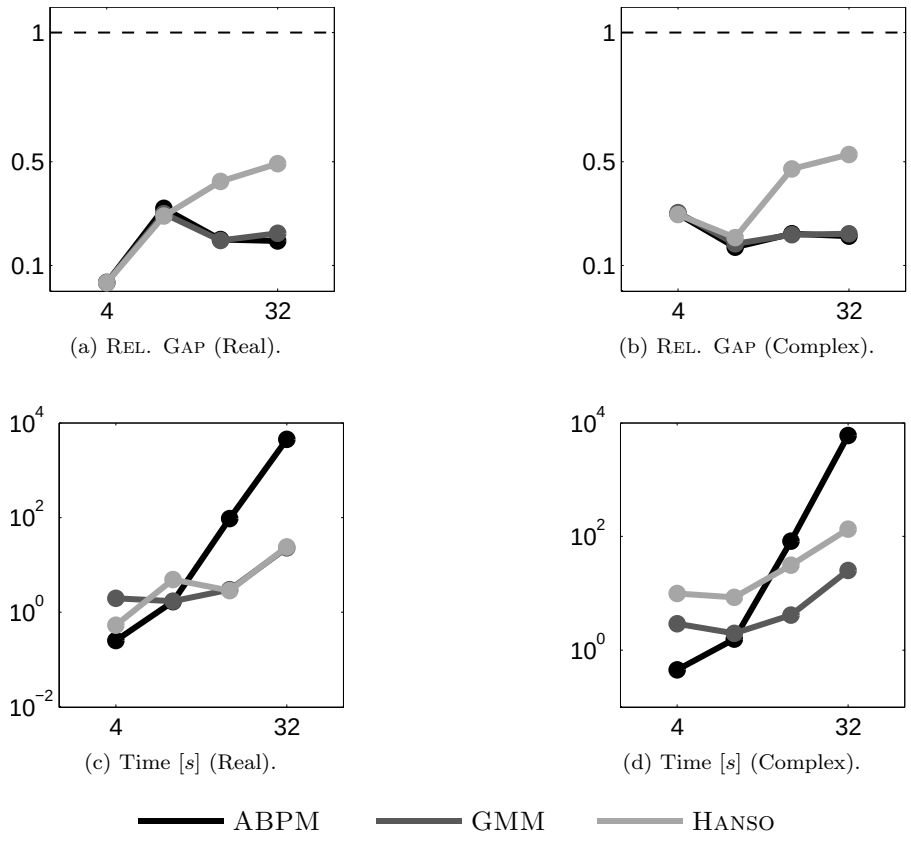


(a) Gap (Real). (b) Rel. Gap (Complex).

(c) Time [s] (Real). (d) Time [s] (Complex).

(e) Mean Acc. (Real). (f) Mean Acc. (Complex).

Figure 3.9: Evolution of the performance of the algorithms with respect to the size of the problem for the benchmark. The lower left coefficient ν_0 in (3.122) was taken as $\nu_0 = 0.1$ in the real case and $\nu_0 = \frac{0.1+j0.3}{|0.1+j0.3|}$ in the complex case. As a better solution than (3.127) was found by Hanso in the complex case for $n = 4$, the Gap was replaced by the relative gap in (b). Still in order to enable comparison the associated data for the gap is given in Tables 3.1 and 3.2 .



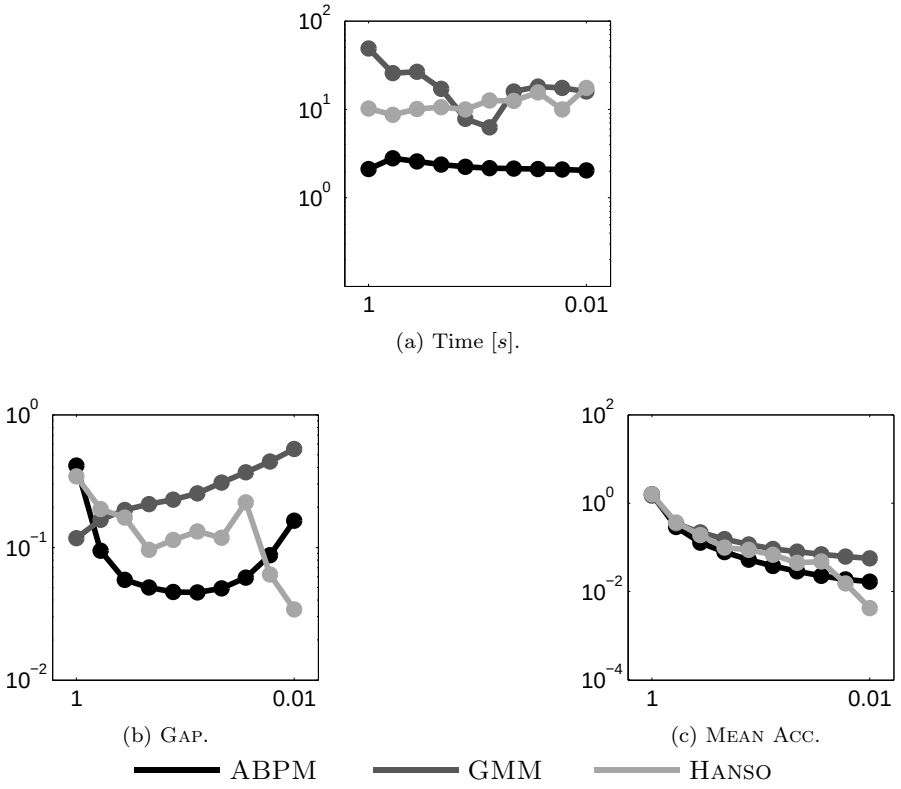
(a) Rel. Gap (Real).

(b) Rel. Gap (Complex).

(c) Time [s] (Real).

(d) Time [s] (Complex).

Figure 3.10: Evolution of the performance of the algorithms with respect to the size of the problem for random matrices. Data is given in Tables 3.3 and 3.4 for real and complex random matrices respectively.



(a) Time [s].

(b) Gap.

(c) Mean Acc.

Figure 3.11: Evolution of the performances of the algorithms with respect to the value of ν_0 for a matrix of size $n = 8$. The values of the parameter ν_0 are taken equidistantly in the interval $[1, 10^{-2}]$. In (a), GMM shows rather high execution times for big values of ν_0. This was not expected since for $\nu_0 \approx 1$ A is close to a normal matrix and is contrary to all previous observations. This is due to the fact that the convergence is very slow on this case but as confirmed by observation in (b) the solution it attains achieves a better gap value. Globally, the gap given in (b) does not vary significantly between the methods but some trends can nonetheless be observed. As Hanso converges indifferently to the solution from the outside or from the inside of the stable set its performance as measured by the gap is rather unaffected by the magnitude of ν_0. On the contrary, ABPM tends to achieve greater gap values as the successive values of ν_0 bring the unstable matrix closer to a singularity of the stable set which is attained for $\nu_0 = 0$. The decreasing feature in (c) indicates that the solutions obtained get closer to the conjectured solution when ν_0 decreases. The figure also confirms that ABPM produces slightly better solutions in this case except for very small ν_0.

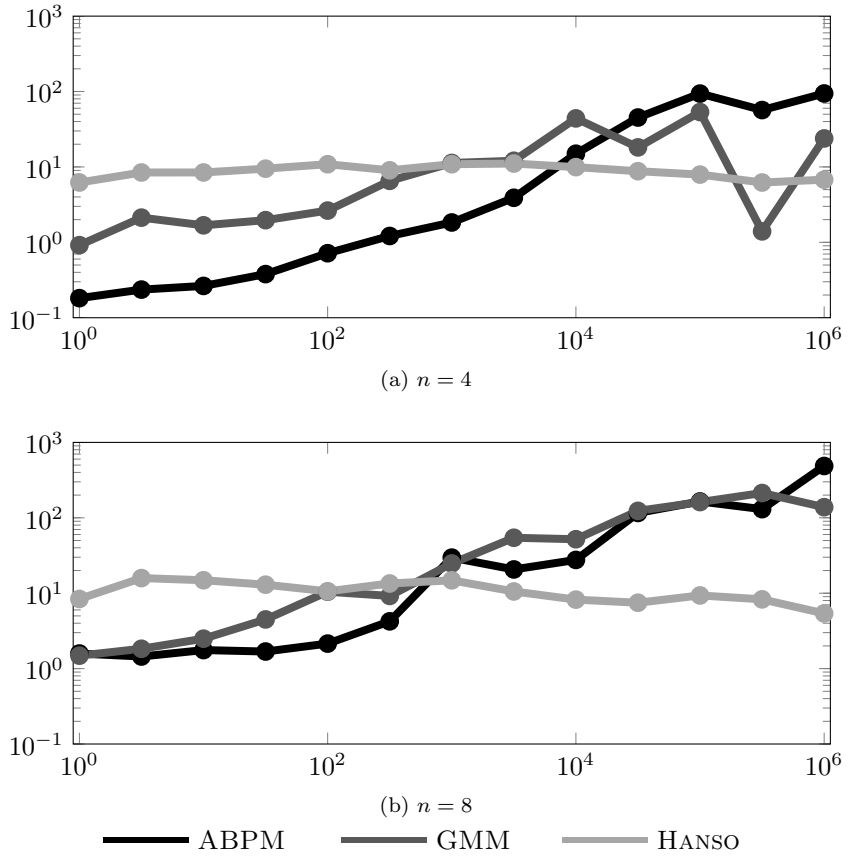


Figure 3.12: Average execution times $[s]$ for matrices of size (a) $n = 4$ and (b) $n = 8$ versus the conditioning of the similarity matrix T given in (3.123). The execution times of Hanso seem unaffected by the conditioning of T while both ABPM and GMM show a high variability in their execution times as soon as T becomes more ill-conditioned. This is confirmed by the standard deviation obtained for each experiment and given in Table 3.5 for $n = 8$. In view of the plots, the effect of the conditioning seems to increase with the size of the matrix.

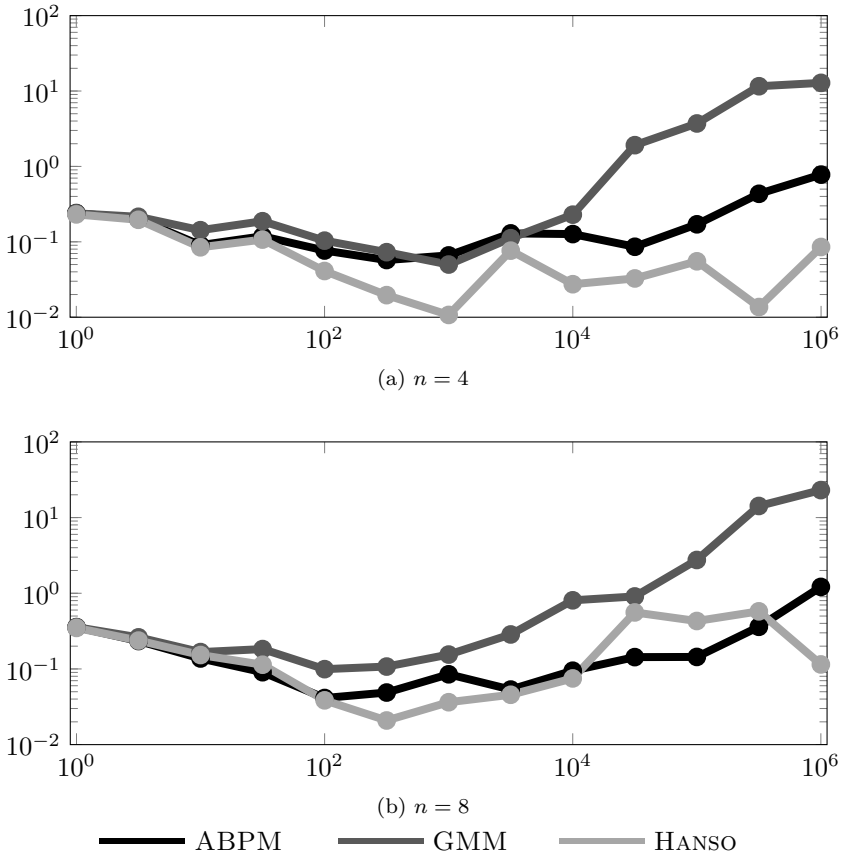


Figure 3.13: Evolution of the average objective function value given by $\|A - X_f\|_F$ for matrices of size (a) $n = 4$ and (b) $n = 8$ in terms of the conditioning of the similarity matrix T given in (3.123). For values of the conditioning close to 1 all methods perform equally well. But soon differences in performance appear due to it. In particular, the precision attained by GMM deteriorates quickly. After some time, the distance $\|A - X_f\|$ logically increases for all methods since the coefficients of the matrices become bigger with the conditioning of T. Nonetheless the effect is much more pronounced for GMM than for ABPM and Hanso.

Conclusions on chapter 3

This chapter's content has shed some light onto the difficulties inherent to the resolution of the nearest stable matrix problem.

The contributions of this chapter are twofold. On the first hand, we suggested two original algorithms for solving the nearest stable matrix problem. On the second hand, we suggested several reformulations of the problem in Section 3.3.1 which could lead to as many variations of the suggested algorithms. In particular, a real arithmetic reformulation suitable for a Newton method was suggested which could prove a promising alternative for the resolution of real problems.

Each of the methods presented here has its own particularities. Theoretically, the first method, which is the topic of Sections 3.1 and 3.2, exploits the mathematical properties associated with self-concordant functions and yields an elegant theory. Unfortunately, our implementation of the method suffers from several weaknesses which makes it suitable for small sizes only. One of the strong points of the methods is its flexibility which enables it to deal with continuous-, and discrete-, time problems as well as with polynomials as the next chapter will show.

The second method tackles the problem from a different point of view. The theory behind it avoids the use of nonconvex constraints and once a final reformulation is attained very simple schemes can be used for its resolution.

We also proved the convergence of both algorithms. We have shown that the Augmented Barrier Projection Method performs at least as well as the Damped Newton method at each step. Its convergence is thus monotonic. The convergence of the Gradient Mapping Method is the one of the usual gradient method.

Numerical experiments have confirmed the relevance of both approaches. Still, our work in Section 3.3 has led us to develop other reformulations. One of them particularly could be solved using a Newton scheme. Preliminary numerical experiments carried on very small examples have returned good results both in terms of time and accuracy.

The work presented in Section 3.1 was communicated in an article [77] in a peer-reviewed journal as well as in several talks and posters at various conferences. The work contained in Section 3.3 was submitted as an article in a peer-reviewed journal [76].

	Time [s]			Gap (3.124)			Mean Accuracy (3.125)		
Size	ABPM	GMM	Hanso	ABPM	GMM	Hanso	ABPM	GMM	Hanso
4	1.2	2.6	1.7	0.0073	0.0023	7.6e-005	9.98e-003	5.59e-003	9.95e-004
8	1.9	2.3	5.2	0.096	0.91	0.12	2.11e-002	1.25e-001	2.41e-002
16	1.2e+002	6.6	3.7	0.7	1.4	2.1	4.98e-002	9.22e-002	1.37e-001
32	4.2e+003	17	30	1.7	2.4	3.3	5.67e-002	7.73e-002	1.06e-001

Table 3.1: Measures of performance of the three methods applied on the benchmark (3.122) with $\nu_0 = 0.1$.

	Time [s]			Gap (3.124)			Mean Accuracy (3.125)		
Size	ABPM	GMM	Hanso	ABPM	GMM	Hanso	ABPM	GMM	Hanso
4	0.2	2.2	9.7	7.3e-010	0.00019	-0.1	3.06e-001	3.13e-001	2.92e-001
8	1.5	19	10	0.39	0.092	0.41	1.95e-001	1.70e-001	1.98e-001
16	2.9e+002	20	52	0.88	0.78	1	1.31e-001	1.20e-001	1.33e-001
32	-	50	2.8e+002	-	1.7	1.8	-	8.77e-002	9.12e-002

Table 3.2: Measures of performance of the three methods applied on the benchmark (3.122) with $\nu_0 = \frac{0.1+j0.3}{|0.1+j0.3|}$.

	Time [s]			Rel. Gap (3.126)			$\|A - X_f\|$		
Size	ABPM	GMM	Hanso	ABPM	GMM	Hanso	ABPM	GMM	Hanso
4	0.3	2.0	0.5	0.033	0.034	0.033	0.069	0.071	0.069
8	1.7	1.7	4.9	0.32	0.3	0.29	0.86	0.81	0.78
16	95	3.0	2.9	0.2	0.2	0.42	0.77	0.76	1.6
32	4.4e+003	23	24	0.19	0.22	0.49	1.1	1.3	2.8

Table 3.3: Measures of performance of the three methods applied to real random matrices.

	Time [s]			Rel. Gap (3.126)			$\|A - X_f\|$		
Size	ABPM	GMM	Hanso	ABPM	GMM	Hanso	ABPM	GMM	Hanso
4	0.4	2.9	10.0	0.3	0.3	0.3	0.48	0.49	0.48
8	1.6	2.0	8.4	0.17	0.18	0.21	0.4	0.43	0.49
16	82	4.1	31	0.22	0.22	0.47	0.73	0.72	1.6
32	6e+003	25	1.3e+002	0.21	0.22	0.53	1	1.1	2.5

Table 3.4: Measures of performance of the three methods applied to complex random matrices.

	Average Time [s]			Standard deviation [%]			obj. f. : $\|A - X_f\|$			Standard deviation of obj. f. [%]		
Cond.	ABPM	GMM	HANSO	ABPM	GMM	HANSO	ABPM	GMM	HANSO	ABPM	GMM	HANSO
10^0	1.6	1.5	8.4	5.4	121.9	31.4	0.36	0.36	0.35	15.9	15.2	16.7
$10^{0.5}$	1.4	1.8	16	14.8	47.0	41.5	0.23	0.26	0.24	36.3	29.7	42.2
10^1	1.8	2.5	15	36.3	30.9	30.5	0.14	0.17	0.15	35.4	23.8	47.8
$10^{1.5}$	1.7	4.5	13	12.6	72.1	27.8	0.091	0.18	0.11	65.7	25.7	84.5
10^2	2.1	10	11	16.9	62.8	36.9	0.041	0.099	0.038	73.1	24.3	99.2
$10^{2.5}$	4.2	9.2	13	68.7	46.8	26.1	0.049	0.11	0.021	131.4	26.1	143.8
10^3	30	25	15	206.7	76.1	25.0	0.085	0.15	0.036	114.0	37.1	83.5
$10^{3.5}$	21	54	11	70.9	72.7	33.9	0.053	0.29	0.046	201.8	30.3	91.3
10^4	28	52	8.2	43.8	67.1	43.8	0.095	0.81	0.075	85.9	60.1	102.1
$10^{4.5}$	1.2e+002	1.2e+002	7.5	149.8	58.4	37.9	0.14	0.91	0.56	87.9	27.0	200.6
10^5	1.6e+002	1.6e+002	9.3	179.4	96.3	44.3	0.14	2.8	0.43	53.3	121.4	247.7
$10^{5.5}$	1.3e+002	2.1e+002	8.3	188.7	157.0	38.7	0.36	14	0.58	104.2	102.0	262.7
10^6	4.9e+002	1.4e+002	5.4	72.4	79.0	31.1	1.2	23	0.11	59.8	124.1	37.2

Table 3.5: Execution times, distance to the stable set and standard deviation for complex random matrices of size $n = 8$. The notation "Cond." refers to the conditioning of the matrix T in (3.123).

3.A Bounds for the Lipschitz constants for the Gradient Mapping Method

We are interested in bounding the values of the Lipschitz constants L_1 and L_2 that appear in the gradient scheme in Theorem 3.19. It is well known that the value of the Lipschitz constants is bounded by the norm of the Hessian of the function; see [73] for example.

The expression of the objective function is given by (3.108) which we rewrite here for convenience

$$f(U,\Lambda) = \|A\|^2 - \frac{\langle AU, U\Lambda\rangle_-^2}{\|\Lambda\|^2+\epsilon} - \frac{1}{2}\langle W_\epsilon \odot C, C\rangle + \epsilon\|\Lambda\|^2 \ . \tag{3.128}$$

It will simplify our computations to restrict ourselves to the particular case $A \in \mathbb{R}^{n\times n}$. When A is real, the Lyapunov equation is real as well and thus both U in (3.128) and A are real. The realness assumption changes the value of the bound but does not change the fact that the bound will be finite.

It appears clear in (3.128) that the critical terms for the computation of the Hessian are those where Λ appears in the denominator of the function. Therefore, and for the sake of simplicity, we limit our computation to those terms given by

$$t_1(\Lambda) = -\frac{\langle \tilde{A}, \Lambda\rangle_-^2}{\|\Lambda\|^2+\epsilon} \ , \tag{3.129}$$

and

$$t_2(\Lambda) = \frac{1}{2}\langle W_\epsilon \odot \tilde{C}, \tilde{C}\rangle \ , \tag{3.130}$$

where the equalities $\tilde{A} = U^*AU$, $\tilde{C} = U^*CU$ were used for convenience.

The first step in bounding the Lispchitz constants consists in changing our formulation for $t_1(\Lambda)$ into the equivalent form

$$t_1(\Lambda) = \left(\tilde{t}_1\right)_+^2 \ ,$$

where

$$\tilde{t}_1(\Lambda) = \frac{\langle \tilde{A}, \Lambda\rangle}{\sqrt{\|\Lambda\|^2+\epsilon}} \ . \tag{3.131}$$

From our assumptions, and based on our definitions of $\tilde{C}$ and W_ϵ in (3.100) and in (3.101) respectively, we can rewrite $t_2(\Lambda)$ as

$$t_2(\Lambda) = \frac{1}{2}\sum_{ij}(\tilde{t}_{2,ij}(\Lambda))_+^2 \ ,$$

where

$$\tilde{t}_{2,ij}(\Lambda) = \frac{1}{\sqrt{\lambda_i^2+\lambda_j^2+\epsilon}}(\tilde{A}_{ij}\lambda_j - \lambda_i\tilde{A}_{ij}^T) \qquad i,j \in [1,n] \ . \tag{3.132}$$

Now let us remark that both $\tilde{t}_1(\Lambda)$ in (3.131) and the various terms $\tilde{t}_{2,ij}$ in (3.132) can be put into a generic form.

Proposition 7. *Let $E \subset \mathbb{R}^n$ be a vector space of appropriate dimensions, different for $\tilde{t}_1$ and $\tilde{t}_{2,ij}$, endowed with the standard scalar product $\langle x, y\rangle_2 = x^T y$ for any two vectors $x, y \in E$. Then for some appropriate vectors $m, x \in E$, both $\tilde{t}_1(\Lambda)$ and $\tilde{t}_{2,ij}(\Lambda)$ can be put into the form*

$$t(x) = \frac{\langle m, x\rangle_2}{\sqrt{\|x\|_2^2 + \epsilon}} \ . \tag{3.133}$$

Proof. Let us prove first that this is true for $\tilde{t}_1(\Lambda)$, we will then prove that the statement holds for $\tilde{t}_{2,ij}(\Lambda)$ as well.

Let $E = \mathbb{R}^n$ and $x_i = \lambda_i, \forall i$, then choosing $m_i = \tilde{A}_{ii}$ yields the desired result for $\tilde{t}_1(\Lambda)$.

In order to prove the same results for the terms $\tilde{t}_{2,ij}$, let us take $E = \mathbb{R}^2$ and $x^T = \begin{bmatrix}\lambda_i & \lambda_j\end{bmatrix}$. Taking $m^T = \begin{bmatrix}-\tilde{A}_{ji} & \tilde{A}_{ij}\end{bmatrix}$ yields the desired result. □

In order to derive a bound for $t_1(\Lambda)$ from the generic form $t(x)$, we need an additional proposition.

Proposition 8. *Let $E \subset \mathbb{R}^n$ be a vector space of appropriate dimensions endowed with the standard scalar product $\langle x, y\rangle_2 = x^T y$ for any two vectors $x, y \in E$. Then consider the scalar functions $\phi(x) : \mathbb{R} \to \mathbb{R}$ and $t(x)$ defined by (3.133) such that*

$$\phi(x) = (t(x))_+^2 \ .$$

If we denote the gradient of t by t' and its Hessian by t'', then the norm of the Hessian ϕ'' of ϕ is bounded almost everywhere with the bound

$$\|\phi''(x)\| \leq \|t' t'^* + t t''\| \ .$$

Proof. It suffices to compute the successive derivatives of ϕ. We have for a direction $h \in E$,

$$\phi'(x) = t' t_+ \ ,$$

and

$$\phi''(x)[h] = \begin{cases} t' t'^T + t t''[h] & \text{if } t > 0 \ , \\ 0 & \text{if } t < 0 \ . \end{cases}$$

Note that the Hessian does not exists if $t = 0$. Thus, the bound

$$\|\phi\| < \|t' t'^T + t t''[h]\|$$

is valid almost everywhere. □

Proposition 8 defines a bound for the Hessian of ϕ which is valid almost everywhere, so we can derive a bound for the Hessian of $t_1(\Lambda)$ from an estimate of the derivatives of the function $t(x)$. For $t_2(\Lambda)$ the bound is defined everywhere for each of the elements of the sum.

Following Propositions 7 and 8, it suffices to compute generic bounds for the derivatives of (3.133) to deduce an approximate bound for the Hessian of (3.128). Let us compute those generic bounds.

The first order directional derivative of (3.133) in a direction $h \in E$ is

$$Dt(x)[h] = \frac{\langle m, h\rangle_2}{\sqrt{\|x\|_2^2 + \epsilon}} - \frac{\langle m, x\rangle_2}{(\|x\|_2^2 + \epsilon)^{\frac{3}{2}}}\langle x, h\rangle_2 \ .$$

For any bounded direction $h \in E$, a bound for $\|Dt(x)[h]\|$ is given by

$$\begin{aligned} |Dt(x)[h]| &\leq \frac{\|m\|\|h\|}{\sqrt{\|x\|_2^2 + \epsilon}} + \frac{\|m\|\|x\|}{(\|x\|_2^2 + \epsilon)^{\frac{3}{2}}}\|x\|\|h\| \\ &\leq \frac{C_1}{\epsilon^{\frac{1}{2}}}\|h\| \ . \end{aligned}$$

In turn, the second order directional derivative is thus

$$D^2t(x)[h,h] = -2\frac{\langle m, h\rangle_2}{(\|x\|_2^2 + \epsilon)^{\frac{3}{2}}}\langle x, h\rangle_2 + 3\frac{\langle m, x\rangle_2}{(\|x\|_2^2 + \epsilon)^{\frac{5}{2}}}\langle x, h\rangle_2^2 \ .$$

For any bounded direction $h \in E$ we obtain, using successive Cauchy-Schwarz inequalities,

$$|D^2t(x)[h,h]| \leq 2\frac{\|m\|\|h\|^2}{(\|x\|_2^2 + \epsilon)^{\frac{3}{2}}}\|x\| + 3\frac{\|m\|\|x\|}{(\|x\|_2^2 + \epsilon)^{\frac{5}{2}}}\|x\|^2\|h\|^2 \ .$$

Therefore, we can bound the product $\|t(x)\||D^2t(x)[h,h]|$ by

$$\begin{aligned} \|t(x)\||D^2t(x)[h,h]| &\leq \frac{\|m\|\|x\|}{\sqrt{\|x\|_2^2 + \epsilon}}\left(2\frac{\|m\|\|h\|^2}{(\|x\|_2^2 + \epsilon)^{\frac{3}{2}}}\|x\| + 3\frac{\|m\|\|x\|}{(\|x\|_2^2 + \epsilon)^{\frac{5}{2}}}\|x\|^2\|h\|^2\right) \ , \\ &= 2\frac{\|m\|^2\|h\|^2\|x\|^2}{(\|x\|_2^2 + \epsilon)^2} + 3\frac{\|m\|^2\|x\|^4\|h\|^2}{(\|x\|_2^2 + \epsilon)^3} \end{aligned}$$

and since the maximum of this last equation is attained for $\|x\|^2 = \delta\epsilon$ for some appropriate $\delta \in \mathbb{R}_+$ we finally obtain

$$\|t(x)\||D^2t(x)[h,h]| \leq \frac{C_2}{\epsilon}\|h\|$$

where C_2 is an appropriate constant.

This last result enables us to conclude that, for an appropriate constant C_3,

$$\|\nabla^2 f(U,\Lambda)\|_F \leq C_3\frac{1}{\epsilon} \ .$$

3.B Expressions of the gradients for the Gradient Mapping Method

Through the equations (3.106) and (3.98) we can implicitly construct the gradients of the function with respect to U and Λ. Let us recall again the objective function

$$f(U,\Lambda) = \|A\|^2 - \frac{\langle AU, U\Lambda\rangle^2}{\|\Lambda\|^2 + \epsilon} - \frac{1}{2}\langle j(AU\Lambda U^* - U\Lambda U^*A^*), H_\epsilon(U,\Lambda)\rangle + \epsilon\|\Lambda\|^2 \ . \tag{3.134}$$

Gradient in Λ

Consider the more easy-to-visualize form, obtained by using the change of variables $\tilde{A} = U^* A U$, $\tilde{H}_\epsilon = U^* H_\epsilon U$.

$$f(U,\Lambda) = \|\tilde{A}\|^2 - \frac{\langle \tilde{A}, \Lambda\rangle_-^2}{\|\Lambda\|^2 + \epsilon} - \frac{1}{2}\langle j(\tilde{A}\Lambda - \Lambda\tilde{A}^*), \tilde{H}_\epsilon(U,\Lambda)\rangle + \epsilon\|\Lambda\|^2 \ .$$

Denoting by $Df[E] = \langle \nabla_\Lambda f, E\rangle$ the directional derivative of f with respect to Λ in the direction E we get a first term

$$\begin{aligned}\Big(\langle j(\tilde{A}\Lambda - \Lambda\tilde{A}^*), \tilde{H}_\epsilon(U,\Lambda)\rangle\Big)' = &-j(\tilde{A}^*\tilde{H}_\epsilon(U,\Lambda) - \tilde{H}_\epsilon(U,\Lambda)\tilde{A}) \\ &+ D^*\tilde{H}_\epsilon(U,\Lambda)[\tilde{N}] \ . \end{aligned} \tag{3.135}$$

Following the definition, the term (3.135) corresponds to the *adjoint* of the directional derivative of $\tilde{H}_\epsilon$ in the direction $\tilde{N} = j(\tilde{A}\Lambda - \Lambda\tilde{A}^*)$. The whole gradient thus becomes

$$\begin{aligned}\nabla_\Lambda f(U,\Lambda) = &-2\frac{\langle \tilde{A}, \Lambda\rangle_-}{(\|\Lambda\|^2 + \epsilon)^2}((\|\Lambda\|^2 + \epsilon)\tilde{A} - \langle \tilde{A}, \Lambda\rangle_- \Lambda) \\ &+ \frac{1}{2} j(\tilde{A}^*\tilde{H}_\epsilon(U,\Lambda) - \tilde{H}_\epsilon(U,\Lambda)\tilde{A}) \\ &- \frac{1}{2} D^*\tilde{H}_\epsilon(U,\Lambda)[\tilde{N}] + 2\epsilon\Lambda \ . \end{aligned} \tag{3.136}$$

Finally, as Λ is a diagonal matrix, $\nabla_\Lambda f(U,\Lambda)$ is diagonal as well and we thus project the gradient (3.136) in the tangential plane by keeping its diagonal elements only. The expression for $\tilde{H}'_\epsilon(U,\Lambda)[\tilde{N}]$ can be computed as the solution of

$$(\Lambda^2 + \epsilon I)D\tilde{H}_\epsilon[\tilde{N}] + D\tilde{H}_\epsilon[\tilde{N}](\Lambda^2 + \epsilon I) = j(\tilde{A}\tilde{N} - \tilde{N}\tilde{A}^*) - (M[\tilde{N}]\tilde{H} + \tilde{H}M[\tilde{N}]) \ , \tag{3.137}$$

where $M = (\tilde{N}\Lambda + \Lambda\tilde{N})$. It can be checked numerically that the operator is not self-adjoint since for two random direction E_1 and E_2 we have

$$\langle D\tilde{H}_\epsilon(U,\Lambda)[E_1], E_2\rangle \neq \langle E_1, D\tilde{H}_\epsilon(U,\Lambda)[E_2]\rangle \ .$$

Let us rewrite the equation (3.137) as

$$(\Lambda^2 + \epsilon I)D\tilde{H}_\epsilon[E_1] + D\tilde{H}_\epsilon[E_1](\Lambda^2 + \epsilon I) = K(U,\Lambda)[E_1] \ ,$$

and let us write the scalar product of this expression with E_2

$$\langle (\Lambda^2 + \epsilon I)D\tilde{H}_\epsilon[E_1] + D\tilde{H}_\epsilon[E_1](\Lambda^2 + \epsilon I), E_2\rangle = \langle K(U,\Lambda)[E_1], E_2\rangle$$

so both adjoints become

$$\langle E_1, D^*\tilde{H}_\epsilon[((\Lambda^2 + \epsilon I)E_2 + E_2(\Lambda^2 + \epsilon I)]\rangle = \langle E_1, K^*(U)[E_2]\rangle \ . \tag{3.138}$$

Thus following (3.138)

$$D^*\tilde{H}_\epsilon[\tilde{N}] = K^*(U,\Lambda)[\hat{N}] \tag{3.139}$$

with $\hat{N}$ the solution of

$$(\Lambda^2 + \epsilon I)\hat{N} + \hat{N}(\Lambda^2 + \epsilon I) = \tilde{N} \ .$$

The expression of $K^*(U,\Lambda)[E]$ is given by

$$K^*(U,\Lambda)[E] = j(E\tilde{A} - \tilde{A}^*E) - (N[E]\Lambda + \Lambda N[E]) \ ,$$

and $N[E]$ is

$$N[E] = E\tilde{H}_\epsilon + \tilde{H}_\epsilon E \ .$$

Gradient in U

As in our preceding development, we denote by $Df[E] = \langle \nabla f_U, E \rangle$ the directional derivative of f with respect to U in the direction E. The gradient of (3.134) is

$$\begin{aligned}\nabla_U f(U,\Lambda) = -2\frac{\langle AU, U\Lambda\rangle_-}{\|\Lambda\|^2 + \epsilon}(A^* + A)U\Lambda - j(H_\epsilon A - A^* H_\epsilon)U\Lambda \\ - \frac{1}{2}D^*H_\epsilon(U,\Lambda)[N] \ .\end{aligned}$$

As the expression of the gradient should belong to the tangent plane T of the manifold of unitary matrices at U, it is appropriate to return a projected gradient $P_T(\nabla_U f)$ using the adequate projection equation [1]

$$P_T(\nabla_U f) = \nabla_U f - \frac{1}{2}U(U^*\nabla_U f + \nabla_U f^* U) \ .$$

Let us now detailed the expression of $\nabla_U f$. First, one can remark

$$D^*H_\epsilon(U,\Lambda)[N] = UD^*\tilde{H}_\epsilon[\tilde{N}]U^*$$

where $N = U\tilde{N}U^*$. This can be observed by carefully putting up both adjoints but is consistent with the fact that DH_ϵ is linear in N. Computing the derivative in U of (3.98) in the direction E gives

$$(\Lambda^2 + \epsilon I)D\tilde{H}_\epsilon[\tilde{E}] + D\tilde{H}_\epsilon[\tilde{E}](\Lambda^2 + \epsilon I) = j(\tilde{A}\tilde{G}_2 - \tilde{G}_2\tilde{A}^*) - (\tilde{G}_1\tilde{H}_\epsilon + \tilde{H}_\epsilon\tilde{G}_1) \ ,$$

where $\tilde{G}_1[\tilde{E}] = (\tilde{E}U^*\Lambda^2 + \Lambda^2 U\tilde{E})$ and $\tilde{G}_2[\tilde{E}] = (\tilde{E}U^*\Lambda + \Lambda U\tilde{E})$. As above, the adjoint

$$D^*\tilde{H}_\epsilon[\tilde{E}] = K^*(U,\Lambda)[\hat{E}]$$

where $K^*(U,\Lambda)[\hat{E}]$ is now, using $\breve{E} = \hat{E} + \hat{E}^*$,

$$K^*(U,\Lambda)[\hat{E}] = j(\breve{E}\tilde{A} - \tilde{A}^*\breve{E})\Lambda U - (\tilde{H}_\epsilon\breve{E} + \breve{E}\tilde{H}_\epsilon)\Lambda^2 U$$

and where $\hat{E}$ is solution of

$$(\Lambda^2 + \epsilon I)\hat{E} + \hat{E}(\Lambda^2 + \epsilon I) = \tilde{E} \ .$$

Note that in case $\tilde{E}$ is symmetric then $\hat{E}$ is symmetric as well and $\breve{E} = 2\hat{E}$.

Chapter 4

Methods for polynomials

The development of methods that are able to find the nearest stable polynomial to an unstable one is an important topic which deserves a discussion on its own. Polynomials are encountered in many applications and the relevance of this question goes well beyond the theoretical interest.

Work on the stability of polynomials goes a long time into the past, and many results have been published. Of particular interest is the well-known Routh-Hurwitz stability criterion which determines the stability of a continuous-time polynomial from its coefficients using simple algebraic conditions. Its discrete-time equivalent is Jury's test. Unfortunately, the Routh-Hurwitz criterion and Jury's test require a large number of computations and thus do not offer a viable framework for finding the nearest stable system. We develop alternative strategies in this chapter.

With appropriate restrictions on the coefficients, the nearest stable polynomial problem can become an easy one. In particular, suppose one is interested in the stability of a discrete-time real polynomial $x(z)$ such that

$$x(z) = z^n - \sum_{i=0}^{n-1} x_i z^i \qquad x_i \in \mathbb{R}_+, \quad \forall i \tag{4.1}$$

Then the following result follows.

Theorem 4.1. *Given a real monic polynomial $x(z)$ of the form (4.1) where each coefficient x_i is nonnegative, we have*

$$\rho(x) \leq 1 \Leftrightarrow \sum_{i=0}^{n-1} x_i \leq 1$$

Proof. We will prove the theorem from the companion form of the polynomial (4.1) which is

$$X = \begin{bmatrix} x_{n-1} & 1 & & \\ \vdots & & \ddots & \\ & & & 1 \\ x_0 & & & 0 \end{bmatrix}$$

As $x_i \geq 0 \ \forall i$, X is a nonnegative matrix. Without loss of generality, let us suppose that $x_0 \neq 0$ so that the matrix X is also irreducible. Otherwise, we can apply the process and the submatrix obtained by suppressing the last row and the last column of X.

Let us prove that $\sum x_i \leq 1 \Rightarrow \rho(X) \leq 1$. As X is a nonnegative matrix, by Theorem 8.3.1 in [50],

$$\rho(X) \leq \max(\sum x_i, 1) \ ,$$

and therefore, $\sum x_i \leq 1 \Rightarrow \rho(X) \leq 1$. This can also been seen by the fact that if $\sum x_i \leq 1$ then X is a substochastic matrix and the same conclusion follows.

In order the prove the converse statement, let us observe that a consequence of Theorem 1.20 due to Perron-Frobenius is that for any nonnegative matrix A if $\rho(A) \leq 1$ then there exists an invertible diagonal matrix D such that

$$\mathbb{1}^T D^{-1} A D \leq \mathbb{1}^T \ . \tag{4.2}$$

Indeed, following Perron-Frobenius for any nonnegative matrix $A \in \mathbb{R}^{n \times n}$, the Perron vector $u > 0$ is such that

$$u^T A = \rho(A) u^T \ .$$

Then any invertible diagonal matrix D satisfies

$$u^T D D^{-1} A D = \rho(A) u^T D \ . \tag{4.3}$$

The matrix D can always be chosen such that

$$u^T D = \mathbb{1}^T \ . \tag{4.4}$$

Plugging (4.4) into (4.3) along with the fact that $\rho(A) \leq 1$ yields (4.2).

This result also applies to X and there exists a matrix D such that

$$\mathbb{1}^T D^{-1} X D \leq \mathbb{1}^T \ .$$

Let us look at the product $D^{-1}XD$ and let us denote the ith diagonal element of D by d_i. As D is a diagonal similarity transformation, it can always be scaled so that

$$d_1 = 1 \ .$$

Then, the expression of the product $D^{-1}XD$ is

$$D^{-1}XD = \begin{bmatrix} x_{n-1} & d_2 & & & \\ \frac{x_{n-2}}{d_2} & & \frac{d_3}{d_2} & & \\ \vdots & & & \ddots & \\ \vdots & & & & \frac{d_n}{d_{n-1}} \\ \frac{x_0}{d_n} & & & & 0 \end{bmatrix} \ .$$

By (4.2), the sum of the elements of each column must be less than 1. When applied on the $n-1$ last columns of X, this last condition imposes that

$$d_n \leq \cdots \leq d_2 \leq 1 \ .$$

The first column yields the condition

$$\sum \frac{x_i}{d_i} \leq 1 \ ,$$

and since $d_i \leq 1 \ \forall i$, we deduce that

$$\sum x_i \leq 1 \ .$$

□

This theorem demonstrates that for a selection of domains, the solution to the nearest stable polynomial to an unstable one is not difficult to find. In the particular case of Theorem 4.1 , we have shown that one can even build a convex optimization problem where the domain is a simplex. This preliminary result also gives us further hope that polynomials behave better than matrices in general.

We will not pursue further the investigation of special cases and our research will focus on the more general coefficients space given by $\mathbb{R}^n$ or $\mathbb{C}^n$ in which case finding the nearest stable polynomial is deemed, unsurprisingly, a difficult problem. The chapter is structured as follows :

Section 4.1 explores the existing methods that stabilize polynomials through simple but efficient techniques. The interest is twofold, it provides a perspective on the existing literature on the subject and introduces algorithms that can be used to construct much needed starting points.

Section 4.2 is the continuation of Section 3.1 and 3.2 while trying to exploit the reduced dimensionality induced by the use of polynomials to obtain a more efficient algorithm. Some time is taken to discuss the pertinence of the choices and the improvements that could be undertaken in the future.

Section 4.3 discusses techniques for finding the nearest stable polynomial that are constructed on the principle that it is easier to optimize over the roots of the polynomials rather than on its coefficients. Optimizing over the roots requires the use of separate algorithms for continuous-time and discrete-time. We also explain why the computation of the gradient of the objective function becomes a delicate step and discuss the best way for building it.

Section 4.4 compares the performances of the various algorithms dedicated to polynomials that were introduced in the chapter or that already exist.

Section 4.5 briefly looks into the implication of using a weighted norm in the objective function instead of the usual Frobenius norm. This section is a very brief feasibility study of the approach and how to handle it in our algorithms.

4.1 Stabilization of polynomials

Stabilizing a polynomial is a well explored topic in the literature. The goal is to construct a stable polynomial from some unstable one, but not necessarily the nearest one. Hence, for those methods the emphasis is put on the efficiency rather than on the quality of the approximant as is done in this thesis.

The first step usually consists in identifying the roots of the polynomial which are responsible for the instability. The second step then consists in modifying the unstable roots in such a way that the entire polynomial then becomes stable.

As we will see, one does not necessarily need to actually compute the roots, which can be a quite time-consuming task, and alternative approaches have been developed in the literature which are applicable to continuous-time or discrete-time instances, or both. Still, most of these alternatives were developed for discrete-time polynomials. The Möbius transform can be in turn used to apply those methods on continuous-time polynomials but the transformation does not behave well numerically.

Our interest in this topic lies in the possibility of getting starting points for our algorithms through low complexity methods.

4.1.1 Stabilization by the roots

Computing the roots of a polynomial and modifying those that are unstable is probably the most natural way of stabilizing a polynomial. The complexity of computing the roots of a polynomial is the one of computing the eigenvalues of a companion matrix which has recently been lowered to $\mathcal{O}(n^2)$ [96, and references therein]. Unfortunately, the time constants are apparently such that for small sizes the complexity remains $\mathcal{O}(n^3)$. Nonetheless, computing the roots is still attractive since it gives some direct insight on the distance to stability through bounds that can be evaluated (see Section 2.2).

Once we know the position of the roots of the polynomials, there are several ways of stabilizing them, for example

- mirroring with respect to some boundary;
- preserving the distribution (translation/contraction) ;
- displacing them to a predefined location.

From those three possibilities, and with the prospect of feeding the stabilized polynomial as a starting point to any of our algorithms, the “mirrored” option is probably the best choice and translation the worst one. This is intuitively justified by the fact that the best choice modifies as little as possible the roots of the original polynomial. Also, unstable roots that are close to the stability boundary are probably best approximated by a stable root just across the stability boundary. These two criteria are best fulfilled by the mirrored method.

Choosing the stability boundary as the reflection axis for our computations is not necessarily the best idea. Indeed, roots that are on the boundary would remain unchanged and therefore could cause the convergence of most methods

to be slow since they are frequently unfitted to treat those instances. Taking a mirror axis slightly inside the stability region by some user-defined margin, we will avoid this pitfall.

As a final comment, one should be cautious when applying this technique to very large polynomials since the magnitude of the coefficients of a polynomial explodes whenever the modulus the roots is larger than 1. This will mainly be a burden for continuous-time polynomials since stable continous-time can have roots with a modulus larger than 1 and thus potentially have coefficients exploding in magnitude.

4.1.2 Factoring into a stable and an unstable polynomial

The incomplete factorization of a polynomial has been investigated for some time. Most articles concentrating on this subject relate to discrete-time polynomials whose numerical properties are much better in this case.

The factorization of a polynomial has been investigated significantly in the literature. The discrete-time case is especially well documented due to its utility to root finding algorithms [45, 55, 81, and reference therein]. On the contrary, the continuous-time case has only a few references. One of these few is the work of Gemignani [36] who adapted the LU matrix decomposition algorithm for the matrix eigenvalue problem into an algorithm for factorizing polynomials with respect to the imaginary axis. The method yields iterates with a complexity in $\mathcal{O}(n(n-n_s))$ where n_s is the number of stable roots, along with linear convergence, but it still has some open questions and room for improvement. In particular, it requires the prior knowledge of the number of stable zeros which we do not know unless we first compute them, but then the algorithm is of no use. Therefore we do not pursue its description here.

The principles of incomplete factorization are easily understood. Given a polynomial $a(\lambda) \in \mathbb{C}^{n+1}$ of degree n, we want to factorize it into its stable part $a_s(\lambda)$ and unstable part $a_u(\lambda)$,

$$a(\lambda) = a_s(\lambda)a_u(\lambda) \ .$$

The unstable part $a_u(\lambda)$ can then be stabilized and a stable polynomial $p(\lambda)$ can be reconstructed. Let us denote by n_u the degree of a_u. In discrete-time, one typically computes

$$p(z) = a_s(z)a_{u,\star}(z) = a_s(z)z^{n_u}\bar{a}_u(z^{-1})$$

which features the original stable roots and the mirrored roots of $a_u(z)$ with respect to the unit circle. The continuous-time equivalent is given by

$$p(s) = a_s(s)a_{u,\star}(s) = a_s(s)\bar{a}_u(-s)$$

which mirrors the unstable roots with respect to the imaginary axis.

Let us first discuss the discrete-time case which behaves much better than the continuous-time one. Indeed in discrete-time, instead of looking for $a_u(z)$ for which the roots might be virtually anywhere outside the unit circle, we look for the polynomial $a_s(z)$ whose roots are contained inside the unit disk.

Factoring a_s out of p then yields the unstable polynomial a_u which can then be processed accordingly. The perspective for the stabilization of continuous-time polynomials is not quite as good since both the stable and the unstable sets of roots are unbounded sets in the complex plane. This explains partly why there is still no appropriate method for the stabilization of continuous-time polynomials.

The method that we discuss now is aimed at the incomplete factorization of discrete-time polynomials and is based on approximating the polynomial $a(z)$ by its Laurent series on the unit disk. For the sake of the argument we will consider that $a(z)$ does not necessarily need to be unstable but most likely is. We define n_s as the number of stable roots of $a(z)$ where n_s can take any value between 0 and n. Let

$$f(z) = \frac{a'(z)}{a(z)} \ , \tag{4.5}$$

and let us set $\gamma = 1 - \delta$ for a small positive δ, we define the annulus $\mathcal{N}$ around $|z| = 1$ by

$$\mathcal{N} \stackrel{\text{def}}{=} \{z \in \mathbb{C} \ : \ \gamma \leq |z| \leq \gamma^{-1}\} \ . \tag{4.6}$$

Provided that $a(z)$ has no zeroes inside $\mathcal{N}$, $f(z)$ is analytic on $\mathcal{N}$ and the Laurent coefficients of $f(z)$ can be computed.

At the same time, as $\mathcal{N}$ separates the roots into two groups we can rewrite $f(z)$ in two sums [45]

$$f(z) = \sum_{m=1}^{\infty} z^{-m} s_{m-1} - \sum_{m=1}^{\infty} z^{-m} t_{-m-1}$$

where

$$s_m = \sum_{i=1}^{n_s} z_i^m \qquad \text{and} \quad t_m = \sum_{i=n_s+1}^{n} z_i^m \ ,$$

are the power sums associated with the inner roots ($|z_i| < \gamma^{-1}$) and outer roots ($|z_i| > \gamma^{-1}$), respectively. Moreover it can be checked by inspection that the power sums s_m correspond to the definition of Laurent coefficients, that is

$$a_{-m} \approx s_{m-1} \quad m = 1, 2, ...$$

This observation makes the use of the Laurent coefficients for approximating the power sums possible with an error given by

$$|a_m - s_{m-1}| \propto \frac{\gamma^d}{1 - \gamma^d} \ , \tag{4.7}$$

where d is the number of Fourier coefficients used for the approximation. The important step here is to notice that the knowledge of the power sums s_m is sufficient for the reconstruction of a stable polynomial $p(z)$ whose roots z_i are only those satisfying $|z_i| < \gamma$. We do not describe the inverse procedure here but it is worth mentioning that the construction of $p(z)$ is carried out in time $\mathcal{O}(n_s \log(n_s))$. We refer the interested reader to the review of Henrici [45] for more details on this step.

The complexity of the overall scheme is proved to be $\mathcal{O}(d \log d + n_s \log n_s)$ where d is the number of Fourier coefficients that are needed for computing the Laurent coefficients at the desired accuracy (4.7) and n_s is the number of stable roots. As d is generally speaking much greater than n_s the complexity is actually $\mathcal{O}(d \log d)$.

We have left aside the discussion on the difficulties that might be encountered in (4.7) if γ becomes close to 1. The constant γ appears first in our definition of the annulus and is intended to separate the inner roots from the outer roots. From the error formula (4.7), we deduce that the method cannot converge for $\gamma = 1$ which corresponds to a root on the unit circle (no separation is possible).

One does not need such a bad case to observe convergence difficulties. The criterion used for fixing the number of Fourier coefficients d, different from (4.7), implies that a value of $d = 320$ is sufficient when $|z_i| = 0.9$, but it can be as high as 70 10^3 when $|z_i| = 0.9999$ for a polynomial of order 6 with 4 inner roots.

In practice and for the use that we anticipate, roots on the unit circle are unlikely, but the possibility of having roots really close to the unit circle is very real. Therefore despite its attractive complexity one should be aware of the drawbacks of this method.

4.1.3 Stabilizing the reflection coefficients

Reflection coefficients became popular during the 70's and 80's in the signal processing area. Speech recognition was a major topic and people were interested in finding filters that would enable them to perform speech synthesis. Most important to them was the stability of those filters. Hence the need for a simple, efficient method for analyzing the root locations of the denominator of the filter on the unit circle. As a consequence, reflection coefficients are only defined for discrete-time polynomials. There is a vast literature over reflection coefficients, also called *Schur parameters* [11, 2] as a reference to their use for characterizing the stability of polynomials. A noteworthy reference is [69].

Let us take a polynomial $a(z)$ which is not necessarily unstable. We define the *reversed* polynomial as

$$\tilde{a}(z) = \sum_{i=0}^{n} \tilde{a}_i z^{-i} = \frac{1}{a_0} z^{-n} a(z) \ , \tag{4.8}$$

with $\tilde{a}_0 = 1$. It has been shown in the literature [69, 31, 13] that it is possible to build two families of polynomials

$$\mathtt{A}_m(z) = \sum_{i=0}^{m} \mathtt{a}_{m,i} z^{-i} \qquad \mathtt{B}_m(z) = \mathtt{A}_\star(z) \tag{4.9}$$

satisfying the following properties

$$\langle \mathtt{A}_m(z), z^{-j} \rangle_{\mathcal{D}} = 0 \qquad j = 1, \ldots, m \tag{4.10}$$

$$\langle \mathtt{B}_m(z), z^{-j} \rangle_{\mathcal{D}} = 0 \qquad j = 1, \ldots, m \ , \tag{4.11}$$

Algorithm 4.1 Step Down procedure

$\mathtt{A}_n(z) = \frac{1}{a_0} z^{-n} a(z)$
for $m = n, \ldots, 1$ **do**

$$k_m = \mathsf{a}_{m,m}$$

$$\mathsf{a}_{m-1,i} = \frac{\mathsf{a}_{m,i} - k_m \bar{\mathsf{a}}_{m,m-i}}{1 - |k_m|^2} \qquad i = 0, \ldots, m-1 \ ,$$

end

where $\langle \cdot, \cdot \rangle_{\mathcal{D}}$ is a scalar product defined over n_z distinct points on the unit circle $\mathcal{D}$, i.e.

$$\langle f(z), g(z) \rangle_{\mathcal{D}} = \frac{1}{n_z} \sum_{i=1}^{n_z} f(z_i) \bar{g}(z_i) \ .$$

In other words (4.10) and (4.11) define orthogonal families of polynomials on the unit circle.

A recurrence can be built from the conditions (4.10) and (4.11) which can be used to build the families $\mathtt{A}_m$ and $\mathtt{B}_m$ [2, 69, 93]

$$\mathtt{A}_m(z) = \mathtt{A}_{m-1}(z) + k_m \mathtt{B}_{m-1}(z) \tag{4.12a}$$

$$z\mathtt{B}_m(z) = \bar{k}_m \mathtt{A}_{m-1}(z) + \mathtt{B}_{m-1}(z) \ . \tag{4.12b}$$

In these equations k_m is the *reflection coefficient* or *Schur parameter*. It is well known, see [69] for example, that a polynomial is stable if and only if $|k_i| < 1 \ \forall i$. Moreover, a result due to Cohn [93] has shown that the number of stable (resp. unstable) roots of $\mathtt{A}_m(z)$ is given by the number of positive (resp. negative) elements in the sequence $\{N_k\}_{k=1}^m$ defined by

$$N_k = \prod_{i=k}^{m} (1 - |k_i|^2) \ .$$

This result can also be deduced from Jury's test [12]. The recurrence (4.12) can be used both to compute the sequence of reflection coefficients k_i and to construct a polynomial from its reflection coefficients and the factors $\mathtt{A}_i$ and $\mathtt{B}_i$. Those two procedures are called the *step down* and *step up* procedures respectively. The recurrence can be rewritten in terms of the coefficients $\mathsf{a}_{m,i}$ of $\mathtt{A}_m(z)$. One then obtain algorithm 4.1 and algorithm 4.2 for the step down procedure and step up procedure respectively.

The use of those algorithms implies that one stores only the sequence of reflection coefficients k_m. Therefore, in order to stabilize a polynomial one only needs to compute the sequence of reflection coefficients $k_i, \ i = 1, ..., n$ using the recurrence (4.12) and modify the sequence following some chosen rule such as

$$\tilde{k}_i = \begin{cases} k_i & \text{if } |k_i| < 1 \ , \\ \beta k_i \ (\beta \in [0, 1[) & \text{if } |k_i| = 1 \ , \\ \frac{k_i}{|k_i|^2} & \text{if } |k_i| > 1 \ , \end{cases}$$

Algorithm 4.2 Step Up procedure

for $m = 1, \ldots, n$ **do**

$$\begin{aligned} &\mathtt{a}_{m,\,0} = 1; \\ &\mathtt{a}_{m,\,i} = \mathtt{a}_{m-1,i} + k_m \bar{\mathtt{a}}_{m-1,m-i} \qquad i = 1, \ldots, m-1\ , \\ &\mathtt{a}_{m,m} = k_m \end{aligned}$$

end
$a(z) = \frac{1}{\mathtt{a}_{n,n}} z^n \mathtt{A}_n(z)$

before using the sequence described by algorithm 4.2 to reconstruct a stable polynomial $p(z)$ from the modified sequence $\tilde{k}$. The complexity of the overall scheme is $\mathcal{O}(n^2)$ at most. One of the drawbacks is that the reconstructed polynomial $p(z)$ might be away from the original polynomial inside the coefficient space and thus its quality as a starting point can be put into question.

Another drawback is that stabilizing a polynomial using reflection coefficients, though pretty fast, is known to be subject to failure if $|k_m| = 1$ in algorithm 4.1. Polynomials with reciprocal roots and in particular those with roots on the unit circle will thus cause a failure in the step down procedure. A way around this difficulty is not straightforward but some propositions have been formulated, see [93].

As a final comment to this section, we point out the fact that the recurrence (4.12) has multiple uses and is linked to many engineering problems. For example, following the definitions (4.10) and (4.11) the families $\mathtt{A}_m$ and $\mathtt{B}_m$ belong to the class of *Szegö polynomials* orthogonal on the unit circle to which a rich literature is dedicated, see [2, 31, 93] and references therein.

4.2 Nearest Stable Polynomial Using Successive Convex Approximations

A new method for finding the nearest stable matrix to an unstable one was presented in Section 3.1. This section shows how we adapt the method for polynomials in continuous- and in discrete-time.

We will see that the changes that must be applied to the original formulation are mostly superficial and we will try to avoid unnecessary repetitions with the previous chapter. Therefore, the equations that we present here are an aggregation of the topics presented in Sections 3.1 and 3.2. In particular, the theory developed in Section 3.1.1 and leading to the construction of the Dikin ellipsoid and its inclusion in the stable set, remains the same.

The simplifications that occur when dealing with polynomials offer better complexity results.

4.2.1 Barrier Projection Method for polynomials

Our developments require the use of adapted notations. We recall that a monic polynomial

$$x(\lambda) = \lambda^n + \lambda^{n-1}x_{n-1} + \cdots + x_0$$

can be put into the form of a companion matrix. In this chapter we consider companion matrices given by

$$X(x) = Z - xe_1^T = \begin{bmatrix} -x_{n-1} & 1 & & \\ \vdots & & \ddots & \\ -x_1 & & & 1 \\ -x_0 & & & 0 \end{bmatrix} \tag{4.13}$$

where Z is, in this case, the upper shift matrix. Likewise we will denote by $A(a)$ the companion matrix A obtained from the vector a of coefficients following the construction (4.13). With these definitions, the Lyapunov and Stein operators associated with the companion matrix $X(x)$ become, respectively,

$$\mathcal{A}_P(x) \stackrel{\text{def}}{=} \mathcal{A}_P(X(x)) = X(x)P + PX^*(x) \ ,$$

and

$$\mathcal{A}_P(x) \stackrel{\text{def}}{=} \mathcal{A}_P(X(x)) = X(x)PX^*(x) - P \ .$$

Given a monic polynomial

$$a(\lambda) = \lambda^n + \langle a, \pi_\lambda^{n-1} \rangle_{\mathbb{C}}$$

of degree n whose vector of coefficients is given by $a \in \mathbb{C}^n$, the nearest stable polynomial problem is first written

$$\begin{array}{rl} \min\limits_{x \in \mathbb{C}^n, P \in \mathbb{C}^{n\times n}} & \dfrac{1}{2}\|A(a) - X(x)\|^2 \\ s.t. & \mathcal{A}_P(x) \prec 0 \\ & P \succ 0 \ . \end{array} \tag{4.14}$$

Companion matrices are a powerful tool to simplify the formulation for the nearest stable polynomial problem. Our resolution of (4.14) follows the one for matrices.

In a first step, we quickly review the successive steps that we implemented for the resolution of the problem. More time is taken to derive specific expressions associated with polynomials in the next section.

The idea behind the Augmented Barrier Projection Method is to create successive convex approximations of the stability region using Dikin ellipsoids. At each iteration, the choice of the Dikin ellipsoid we make fixes P through the use of Theorem 3.2, p.73, which implies that we solve the system of equations in P and Q given by (3.8) and (3.9) and which we rewrite here as

$$\Psi_{X^*}(Q^{-1}) + nI = 0 \ , \tag{4.15a}$$

$$\Psi_X(P) + Q = 0 \ . \tag{4.15b}$$

Here $\Psi_X(P)$ is either

$$\Psi_X(P) = XP + PX^* \ , \qquad \text{(continuous-time)}$$

or

$$\Psi_X(P) = XPX^* - P \ . \qquad \text{(discrete-time)}$$

The ellipsoids are built around their central point which is taken as the last iterate found in $X(x)$. A new update for $X(x)$ is obtained by solving a convex optimization problem. In order to state the corresponding convex optimization problem let us introduce the self-concordant barrier function $b_P(x)$ given by

$$b_P(x) \stackrel{\text{def}}{=} f(-\mathcal{A}_P(x)), \qquad f(Y) = -\ln\det Y.$$

Then, the use of companion matrices implies that the convex minimization problem (3.43) evolves into

$$\begin{array}{l} \min\limits_{h\in\mathbb{C}^n} \ \left[f_\mu(x+h) = \frac{1}{2}\|x+h-a\|_2^2 + \mu b_P(x+h)\right] \\ \qquad \langle b_P''(x)[h], h\rangle \leq \alpha \ , \end{array} \tag{4.16}$$

where, at this stage, the expression of $b_P''(x)[h] \in \mathbb{C}^n$ is still to be determined and $\mu, \alpha \in \mathbb{R}_+$ are parameters and α satisfies the additional constraint $\alpha < 1$.

When discussing the same method for matrices, we observed that the interest of this approach consisted in a much easier procedure for finding a new iterate both in X and P than for the original nonconvex formulation. Now, we observe that the resolution is even simpler in the case of companion matrices.

Indeed, the problem (4.16) supposes that we find a rank 1 correction matrix $H(h) = -he_1^T$ such that $X(x) + H(h)$ is the closest stable companion matrix within the Dikin ellipsoid. For the sake of conciseness, we denote the companion matrices $X(x)$ and $A(a)$ by X and A and the matrix $H(h)$ by H in the following. We can equally say that we are looking for a correction vector $h \in \mathbb{C}^n$ such that

$$\lambda^n + \langle x + h, \pi_\lambda^{n-1}\rangle_{\mathbb{C}}$$

is the closest stable polynomial to $a(\lambda)$ inside the convex domain defined by the Dikin ellipsoid.

An analytic solution for (4.16) can be found based on the first optimality conditions that are both necessary and sufficient :

$$x + h - a + \mu b_P'(x+h) + \lambda\,(b_P''(x)[h]) = 0 \ , \tag{4.17}$$

$$\frac{\lambda}{2}\,(\langle b_P''(x)[h], h\rangle - \alpha) = 0 \ , \tag{4.18}$$

$$\lambda \in \mathbb{R}_+ \ .$$

Note that in view of (4.17) and the fact that $x + h$ is stable and a is not, λ cannot be zero whenever μ is zero. Let us suppose for now that $\lambda > 0$; the case $\lambda = 0$ does not yield any additional difficulty since, following the developments below, the condition (4.17) reduces then to a linear system in h.

Algorithm 4.3 Augmented Barrier Projection Method (ABPM)

Choose a stable polynomial x_0, a tolerance $\epsilon_{tol} > 0$ and $\mu_K > 0$;
for $k \geq 0$ **do**

(a) Find P_k and Q_k by solving (3.8)-(3.11)
(b) Solve (4.22) to get λ_k;
(c) Find h_k by solving (4.20);
(d) $x_{k+1} = x_k + h_k$;
(e) **If** $f_{\mu_K}(x_k) - f_{\mu_K}(x_{k+1}) < \epsilon_{tol}$ **then** return **end**
end

When $\lambda > 0$, the complementary slackness condition (4.18) imposes that the optimal solution for h yields a point on the boundary of the ellipsoid since

$$\langle b''_P(x)[h], h\rangle - \alpha = 0 \ .$$

Furthermore, when $b'_P(x)$ is a linear operator, its Taylor expansion summarizes to

$$b'_P(x+h) = b'_P(x) + b''_P(x)[h] \ . \tag{4.19}$$

Let us assume for now that (4.19) holds. We will prove further that this is indeed the case both in continuous-time and in discrete-time.

Using (4.19), the optimal direction h can again be computed as the solution of a linear equation given by the first optimality condition (4.17). Indeed, we find

$$h(\lambda) = [I + (\mu+\lambda)B]^{-1} w_\mu \ , \tag{4.20}$$

where B is the matrix representation of $b''_P(x)[h]$ and w_μ accounts for all the terms independent of h in (4.17).

In order to find λ we can introduce (4.20) in the complementary slackness condition (4.18) to obtain an equation in λ

$$\langle b''_P(x)[h(\lambda)], h(\lambda)\rangle - \alpha = 0 \ . \tag{4.21}$$

Solving this last equation requires the use of the matrix representation B of $b''_P(x)$. As in the matrix case, it is real and positive definite, and can therefore be put into the diagonal form $B = UDU^T$. Thus, the equation (4.21) is equivalent to

$$\psi_\mu(\lambda) = \sum_i \frac{d_{B,i}\hat{w}^2_{\mu,i}}{(1+(\mu+\lambda)d_{B,i})^2} - \alpha = 0 \ , \tag{4.22}$$

where $\hat{w}_\mu = U^T(a - x - \mu b'_P(x))$.

We now have everything at hand to describe our algorithm which we do in algorithm 4.3. Its convergence proof is identical to the one obtained for the matrix version.

4.2.2 Detailed expressions for polynomials

In this subsection, we give detailed expressions for all the operators we have been using up to now. Unfortunately, the presence of the barrier term in the objective (4.16) makes it necessary to separate the continuous-time case from the discrete-time case.

Continous-time expressions In continuous-time, the expression of $b''_P(x)$ is given by the following proposition

Proposition 9. *The gradient and the Hessian of $b_P(x)$ are given by*

$$b'_P(x) = -2Q^{-1}p \ , \tag{4.23}$$

$$b''_P(x)[h] = 2Q^{-1}\left(hp^* + ph^*\right)Q^{-1}p \ , \tag{4.24}$$

respectively, where $p = Pe_1$.

Proof. The proof implies only basic calculus. Let us consider the first and second order derivatives of $\mathcal{A}_P(x)$ denoted by $D\mathcal{A}_P(x)[h]$ and $D^2\mathcal{A}_P(x)[h,h]$, respectively. In continuous-time, we have for the first order derivative

$$D\mathcal{A}_P(x)[h] = -hp^* - ph^* \ .$$

Its second order derivative is 0 everywhere so $D^2\mathcal{A}_P(x)[h,h] = 0$. As $D\mathcal{A}_P(x)[h]$ is an operator defined on $\mathbb{C}^n \to \mathbb{C}^{n\times n}$, its adjoint is defined as

$$D^*\mathcal{A}_P(x)[Y] = -2Yp \ , \tag{4.25}$$

in continuous-time. See Section 1.1.1 for details on how to compute the adjoint of an operator. The proof for the form of the Hessian is now a complete analog to the one derived in Theorem 3.3 for the matrix case. Since $\nabla f(Y) = -Y^{-1}$, it can be shown that $\nabla^2 f(Y)[H] = Y^{-1}HY^{-1}$, which together with the chain rule (see Section 1.2.2), gives

$$\begin{aligned}\langle b'_P(x), h\rangle &= \langle \nabla f(Q), -D\mathcal{A}_P(x)[h]\rangle \\ &= \langle Q^{-1}, -hp^* - ph^*\rangle \end{aligned} \tag{4.26}$$

$$\begin{aligned}\langle b''_P(x)[h], h\rangle &= \langle \nabla^2 f(Q)[D\mathcal{A}_P(x)[h]], D\mathcal{A}_P(x)[h]\rangle \\ &= \langle Q^{-1}D\mathcal{A}_P(x)[h]Q^{-1}, D\mathcal{A}_P(x)[h]\rangle \\ &= \langle D^*\mathcal{A}_P(x)[Q^{-1}(-hp^* - ph^*)Q^{-1}], h\rangle \ . \end{aligned} \tag{4.27}$$

At this point, all that remains to be done is to introduce (4.25) in the Hessian of the barrier (4.27) and obtain

$$\langle b''_P(x)[h], h\rangle = 2\langle Q^{-1}\left(hp^* + ph^*\right)Q^{-1}P, h\rangle \ .$$

□

Discrete-time expressions Our discussion in Section 4.2.1 makes it clear that h is the solution of a linear equation if $b'_P(x)$ is a linear operator (due to the fact that (4.19) must hold). Clearly $b'_P(x)$ cannot be a linear operator in discrete-time since $\mathcal{A}_P(x)$ is quadratic.

Therefore, our suggestion is to rewrite $\mathcal{A}_P(x)$ as an LMI which is again similar to the way we treated the discrete-time in the matrix case. As P is fixed in the resolution of the convex optimization problem (4.16), we write

$$M(x) = \begin{bmatrix} P & X(x)P \\ PX(x)^* & P \end{bmatrix} \quad \text{and} \quad R(h) = \begin{bmatrix} 0 & H(h)P \\ PH(h)^* & 0 \end{bmatrix} = \begin{bmatrix} 0 & hp^* \\ ph^* & 0 \end{bmatrix} .$$

Then the discrete-time stability constraint becomes

$$M(x) \succ 0 .$$

We can keep the formulation (4.16) unchanged provided we use a new definition of $b_P(x)$ namely

$$b_P(x) = \phi(M(x)) \qquad \phi(\cdot) \ : \ \mathbb{C}^{2n\times 2n} \to \mathbb{R} \ : \ \cdot \to -\log\det(\cdot) .$$

Proposition 10. *The gradient and the Hessian of $b_P(x)$ are given by*

$$b'_P(x) = -2Q^{-1}Xp ,$$

and

$$b''_P(x)[h] = 2Q^{-1}Xph^*Q^{-1}Xp + (p_{11} + p^*X^*Q^{-1}Xp)Q^{-1}h ,$$

respectively, where $p = Pe_1$ and $p_{11} = e_1^T Pe_1$.

Proof. The first and second derivatives of $b_P(x)$ follow the same type of developments as (4.26) and (4.27). Therefore we obtain

$$Db_P(x)[h] = \langle M^{-1}, R(h)\rangle ,$$

and

$$D^2b_P(x)[h,h] = \langle M^{-1}R(h)M^{-1}, R(h)\rangle .$$

We first compute the adjoint of $R(h)$ which is given by

$$R^*(S) = 2S_2p \tag{4.28}$$

where S is a Hermitian matrix belonging to $\mathbb{C}^{2n\times 2n}$

$$S = \begin{bmatrix} S_1 & S_2 \\ S_2^* & S_3 \end{bmatrix} .$$

In view of the adjoint (4.28), we need to find the corresponding block in M^{-1} and in $M^{-1}R(h)M^{-1}$. The developments are then analogous to the ones done for the matrix case which are found in Proposition 4. □

Expression for B The complete analogy between the polynomial and matrix case is pretty clear at this point. The main difference lies in the dimensions of the Hessian which is now an operator $b''_P(x) : \mathbb{C}^n \to \mathbb{C}^{n\times n}$. The expression of B introduced in (4.20) can be developed. In continuous- as well as in discrete-time, it is possible to find a real symmetric matrix $B \in \mathbb{R}^{n\times n}$ representing the operator $b''_P(x)$ such that

$$\begin{bmatrix} h_{re} \\ h_{im} \end{bmatrix} = \big[I + (\mu+\lambda)B\big]^{-1} \begin{bmatrix} w_{\mu,re} \\ w_{\mu,im} \end{bmatrix} \tag{4.29}$$

with B having the structure

$$B = 2\begin{bmatrix} T_{1,re} + T_{2,re} & -T_{1,im} + T_{2,im} \\ T_{1,im} + T_{2,im} & T_{1,re} - T_{2,re} \end{bmatrix} , \tag{4.30}$$

and where the subscripts re and im stand for the real and the imaginary part of the matrices. Now we can recall we have seen that the generic solution found for h (4.20) is a linear system of the form

$$w_\mu = [I + (\mu+\lambda)B]h ,$$

where w_μ accounts for all the terms in (4.17) which are independent of h. In continuous-time, using Proposition 9 we get

$$\begin{aligned} a - x + 2\mu Q^{-1}p &= h + 2(\mu+\lambda)\left[p^*Q^{-1}pQ^{-1}h + Q^{-1}ph^*Q^{-1}p\right] \\ &= h + 2(\mu+\lambda)\big[\underbrace{p^*Q^{-1}pQ^{-1}}_{T_1}h + \underbrace{Q^{-1}p(Q^{-1}p)^T}_{T_2}\bar{h}\big] . \end{aligned}$$

Similarly in the discrete-time case, we find using Proposition 10

$$\begin{aligned} a - x + 2\mu Q^{-1}Xp = h + 2(\mu+\lambda)[&\underbrace{\big(Q^{-1}(p^*X^*Q^{-1}Xp) + Q^{-1}p_{11}\big)}_{T_1}h \\ &+ \underbrace{Q^{-1}Xp(Q^{-1}Xp)^T}_{T_2}\bar{h}] . \end{aligned}$$

By taking the real and imaginary parts of T_1 and T_2, we find B using (4.30). By inspection, we remark that T_1 is Hermitian and T_2 is complex symmetric which enables us to conclude that B is real and symmetric as well. Hence, it is diagonalizable using orthogonal transformations as was sought in order to solve (4.22).

4.2.3 Numerical results

Benchmark

We can test our method on our benchmark (2.48) for which the unstable corresponding continuous-time polynomial is

$$a(s) = s^n - \nu_0 .$$

We specify once more that it is conjectured that the optimal solution is reached when $\nu_0 = 0$ yielding

$$x_*(s) = s^n \ .$$

We compare the solutions obtained by the polynomial algorithm with the solution obtained by the matrix version of the algorithm applied to the unstable companion matrix corresponding to $a(s)$. We take the same instantiation of the problem. Hence, we use the values $n = 4$, $\nu_0 = -0.1$ and a tolerance of $\epsilon_{tol} = 10^{-9}$, $\mu_K = 10^{-12} \frac{\|a - x_0\|^2}{b_P(x)}$ and $\gamma_\mu = 10$ for the stopping criterion.

We furthermore implement the same starting point $p(s)$ which is obtained by mirroring the unstable roots of $a(s)$ which yields two stable double roots

$$s_i = (0.1)^{1/4} \ e^{\pm j \frac{3\pi}{4}} \ .$$

The expression of the starting point is

$$\begin{aligned} p(s) &= s^4 - 4s^3(0.1)^{1/4} \cos(\frac{3\pi}{4}) + 4s^2(0.1)^{1/2} - 4s(0.1)^{3/4} \cos(\frac{3\pi}{4}) + 0.1 \ , \\ &= s^4 + 1.59s^3 + 1.26s^2 + 0.5s + 0.1 \end{aligned}$$

and the corresponding companion matrix is

$$C_p = \begin{bmatrix} -1.59 & 1 & & \\ -1.26 & & 1 & \\ -0.5 & & & 1 \\ -0.1 & & & 0 \end{bmatrix} \ .$$

The results of both methods are depicted in Figure 4.1. Let us denote by x_* the solution returned by the ABPM for polynomials and by X_* the result returned by the ABPM for matrices. The objective value is given by

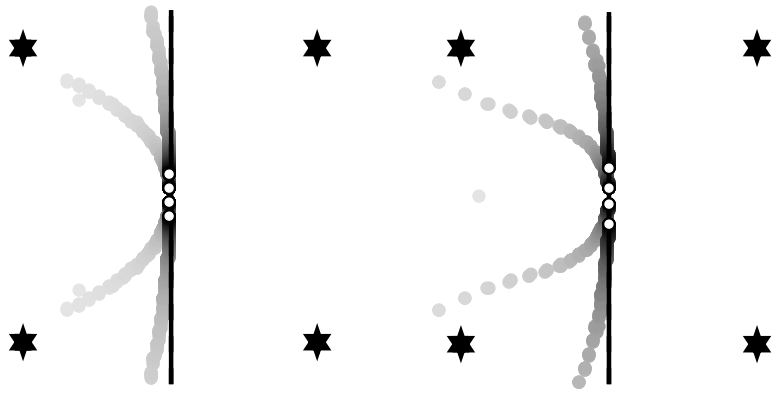
$$\|a - x_*\|_2 - 0.1 = 5\,10^{-4} \ ,$$

in 4304 iterations and 8.5 seconds for the polynomial version, and

$$\|A(a) - X_*\|_F - 0.1 = 1\,10^{-3} \ ,$$

in 1460 iterations and 4 seconds for the matrix version. It appears that the algorithm adapted for polynomials is performing better when started from the same point but takes more iterations. The difference of the performances can be explained.

Indeed, as our notations suggest and since the matrix case enjoys much more degrees of freedom than the polynomial one, the solution X_* obtained by the matrix version is not a companion matrix anymore and several elements which are zeros for a companion matrix are (really) small nonzero elements in X_*. All those small elements create the difference in precision between the two methods. On the other side, the advantage of the matrix version of the method is that it enjoys much more degrees of freedom and fewer iterations are required to reach a solution.



(a) (b)

Figure 4.1: Solutions attained by (a) the polynomial and (b) matrix versions of the ABPM algorithm for $n = 4$. The solution in (a) achieves an objective value of $\|a - x_*\|_2 - 0.1 = 5\,10^{-4}$ in 4304 iterations. The solution in (b) achieves an objective value of $\|A(a) - X_*\|_F - 0.1 = 1\,10^{-3}$ in 1460 iterations.

Randomly generated certificates

We have recalled at the beginning of this section that the Augmented Barrier Projection Method is intuitively understood as a succession of Dikin ellipsoids inside the set of stable systems on which we project the unstable system, either under the form of a matrix or a polynomial. Such a construction is dependent on the choice of a suitable P which can be seen as a stability certificate.

Back in Section 3.1.1, we have described our technique for choosing a P from the set of all acceptable stability certificates for our current point X : we take the analytic center of the set $\mathbb{P}_X$ defined as

$$\mathbb{P}_X \stackrel{\text{def}}{=} \{P \in \mathbb{K}^n_{++} \ : \ \Psi_X(P) \prec 0, \quad \langle P, I\rangle = 1\}. \tag{4.31}$$

The analytic center is obtained by solving the optimization

$$P_* \stackrel{\text{def}}{=} \arg\min\{b_X(P) \ : \ P \in \mathbb{P}_X\},$$

where

$$b_X(P) \stackrel{\text{def}}{=} f(-\Psi_X(P)), \qquad f(Y) = -\ln\det Y,$$

This choice is probably the best one but cannot be proved to be optimal since this would imply that the volume of the ellipsoid is maximal for our choice of P.

If our assumption that the analytic center of $\mathbb{P}_X$ is indeed the best choice then it surely yields better solutions, and thus better convergence, than randomly generated P's.

In order to illustrate this point, we carry a two-phase experiment based on an example from [71]. In the first phase, we generate 50 unstable perturbations of a discrete-time stable real polynomial given by

$$p(z) = z^4 - 2.7607z^3 + 3.8106z^2 - 2.6535z + 0.9238 \ ,$$

by applying a Gaussian noise distributed according to the normal law $\mathcal{N}(0, 10^{-2})$ on its coefficients vector

$$p^T = \begin{bmatrix} -2.7607 & 3.8106 & -2.6535 & 0.9238 \end{bmatrix} \ .$$

Those polynomials are then stabilized using our Augmented Barrier Projection Method. This gives us a baseline for our comparison. We give the distribution of the roots of the unstable polynomials in Figure 4.2a while the roots of the stabilized approximations are given in Figure 4.2b.

In its second phase, the experiment is carried as before, except that the stability certificates P will not be taken as the analytic center of $\mathbb{P}_X$, as explained above, but will be computed as follows. Take L_q as a random matrix and compute $Q = L_q L_q^*$. Clearly, Q is positive definite and it is regenerated at each iteration of each instance. Hence, solving

$$XPX^* - P = -Q \tag{4.32}$$

yields a P which is positive definite and can be used to define our ellipsoidal approximation. Such a procedure is expected to provide really small steps h since the randomly generated P's will not be *central* inside $\mathbb{P}_X$, and thus the associated Dikin ellipsoids are going to be very small.

The plots of the solution are shown in Figure 4.2c where it is clear that the solutions obtained are of lesser quality on average compared to Figure 4.2b. In the solution that we obtained, the mean distance is 10 times what it used to be. One will notice that some roots have barely moved from their original positions shown in Figure 4.2d. As expected, the steps leading to the solutions have been observed as really small, of the order of 10^{-6} on average and hence the number of iterations was really high, up to 50 10^3 for each instance.

Conclusion

We have managed to obtain a method to compute a solution *close* to a nearest stable polynomial to an unstable one from a theory primarily developed for matrices. The method benefits from the same convergence guarantees as the ones we have detailed for matrices in chapter 3. Accordingly, the complexity of the polynomial case is reduced compared to the matrix one. Nonetheless, its complexity is still in $\mathcal{O}(n^3)$ and intuitively faster methods can be derived but a few improvements should be carried if we want to speed up the algorithm 4.3.

Towards further improvements Exploiting the available structure of the convex optimization problem (4.16) has yielded significant computational gains

but the same cannot be said for the equations (4.15a) and (4.15b) which we rewrite here in continuous-time for a given companion matrix $X \in \mathbb{S}^n$

$$X(x)^* Q^{-1} + Q^{-1} X(x) + nI = 0 \ , \tag{4.33}$$

$$X(x)P + PX(x)^* + Q = 0 \ . \tag{4.34}$$

These two equations determine the choice of the Dikin ellipsoid through the variable P.

An algorithm that efficiently exploits the structure for those equations would also bring significant improvements in terms of computational complexity. Currently and independently of the structure of the matrix, the resolution of (4.33)-(4.34) requires the solution of two Lyapunov equations and one matrix inversion and the cost of each of those steps is in $\mathcal{O}(n^3)$. The goal would be to lower this complexity to $\mathcal{O}(n^2)$.

Fortunately, exploiting the structure of companion matrices embedded in Lyapunov equations has been of interest to some authors. In their paper [44] G. Heinig and U. Jungnickel developed a theory to get the solution S of an equation of the type

$$A(a)S + SB(b) = Y \ , \tag{4.35}$$

where $A(a)$ and $B(b)$ are the companion matrices associated with the two polynomials a and b and Y is a matrix of compatible size. Equation (4.35) has the same structure as (4.33). Another paper [8] from A. Betser, N. Cohen and E. Zeheb shows a better way of solving an equation of the type

$$\Psi_X(P) + Q = 0 \tag{4.36}$$

where $\Psi_X(P)$ is either continuous-time in which case it is given by (4.34), or discrete-time.

But despite this encouraging observation, those two papers do not make possible to lower the complexity of resolution of the system (4.33)-(4.34) for a few reasons. First, the two papers do not use compatible forms of companion matrices themselves. Though the theory developed in this section is compatible with any structure of companion matrices, this does not necessarily apply to these papers. It is not clear whether this difficulty can be surmounted and whether a framework which would unite these two theories for the resolution of the system (4.15) can be built.

Whatever the conclusion to this first question is, a second difficulty arises from one of the results obtained by Heinig and Jungnickel [44]. Indeed, they showed that a solution S to (4.35) can be decomposed into a sum of matrices

$$S = Z + K$$

where Z is the lower shift matrix and K is a Hankel matrix. If we suppose that an efficient way of reaching this solution can be found indeed, the open problem is then to find a way of inverting this matrix efficiently.

Because of these two problems we did not try to go further in this direction with our research and preferred to explore other possible algorithms. In particular, algorithms based on the roots of the polynomials could be a solution to solve the nearest polynomial problem while achieving a complexity of $\mathcal{O}(n^2)$ for each iteration.

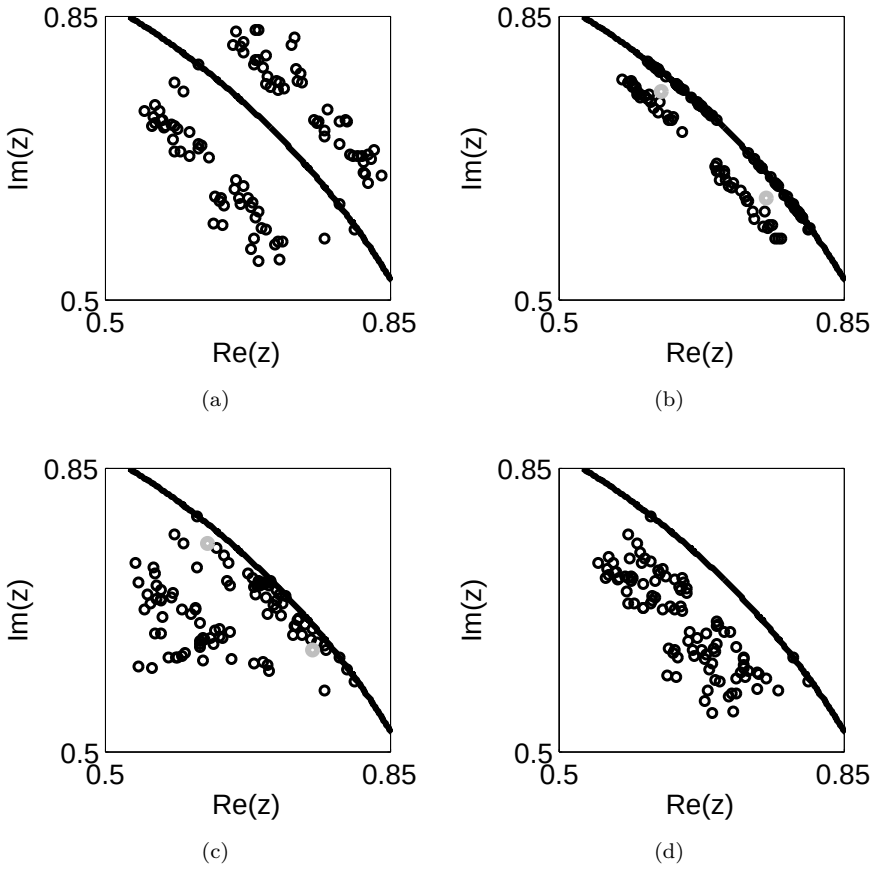


Figure 4.2: Comparisons of the effects of taking a stability certificate P as the analytic center of $\mathbb{P}_X$ (4.31) or as a solution of (4.32) where Q is random. Close-up of the distribution of the two upper right roots of the 50 polynomials (a) before stabilization, (b) after stabilization when P is the analytic center of $\mathbb{P}_X$. (c) features the solution obtained when Q is chosen randomly in (4.32). The roots of the 50 starting points as obtained using the mirrored method are represented in (d). The two corresponding original roots of $p(z)$ are plotted as light gray circles in (b) and (c).

4.3 Roots characterization based methods

Polynomials admit an easy characterization of their coefficients with respect to their roots. This is evident from the definition of a monic polynomial

$$p(\lambda; r) = \prod_{i=1}^{n} (\lambda - r_i) \ , \tag{4.37}$$

which can be evaluated through a simple linear recursion. Accordingly, the Jacobian of the polynomial with respect to the roots can also be evaluated from (4.37) very easily since the derivative of $p(\lambda)$ with respect to a root is another polynomial $\tilde{p}_j(\lambda; r)$ which is

$$\tilde{p}_j(\lambda; r) = -\prod_{i \neq j} (\lambda - r_i) \ .$$

The relation (4.37) is most useful to solve problems formulated in the coefficient space of the polynomials while working with the actual roots of the polynomial. We cover two methods that use this concept in this section, one for the continuous-time and one for the discrete-time.

The continuous-time method is due to an idea of Marc Van Barel and is an application of an optimization method developed in [91] to continuous-time problem for which an algebraic trick can be used. The discrete-time method was developed using the very same idea of root characterization of the problem but for discrete-time problems.

Though the two methods are developed on the principle that root optimization is applicable to the search of a nearest stable polynomial, their performances are by no means comparable. Indeed, the method for continuous-time problem uses a second order scheme while its discrete-time equivalent uses a first order scheme. Differences in the geometry of the stable sets in the spectral space and in the algebraic conditions that describe them in continuous- and in discrete-time will explain this difference in the performance.

4.3.1 Continuous-time Root Method

Instead of writing a constrained optimization problem the idea for continuous-time problems is to exploit the very simple description of the stable set. Let us represent each root as

$$r_i = -c_i^2 + j d_i \qquad \forall i \ , \tag{4.38}$$

where $c, d \in \mathbb{R}^n$ are taken as the variables of the problem. With this choice of variables the roots of each of our iterates will clearly be stable. Therefore, this parametrization implies that constraints that enforce stability of the roots become superfluous. Hence we now face an unconstrained optimization problem which is

$$\min_{c,d \in \mathbb{R}^n} \quad \|a - x(c, d)\|^2 \ , \tag{4.39}$$

where $a = [a(s)]_c$ and

$$x(c,d) = \left[\prod_{i=1}^{n}(s - r_i(c_i,d_i))\right]_c ,$$

and $r_i(c_i,d_i)$ is given by (4.38). The problem (4.39) is a nonlinear least square problem which is valid for real as well as for complex polynomials and on which the generalized Gauss-Newton method described in [91] applies. The convergence is quadratic in most cases.

As an illustration of the performance of this method we can look at the solution obtained when we apply it to our benchmark for the case $n = 4$ and $\nu_0 = 0.1$. The solution reached is

$$x_* \approx s^n ,$$

and the value of the objective function is $0.1 + \mathcal{O}(10^{-14})$ which means that the algorithm actually reached the conjectured solution.

4.3.2 Discrete-time Root Method

In discrete-time, the geometry of the stability region does not lend itself to a similar parametrization as the one obtained in continuous-time (4.38). A choice is thus to take the usual parametrization

$$r_i = c_i + jd_i , \tag{4.40}$$

where both c_i and d_i are elements of c and $d \in \mathbb{R}^n$. As pointed out by M. Van Barel [95], it would still be possible to use a polar parametrization of the roots to obtain an unconstrained optimization problem but this approach was not investigated at the time. Using the parametrization (4.40), the nearest stable polynomial problem becomes

$$\begin{aligned} &\min_{c,d} \quad [\|a - x(c,d)\|^2 = f(c,d)] \\ &s.t. \quad c_i^2 + d_i^2 \leq 1 \qquad \forall i . \end{aligned} \tag{4.41}$$

The objective function f is continuously differentiable everywhere on its domain. Therefore, the optimization problem (4.41) can be solved using successive first-order approximations of the problem augmented with quadratic regularization terms. As the problem is not separable in the real and imaginary parts of the roots, we must work with a full first order approximation of the function f at $(\hat{c},\hat{d})$

$$\phi(c,d) = f(\hat{c},\hat{d}) + \langle \nabla_c f, c - \hat{c}\rangle + \langle \nabla_d f, d - \hat{d}\rangle + \frac{L_1}{2}\|c - \hat{c}\|^2 + \frac{L_2}{2}\|d - \hat{d}\|^2 . \tag{4.42}$$

Based on (4.42), we write the optimization problem

$$\begin{aligned} &\min_{c,d\in\mathbb{R}^n} \ \langle \nabla_c f(\hat{c},\hat{d}), c - \hat{c}\rangle + \langle \nabla_d f(\hat{c},\hat{d}), d - \hat{d}\rangle + \frac{L_1}{2}\|c - \hat{c}\|^2 + \frac{L_2}{2}\|d - \hat{d}\|^2 \\ &\quad s.t. \quad c_i^2 + d_i^2 \leq 1 \qquad \forall i . \end{aligned} \tag{4.43}$$

Even though the optimization problem (4.43) is not separable in its real and imaginary part, it is separable and convex in each of the roots $r_i = c_i + jd_i$. Therefore, the first-order KKT optimality conditions are both necessary and sufficient and are, for all i,

$$\begin{aligned} \nabla_{c_i} f(\hat{c},\hat{d}) + L_1(c_i - \hat{c}_i) + 2\mu_i c_i = 0 \ , \\ \nabla_{d_i} f(\hat{c},\hat{d}) + L_2(d_i - \hat{d}_i) + 2\mu_i d_i = 0 \ , \\ \mu_i(c_i^2 + d_i^2 - 1) = 0 \ , \\ \mu_i \geq 0 \ . \end{aligned} \tag{4.44}$$

We deduce from (4.44) that any solution to the KKT conditions (4.44) must also satisfy

$$c_i = \frac{L_1\hat{c}_i - \nabla_{c_i} f(\hat{c},\hat{d})}{L_1 + 2\mu_i} \qquad \forall i \ , \tag{4.45}$$

$$d_i = \frac{L_2\hat{d}_i - \nabla_{d_i} f(\hat{c},\hat{d})}{L_2 + 2\mu_i} \qquad \forall i \ , \tag{4.46}$$

where $\mu_i \geq 0$ is found by solving

$$0 = \mu_i \left(\left(\frac{L_1\hat{c}_i - \nabla_{c_i} f(\hat{c},\hat{d})}{L_1 + 2\mu_i} \right)^2 + \left(\frac{L_2\hat{d}_i - \nabla_{d_i} f(\hat{c},\hat{d})}{L_2 + 2\mu_i} \right)^2 - 1 \right) \qquad \forall i \ . \tag{4.47}$$

Two different ways of solving (4.47) are available. Let us denote

$$\phi(\mu_i) = \left(\frac{L_1\hat{c}_i - \nabla_{c_i} f(\hat{c},\hat{d})}{L_1 + 2\mu_i} \right)^2 + \left(\frac{L_2\hat{d}_i - \nabla_{d_i} f(\hat{c},\hat{d})}{L_2 + 2\mu_i} \right)^2 - 1 \ .$$

It is clear that $\phi(\mu_i)$ is convex on the interval $\mathcal{D} = [\max(-L_1/2, -L_2/2), \infty]$ and therefore the equation $\phi(\mu_i) = 0$ has only one solution in $\mathcal{D}$ which is finite. In order to realize this, it suffices to observe that $\phi(\mu_i)$ decreases monotonically and is bounded below on $\mathcal{D}$ when L_1 and L_2 are positive. In particular, it tends to -1 from above as μ_i tends to infinity. As we are interested in positive values of μ_i we can thus use a Newton scheme to find the solution in this interval.

Alternately, a solution to the complementary slackness condition (4.47) can be found by considering the largest Lipschitz constant of the two, $L = \max(L_1, L_2)$ and solve the resulting quadratic expression in μ_i. Though very simple, this last technique yields less performing steps since then the expressions (4.45) and (4.46) share the same Lipschitz constant resulting in an oversmoothing for one of the variables at the benefit of the computation time.

In either case, we keep positive values of μ_i only and the process must be repeated for each of the roots.

Before presenting our algorithm, we discuss the criteria that must be satisfied by the Lipschitz constants L_1 and L_2. By definition, see Property 1.45 on page 18, $\phi(c,d)$ in (4.42) is a local upper approximation of $f(c,d)$ at $(\hat{c},\hat{d})$. The Lipschitz constants L_1 and L_2 must thus ensure the inequality

$$f(c,d) \leq \phi(c,d) \ . \tag{4.48}$$

The fulfillment of the condition (4.48) is obviously dependent on both Lipschitz constants, but there are many possible strategies to update them.

Typically, one doubles the value of L_1 and L_2 when (4.48) is violated but more evolved strategies can be used. Suppose we have a known pair $(\hat{c}, \hat{d})$ which we used to build $\phi(c, d)$ in (4.42) and a candidate for the update (c^+, d^+) that does not satisfy (4.48) such that

$$f(c^+, d^+) > \phi(c^+, d^+) \ . \tag{4.49}$$

One might look for the Lipschitz constant responsible of the violation of (4.48) by evaluating the two conditions

$$f(c^+, \hat{d}) > \phi(c^+, \hat{d}) \ , \tag{4.50}$$

$$f(\hat{c}, d^+) > \phi(\hat{c}, d^+) \ . \tag{4.51}$$

A positive result for (4.50) means that L_1 must be increased. The same conclusion holds for (4.51) regarding L_2. In all cases, the condition (4.49) has priority over both conditions (4.50)-(4.51) in the sense that if (4.49) is verified, at least one of the two Lipschitz constants must be increased whatever result is returned by the conditions (4.50)-(4.51).

Before we introduce our algorithm, let us point out that the gradients $\nabla_c f$ and $\nabla_d f$ computed with respect to c and d are both obtained from the gradient $\nabla_r f$ computed with respect to the complex root r. Indeed, following our definition of scalar product, we have that

$$\langle \nabla_c f, c - \hat{c} \rangle = \langle \nabla_r f, c - \hat{c} \rangle \ ,$$
$$\langle \nabla_d f, d - \hat{d} \rangle = \langle -j\nabla_r f, d - \hat{d} \rangle \ .$$

The Discrete-time Roots Method is given in algorithm 4.4.

4.3.3 Computation of the gradient

The importance of the efficiency in evaluating the function and its gradient must be stressed. Given the techniques developed earlier in this section, it is quite clear that the impact of the gradient evaluation in the overall computational cost of the algorithm 4.4 is substantial.

We are interested in computing the gradient with respect to a vector of roots r. Let us thus consider the expression for the objective function given by

$$f(r) = \frac{1}{2}\|a - x(r)\|^2 \ . \tag{4.52}$$

The directional derivative of the function f in a direction $h \in \mathbb{C}^n$ yields the expression

$$\langle \nabla f(r), h \rangle = \langle 2J_x^*(r)(x(r) - a), h \rangle \ , \tag{4.53}$$

where $J_x^*(r)$ denotes the Jacobian of the vector coefficients x with respect to the vector of roots r. In (4.53) the costly task is to determine the expression of the Jacobian $J_x(r)$. In particular, one might be interested in avoiding the

Algorithm 4.4 Discrete-time Roots Method (DRM)

Choose $r_0 = c_0 + jd_0,\ c_0, d_0 \in \mathbb{R}^n$, ϵ_{tol},
and take small L_1 and L_2. Take $k = 0$.

$\nabla_{c_0} f = \nabla_{r_0} f$;
$\nabla_{d_0} f = -j\nabla_{r_0} f$;
while $\sim (L_1\|c_{k+1} - c_k\|^2 + L_2\|d_{k+1} - d_k\|^2 < \epsilon_{tol})$ **do**

(a) Compute the vector μ_k using (4.47);
(b) Use (4.45) and (4.46) to compute tentative updates c^+ and d^+;
(c) Compute the updates $L_1^+,\ L_2^+$ using (4.48) ;
(d) **If** $(L_1, L_2) == (L_1^+, L_2^+)$ **then**

$c_{k+1} = c^+$;
$d_{k+1} = d^+$;
$\nabla_{c_{k+1}} f = \nabla_{r_{k+1}} f$;
$\nabla_{d_{k+1}} f = -j\nabla_{r_{k+1}} f$;

else

$(L_1, L_2) = (L_1^+, L_2^+)$

end
end

explicit computation of the Jacobian for the evaluation of (4.53) and prefer an implicit version of (4.53) which would save time and memory.

We have seen that the derivative of $x(\lambda; r)$ with respect to one of its roots is a new polynomial

$$\tilde{x}_j(\lambda; r) = -\prod_{i=1 i\neq j}^{n} (\lambda - r_i)\ . \tag{4.54}$$

The same polynomial can be obtained by dividing the original polynomial by the monomial $(\lambda - r_j)$

$$\tilde{x}_j(\lambda; r) = -\frac{x(\lambda; r)}{(\lambda - r_j)}\ . \tag{4.55}$$

We can thus either use (4.54) or (4.55) to compute the Jacobian $J_x(r)$. Another possible option consists in performing an implicit computation of $\nabla f(r) = J_x^*(r)(x(r) - a)$ which avoids the costly construction of $J_x(r)$. The remainder of this section is dedicated to the description and the analysis of the available techniques. We specifically reserve the notation $x(r)$ for our description of the nearest stable polynomial and the notation $x(r)$ is thus exclusively associated with the formulation (4.52). The more general notation $p(r)$ will denote any polynomial constructed from a vector of roots $r \in \mathbb{C}^n$. This difference in notations will underline the difference between simple methods which can be applied to any polynomial $p(r)$ and methods that apply to (4.52) only.

Algorithm 4.5 poly

$v_1 = \begin{bmatrix} 1 & -r_1 \end{bmatrix}$
for k = 2:n **do**
$v_k = \begin{bmatrix} v_{k-1} & 0 \end{bmatrix} - r_k \begin{bmatrix} 0 & v_{k-1} \end{bmatrix}$
end
$p(r) = v_n$

Algorithm 4.6 Horner

$q_{n-1} = 1;$
for $k = 1, \ldots, n-1$ **do**
$q_{n-(k+1)} = r_i q_{n-k} + p_{n-k};$
end

The simplest way for evaluating the coefficients of a polynomial $p(r)$ from its vector of roots r is to use the recurrence described in algorithm 4.5. This simple recurrence evaluates the vector of coefficients in $\mathcal{O}(n^2)$ operations. Therefore the cost of an evaluation of the Jacobian $J_p(r)$ using the recurrence used in algorithm 4.5 is $\mathcal{O}(n^3)$ since n calls to the routine must be performed. We will now analyse the cost of two other ways of evaluating the function and its Jacobian.

Deflation of polynomials

The literature on the deflation of polynomials is quite developed, see [26, 47, 90]. The goal is to suppress one of the factors of a polynomial, and as such it is also quite often associated with Horner's method.

Horner's method is based on the simple fact that a polynomial $p(\lambda) = \lambda^n + p_{n-1}\lambda^{n-1} + \cdots + p_0$ can be transformed into the remainder form

$$p(\lambda) = q(\lambda)(\lambda - \lambda_0) + R \tag{4.56}$$

where $q(\lambda)$ is a polynomial of degree $n-1$ called the quotient polynomial and $R = p(\lambda_0)$ is the remainder. In the case λ_0 is a root r_i of the polynomial, the remainder is zero and the whole equation can be rearranged to write the recurrence given in algorithm 4.6 which computes the gradient $\tilde{p}_i = q$ of the polynomial p with respect to the root r_i. Horner's algorithm given in algorithm 4.6 is stable for any roots r_i such that $|r_i| < 1$, but is also applicable when $|r_i| > 1$ provided we use a variant of the above. The operation count for the algorithm presented in algorithm 4.6 is $2n-1$. Hence, the complexity of a full Jacobian is $\mathcal{O}(n^2)$ which is better than the direct evaluation of the product.

Computation of the gradient using Algorithmic Differentiation

Algorithmic differentiation makes use of the simple idea that all functions as implemented on computers can be seen as the composition of elementary functions and, as such, can be derived using the chain rule. For example, consider any scalar function $z = f(y) \ : \ E \to \mathbb{R}$, and suppose that $f(y)$ can be expressed as

$$f(y) = (f_1 \circ f_2 \circ f_3)(y) \ ,$$

then the directional derivative of f is obtained by the chain rule

$$Df[h] = Df_1\Big[Df_2\big[Df_3[h]\big]\Big] \ ,$$

where only the directions of evaluations were indicated for clarity. Algorithmic differentiation applies this principle to a much larger scale. Algorithmic differentiation offers the possibility of computing the coefficients of a polynomial and its Jacobian in an *efficient* way. The basic idea is best illustrated using a small example. Suppose you want to compute the value of the function $f(u, v) = z(u, v) \ : \ \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ and $z(u, v) = uv$. The evaluation of the function can be represented using a tree as shown in Figure 4.3a. The value of z can be evaluated very simply from its two children. If we now want to evaluate both partial derivatives of f with respect to u and v, we write using the chain rule

$$\frac{\partial f}{\partial u} = \frac{\partial f}{\partial z} v \ . \tag{4.57}$$

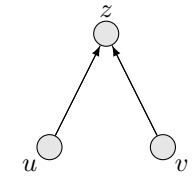
Again a tree representation of the process is possible, see Figures 4.3b and 4.3c. The value of a partial derivative is obtained from its parent node and its sibling in the tree. As the function f is directly proportional to the root z, we initiate the algorithm with the value $\frac{\partial f}{\partial z} = 1$. Finally, the strength of the method is that all three evaluations can be integrated in one tree which reuses some previous computations (if you suppose the u and/or v result from more unrepresented computations). In that case, the function is evaluated in a first sweep, bottom to top, called the *forward sweep*. A second sweep top to bottom is then performed to evaluate the derivatives. This second sweep is called the *reverse sweep*.

We apply the same principles to a larger tree which will enable us to evaluate the value of the objective function

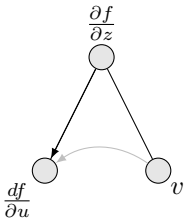
$$f(r) = \frac{1}{2}\|a - x(r)\|^2 \ ,$$

and its gradient. We underline the fact that the principles of algorithmic differentiation can be applied to any algorithm implemented on a computer, but the algorithm that we will develop is tailored to our needs.

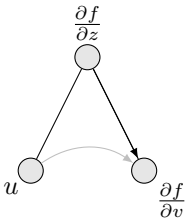
Many introductory [39, 38, 72, 5] and more advanced [40, 10] articles and books cover the topics of algorithmic differentiation and automatic differentiation. More features are available for computing a function and its gradient than the very basic principle presented here. In this thesis, the term algorithmic differentiation will refer to the search of an appropriate differentiation scheme



(a) Evaluation of the function



(b) Evaluation of $\frac{\partial f}{\partial u}$



(c) Evaluation of $\frac{\partial f}{\partial v}$

Figure 4.3: Evaluation of the function and its derivative using a tree

for a specific algorithm. It therefore contrasts with automatic differentiation where the ultimate goal is to develop software that is able to automatically differentiate a user-designed function using black-box methods. This last topic goes outside the scope of this thesis, but the interested reader can look further into existing software, see [43] for an example.

Evaluating the objective function and its gradient jointly Consider the objective function

$$f(r) = \frac{1}{2}\|a - x(r)\|^2 \ . \tag{4.58}$$

A binary tree can be drawn to evaluate the function $x(r) \in \mathbb{C}^{n+1}$ from its vector of complex roots $\hat{r}$ which we suppose throughout to be known and to be, without loss of generality, of size $n = 2^K$ for some positive integer K. The objective function can then be evaluated by simply applying (4.58) to the result.

Let us detail such a binary tree which we represent in Figure 4.4. Each leaf i contains the vector $x_{K,i}$ of coefficients of its corresponding monomial $x_{K,i} = [(\lambda - \hat{r}_i)]_c$. As we climb up the tree, we convolve all pairs of sibling polynomials $x_{k,i}$ and $x_{k,i+1}$, in a forward sweep until we reach the root of the tree where x_0 is the desired vector of coefficients

$$x_0 = x(\hat{r}) \ .$$

The algorithm 4.7 evaluates the value of the coefficients of a polynomial and the value of the objective function associated to it. Once all the nodes in

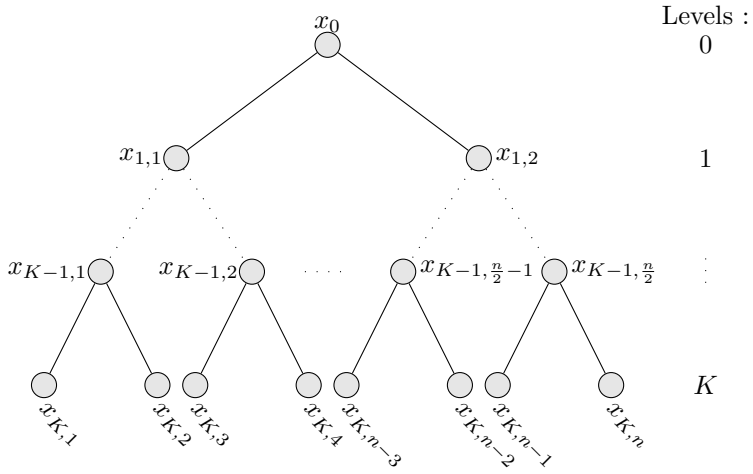


Figure 4.4: Binary tree used for the evaluation of the coefficients of the polynomial and for the evaluation of the Jacobian. For clarity, the variables used for the computation of the Jacobian are not present on the figure.

---

the tree in Figure 4.4 have been evaluated and the value of f is known, we can perform a reverse sweep of the tree to obtain the gradient $\nabla f(\hat{r})$. The principles that we presented in the Figures 4.3b and 4.3c apply to our representation of the problem with two small differences.

First, the value of the variable $\tilde{x}_0$ is initialized at the beginning of the reverse sweep with the value

$$\tilde{x}_0 = \nabla_x f(\hat{r}) = x_0 - a$$

while it was logically 1 in our introductory example. Second, by definition the ith node at the level k in the reverse trees, labelled $\tilde{x}_{k,i}$, must contain the gradient of the objective function with respect to itself, i.e.

$$\tilde{x}_{k,i} = \nabla_{\tilde{x}_{k,i}} f(\hat{r}) \ .$$

As each node is the result of the convolution of two polynomials, we must compute the derivative of a convolution.

Let us take an example to show how to build such a derivative and let us consider the polynomial $p \in \mathbb{C}^{n+1}$ evaluated in the forward sweep. Let us assume that the same node which contains p also contains the gradient $\tilde{p} = \nabla_p f$. Let us look at how we need to compute the derivative of p with respect to its children. Let us suppose that those children are given by p_1 and p_2. The polynomial p is obtained from a convolution of p_1 and p_2

$$p(p_1, p_2) = p_1 * p_2 = C(p_1)p_2 = C(p2)p_1 \ , \tag{4.59}$$

where $p_1 \in \mathbb{C}^{m+1}$, $m < n$, $p_2 \in \mathbb{C}^{n-m+1}$ and, for a polynomial $q \in \mathbb{C}^{l+1}$,

Algorithm 4.7 Forward Sweep

```
% Height of the tree
K = log_2(n);
x_{K,i} = [λ − r̂_i]_c;   ∀i
% Going through each level in the tree
for k = K-1,...,0 do
    % Number of nodes at the current level
    u = 2^k;
    % Going through each node at the current level
    for i = 1,...,u do
        % Convolution of the polynomials
        x_{k,i} = x_{k+1,2i-1} * x_{k+1,2i};
    end
end
f = (1/2)‖a − x_0‖^2;
```

$C(q) \in \mathbb{C}^{n+1\times n-l+1}$ is a Toeplitz matrix defined by

$$C(q) = \begin{bmatrix} q_{1,l} & & \\ \vdots & \ddots & \\ q_{1,0} & & q_{1,l} \\ & \ddots & \vdots \\ & & q_{1,0} \end{bmatrix} .$$

In order to facilitate our work we can take $C_1 = C(p_1)$ and $C_2 = C(p_2)$ as constant matrices and write the convolution (4.59) as a function of a single variable

$$g_1(p_1) = C_2 p_1 \ , \tag{4.60}$$

or

$$g_2(p_2) = C_1 p_2 \ , \tag{4.61}$$

and switch between the two when necessary.

Then, the directional derivatives of p in a direction h with respect to p_1 and p_2 respectively is given by

$$\begin{aligned} D_{p_1} f(\hat{r})[h] &= \langle \nabla_p f(\hat{r}), Dg_1(p_1)[h] \rangle \\ &= \langle C_2^* \tilde{p}, h \rangle \\ &= \langle \nabla_{p_1} f, h \rangle \ , \\ D_{p_2} f(\hat{r})[h] &= \langle \nabla_p f(\hat{r}), Dg_2(p_2)[h] \rangle \\ &= \langle C_1^* \tilde{p}, h \rangle \\ &= \langle \nabla_{p_2} f, h \rangle \ . \end{aligned}$$

Algorithm 4.8 Reverse Sweep

$\tilde{x}_0 = x_0 - a$
for k = 1,...,K **do**
  $u = 2^k$;
  for i = 1:2:u **do**
    $\tilde{x}_{k,i} = C^*(x_{k,i+1})\tilde{x}_{k-1,\frac{i+1}{2}}$;
    $\tilde{x}_{k,i+1} = C^*(x_{k,i})\tilde{x}_{k-1,\frac{i+1}{2}}$;
  end
end
$\nabla_r f = [\tilde{x}_{K,i}]_{i=1,\ldots,n}$

Therefore, we finally obtain

$$\tilde{p}_1 = \nabla_{p_1} f = C_2^* \tilde{p} \tag{4.62}$$
$$\tilde{p}_2 = \nabla_{p_2} f = C_1^* \tilde{p}\ . \tag{4.63}$$

The reader can easily check the compatibility of the dimensions with those of p_1 and p_2. The matrix products (4.62) and (4.63) do not form a convolution or a deconvolution and the use of a matrix product such as this one implies an operation count of $\mathcal{O}(n^2)$.

Still, the same operations could be carried out with the use of FFT matrices for which the asymptotic complexity would be much more favorable, of the order of $\mathcal{O}(n \log n)$. But the complexity gains with the FFT do not materialize before we reach sizes greater than $n = 50$ which is already outside the range of sizes we are able to deal with.

With the above results, we can eventually write down the algorithm 4.8 for the reverse sweep.

Complexity of the reverse mode of algorithmic differentiation The performance and the relevance of the presented scheme can be assessed by computing the complexities of each of the proposed algorithms, the forward sweep detailed in algorithm 4.7 and the reverse sweep in algorithm 4.8.

A similar development as the one we do now could be carried for the FFT version of the algorithm. As we did not implement this version, we choose to develop the complexity analysis for the method which used convolution matrices. Still we will comment the complexity of the FFT version at the end of this analysis.

At each level k in the tree, the *forward* sweep, implemented in algorithm 4.7, requires 2^k convolutions of size $\frac{n}{2^{k-1}}$ of polynomials of size $\frac{n}{2^k}$. If we consider that a convolution of size n is carried with a matrix product of the type (4.60) or (4.61)[1] which implies that we perform $\frac{n}{2^k}$ times $2\frac{n}{2^{k+1}}$ scalar products and

[1]This is not the optimal way of performing a convolution but enables comparison later in the text

additions, the operation count sums up into

$$\text{work}\{\text{f}\} = \sum_{k=0}^{K-1} 2^k \left(\frac{n}{2^k}\right)^2 . \tag{4.64}$$

We have neglected the cost of constructing the convolution matrix in (4.64) but it can be supposed that the cost of constructing the convolution matrix is proportional to its number of elements. Therefore, this would imply only a change of the coefficient in (4.64). After translating the counter of 1 and using the identity

$$\sum_{k=1}^{m} y^k = y\frac{1-y^m}{1-y} , \tag{4.65}$$

the count (4.64) yields

$$\text{work}\{\text{f}\} \approx 2n^2 .$$

With the use of the FFT, the complexity is logically reduced to $\mathcal{O}(n \log n)$.

The same approach can be adopted for computing the complexity of the accumulation of the gradient as performed by the *reverse* sweep in algorithm 4.8. The number of convolutions is the same as in the forward sweep. The difference lies in the size of the matrix products. As the adjoint of the convolution matrix has a size $\frac{n}{2^k} \times \frac{n}{2^{k-1}}$ each vector matrix product yields an operation count (multiplication and addition) of $\left(\frac{n}{2^{k-1}}\right)^2$. Therefore the total complexity is

$$\begin{aligned}\text{work}\{\nabla f\} &= \sum_{k=1}^{K} 2^k \left(\frac{n}{2^{k-1}}\right)^2 \\ &\approx 4n^2\end{aligned}$$

Again an implementation of the reverse sweep using FFT matrices would also asymptotically lower the complexity to $\mathcal{O}(n \log^2 n)$ and therefore the total complexity for the accumulation of the Jacobian using FFT matrices would be in $\mathcal{O}(n \log^2 n)$.

The spatial complexities of the forward and the reverse sweep can also be assessed. In the forward sweep, the intermediate variables of the forward sweep must all be kept for the computation in the reverse sweep. Hence its spatial complexity is $\mathcal{O}(n \log n)$. In the reverse sweep now, the values located at a level k are only used for one computation at the level k. A vector of length n thus suffices to store the variables of interest. Hence the spatial complexity of the reverse sweep is $\mathcal{O}(n)$.

Horner or algorithmic differentiation, what's best ? The use of FFT in the algorithmic differentiation scheme we have suggested would lower the complexity to $\mathcal{O}(n \log n)$ making it superior to the use of Horner's method.

Still, for small sizes we have seen that the operation counts are really close to each other. In practice, Horner's method appears to be faster than the algorithmic differentiation scheme above as Figure 4.5 suggests.

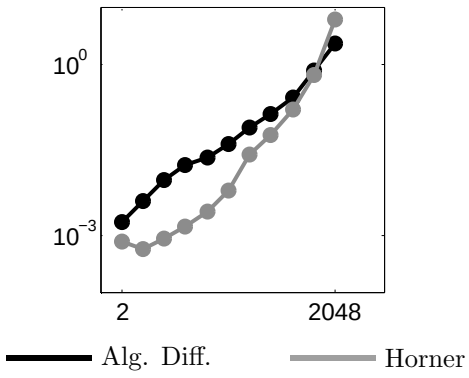


Figure 4.5: Evolution of the averaged computation times of 10 gradients of the objective function (4.58) versus the degree of the polynomial. Time is in seconds.

4.4 Comparison of the methods

We start by introducing two additional methods used for our comparisons but which were not part of our investigations in the framework of this thesis.

4.4.1 Nearest stable polynomial using reflection coefficients

We have seen at the beginning of this chapter that it was possible to modify the sequence of reflection coefficients associated to a polynomial $a(z)$ in order to get a stable polynomial $p(z)$.

The idea can be pursued further to obtain a method for finding the nearest stable polynomial in discrete-time. This was first developed by Moses and Liu [71] who noticed two important features.

First, reflection coefficients of a polynomial $a(z) \in \mathbb{R}^n$ form the unit hypercube in $\mathbb{R}^n$ given by $R = \{k_m \in [-1, 1] \quad m = 1, \ldots, n\}$, provided marginally stable solutions are acceptable. Therefore, the domain of the reflection coefficients is convex and bounded.

Second, each element a_i of $a(z)$ is a multilinear function of the reflection coefficients through the step up procedure given in algorithm 4.2. Therefore, the minimization problem

$$\begin{aligned} \min_x \quad & \|a - x\|^2 \\ s.t. \quad & x \in \mathrm{cl}(\mathbb{S}^n) \end{aligned}$$

is reformulated into

$$\begin{aligned} \min_{k_1,\ldots,k_n} \quad & [\|a - x(k_1, \ldots, k_n)\|^2 = f(k_1, \ldots, k_n)] \\ s.t. \quad & k_i \in [-1, 1] \quad \forall i \end{aligned}$$

where it appears that $f(k_1, \ldots, k_n)$ is a quadratic function of each of the k_i's on the domain $[-1, 1]$. The approach suggested in [71] is to optimize alternatively on each k_i by minimizing the quadratic function

$$g(k_i) = f(k_1, \ldots, k_n)\Big|_{k_{j,j\neq i}} = \alpha k_i^2 + \beta k_i + \gamma \ .$$

The constants α, β and γ are determined each time by interpolating the function $g(k_i)$ at the points $k_i = 1$, $k_i = -1$ and $k_i = 0$. This yields a system of three equations into the unknowns α, β and γ which can be solved. The optimal value of k_i is then found by finding the minimal value of $g(k_i)$. Each iteration of the algorithm has a complexity in $\mathcal{O}(n^2)$.

The results in [71] show that this approach works remarkably well and is pretty fast. As an alternative optimization scheme is used, convergence of the algorithm is not guaranteed but, in most instances, some actions can be taken to avoid that it gets stuck.

4.4.2 Hanso

As in the matrix case, we will also compare the performance of our methods with Hanso introduced briefly in Section 3.4.1. The algorithm is a first order method and thus needs the value of the function and its gradient at a point. In this case the objective function is

$$\min_x \quad [f(x) = \|a - x\|] \ ,$$

whenever x is stable and

$$\min_x \quad [f(x) = \|a - x\| + \mu\gamma(x)] \ ,$$

whenever x is unstable and where $\gamma(x)$ is either the root abscissa $\alpha(x)$ or the root radius $\rho(x)$ depending on the type of problem we face. The gradient $\nabla f(x)$ is computed from the directional derivatives for $\gamma(x)$ given by the equations (1.48) and (1.49) which we have seen in the first chapter. According to those expressions, the gradient of f is either

$$\begin{aligned}
\nabla f(x) &= \frac{x-a}{\|a-x\|} && \text{if } x \text{ is stable,} \\
\nabla f(x) &= \frac{x-a}{\|a-x\|} + \mu\bar{\xi}_i && \text{if } x \text{ is unstable and } a \text{ is continuous-time,} \\
\nabla f(x) &= \frac{x-a}{\|a-x\|} + \mu\frac{\lambda_i}{|\lambda_i|}\bar{\xi}_i && \text{if } x \text{ is unstable and } a \text{ is discrete-time,}
\end{aligned}$$

where

$$\xi_i = \frac{\pi_{\lambda_i}^{n-1}}{\prod_{j\neq i}(\lambda_i - \lambda_j)}$$

The default parameters of Hanso are otherwise used.

4.4.3 Performance and accuracy of the methods

In order to evaluate the performance of our algorithms we have tested continuous-time and discrete-time algorithms on several examples. As their behavior varies depending on the difficulty of the examples we chose four examples that we think representative of both difficult and easy problems. Hence the experiments were carried out on

- the benchmark (2.48) (continuous-time);
- Perturbation of a real polynomial with imaginary unit roots (discrete-time);
- Real random continuous-time polynomials;
- Real random discrete-time polynomials.

We did not carry out experiments with complex polynomials for two main reasons. First, we designed our experiments so that we could test as many algorithms as possible in the same conditions and the method of Moses cannot be transposed to complex polynomials. Second, the study of complex cases showed that the behavior of the algorithms were practically identical to the real case.

Let us make a quick review of the examples we tested.

Benchmark

Our benchmark is an ideal example of a difficult continuous-time instance. The notations are the following. The unstable polynomial is given by

$$a(s) = s^n - \nu_0 \ ,$$

and we adopt the value $\nu_0 = 0.1$. The conjectured globally optimal solution is given by

$$x_*(s) = s^n \ .$$

Perturbation of a real polynomial with imaginary unit roots Because multiple roots located on the unit circle in the spectral space correspond to a singularity on the boundary of the stable set in the coefficient space. We first construct a stable discrete-time polynomial $x_*(z) \in \mathbb{C}^{n+1}$ which has a single pair of complex conjugate roots repeated multiple times. Its expression is

$$x_*(z) = (z+j)^{\frac{n}{2}}(z-j)^{\frac{n}{2}} = z^n + z^{n-2} + \cdots + z^2 + 1 \ . \tag{4.66}$$

An unstable instance $a(z)$ is then generated by perturbing the vector of coefficients. More specifically, we fixed the expression of $a(z)$ to

$$a(z) = z^n + z^{n-2} + \cdots + z^2 + (1 + \nu_0) \ , \tag{4.67}$$

with a value of $\nu_0 = 0.1$. This choice of direction and amplitude for the perturbation is arbitrary but it is analogous to the type of perturbation chosen

for the benchmark and its amplitude makes $x_*(z)$ a likely solution to the problem, which is an interesting possibility. Furthermore, we have seen repeatedly that multiple eigenvalues are very sensitive to perturbations. Therefore, the direction and the magnitude of the perturbation appears to create unstable instances whose nearest stable polynomial is difficult enough to find.

Real random polynomials In order the offer a comprehensive analysis of the performances of the methods described in this thesis we have also generated unstable random real polynomials of sizes between $n = 4$ and $n = 32$ from a uniform distribution of the coefficients on the interval $[0, 1]$ to test both the continuous- and discrete-time case.

Unstable discrete-time real random polynomials generated from a uniform distribution have their roots naturally spread around the unit circle both in angle and in modulus. We cannot conjecture on the stable solution in this case, and it is therefore more complicated to evaluate the accuracy of the methods even though the random cases are typically easier to process since the polynomials are not located at a singularity of the stable set in the coefficient space.

Different polynomials were generated to carry out the experiments in continuous- and in discrete-time. It is worth noting that we generated only one polynomial to test all the methods for a case (continuous or discrete) and a size so that all methods are supposed to achieve the same optimal value of the objective function for a given size.

Results In continuous-time, we can compare three approaches

- the Augmented Barrier Projection Method (ABPM);
- the Continuous Root Method (CRM);
- Hanso (HANSO) .

In discrete-time, four algorithms could be applied, those are

- the Augmented Barrier Projection Method (ABPM);
- the Discrete Root Method (DRM);
- Hanso (HANSO),
- The optimization over the reflection coefficients (MOSES)

The final solution reached by each algorithm is denoted generally by x_f.

The same criteria that were used in the matrix case can be used here. After a study of the different criteria it appears that the MEAN ACCURACY does not bring additional insight for polynomials and we thus discard it. As for matrices our measures of performance depend on whether we have a conjecture on the solution that we should attain. In the affirmative case we use the expression

$$\text{GAP} = \|x_f - a\| - \|x_* - a\| \tag{4.68}$$

along with the execution time taken by each algorithm. Otherwise, we use the *relative gap* with respect to a suboptimal solution x_{so} which is defined as

$$\text{Rel. Gap} = \frac{\|x_f - a\|}{\|x_{so} - a\|} \ , \tag{4.69}$$

as a measure of performance. The criteria (4.69) is somehow a normalization of the first criteria (4.68). Indeed, the denominator of (4.69) represents the distance of a suboptimal stable polynomial to a. Therefore, the Rel. Gap indicates, independently of the size of the problems, the relative improvement that the solution x_f achieves relative to x_{so}.

The nature of x_{so} changes from one example to the other. In continuous-time, we take $x_{so}(s) = s^n$ which has all its roots at the origin. In discrete-time, we construct x_{so} by projecting any unstable roots on the spectral stability boundary. In both cases, x_{so} is a suboptimal solution which can be used in the criterion (4.69).

Each algorithms has its own stopping criterion. Therefore we must take a different parametrization for each method. As in the matrix case, we chose the values after several trials and they were chosen so that the execution time and the precision of the method as returned by either (4.68) or (4.69) were balanced. Hanso and CRM are used with their default parametrization. DRM quits whenever the norm between two successive vectors of roots is smaller than 10^{-6}. ABPM stops whenever the improvement of the objective function is smaller than $\epsilon_{tol} = n\,10^{-8}$ ($\gamma_\mu = 10\,n$, $\epsilon_\mu = n\,10^{-10}$) while Moses stops when given two successive sequences of reflection coefficients k

$$\max_j |k_j^{\{i\}} - k_j^{\{i+1\}}| < 10^{-3} \qquad \text{for some } i \ ,$$

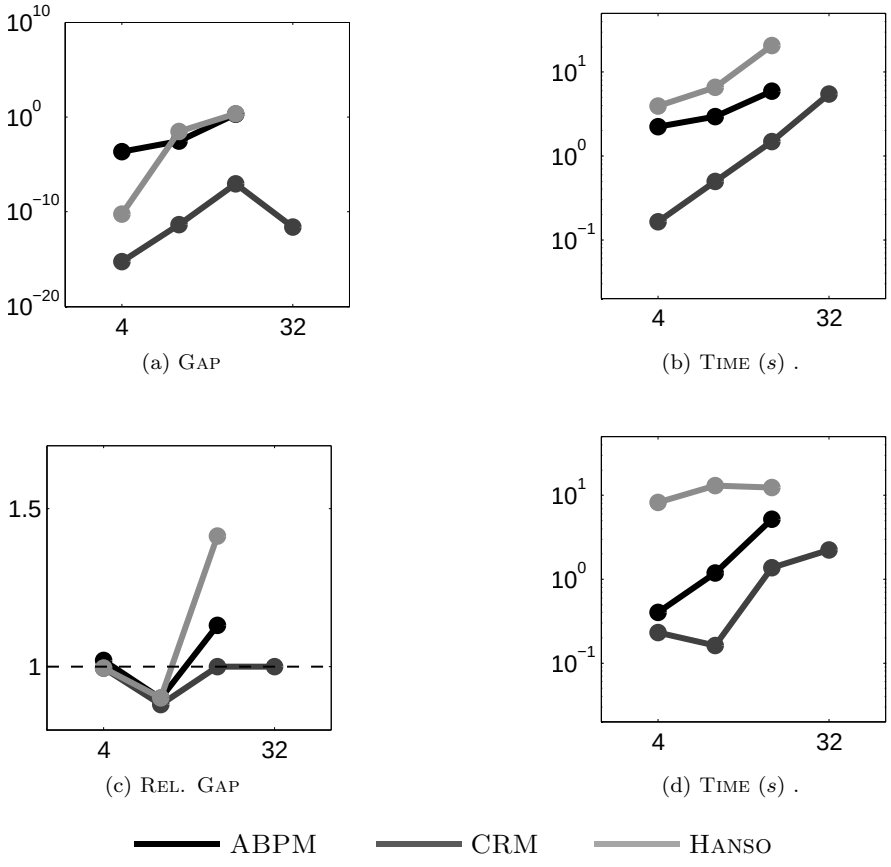
where the superscript $\{i\}$ denotes the i^{th} iteration. Though this seems a much looser tolerance compared to the others, this value seems to produce a good precision-time ratio which is not the case when the tolerance is taken as 10^{-6} for Moses.

We are now ready to analyse the results of our experiments. For each type of polynomial we created four experiments of sizes taken as powers of 2 between 4 and 32. The results are given in Figure 4.6 for continuous-time algorithms and in Figure 4.7 for the discrete-time ones. Let us start by commenting on the results obtained for the continuous-time case.

We give two general comments before making a short analysis on the performance of the various methods and the reasons for their success and failures.

A first comment can be made from the overall look of Figure 4.6 and 4.7. There is a clear difference in performance between the cases deemed difficult and the ones that are deemed easy. This can be observed directly from the associated computation times which have a tendency to be much higher for the difficult cases compared to the easy ones.

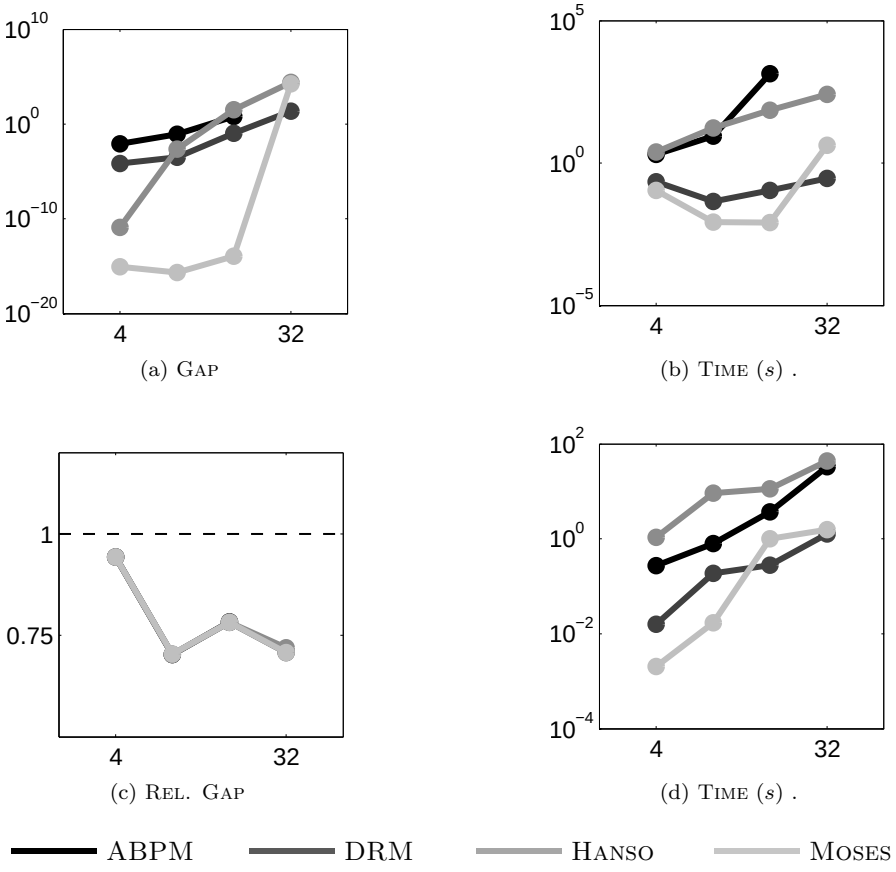
The second general comment is focused on the conclusions that one can draw from the Figures 4.6c and 4.7c which represent the Rel. Gap criterion. As explained above, the relative gap is a measure of performance with respect to another admissible solution x_{so}. Therefore, the continuous-time methods



(a) Gap (b) Time (s) .

(c) Rel. Gap (d) Time (s) .

Figure 4.6: Evolution of the performance of the continuous-time algorithms with respect to the sizes of the instances. (a) and (b) Benchmark,(c) and (d) Real random polynomials. Failed experiments are not shown. Data is given in Tables 4.1 and 4.2.



(a) Gap (b) Time (s) .

(c) Rel. Gap (d) Time (s) .

Figure 4.7: Evolution of the performance of the discrete-time algorithms with respect to the sizes of the instances. (a) and (b) Real polynomials with imaginary unit roots,(c) and (d) Real random polynomials. Failed experiments are not shown. The curves in (c) are superposed due to the fact that their solutions are close to each other. Data is given in Tables 4.3 and 4.4.

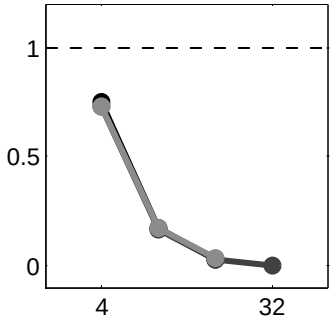


Figure 4.8: Illustration of the effects that the use of the REL. GAP criterion implies. This graphic was generated using the same data as Figure 4.6c except that the REL. GAP criterion is now computed with respect to a stable polynomial which was created from the stable roots of $a(s)$ and the imaginary part of the unstable roots of $a(s)$.

in Figure 4.6c only appear to perform badly because the polynomial $x_{so} = s^n$ always achieve an equally good distance to the unstable polynomial $a(s)$. A completely different conclusion could have been drawn if x_{so} had been chosen differently as Figure 4.8 suggests.

Another comment can be made on the various performances achieved by the algorithms. In continuous-time, it is clear in Figure 4.6 that CRM performs much better than the two other methods since the curves related to it are systematically below the other curves whatever measure of performance we consider. This was expected since the method deals with an unconstrained problem whilst both ABPM and HANSO suffer from various numerical complications which caused their failures for $n = 32$. ABPM suffers from the bad conditioning of the matrices that defines the Dikin ellipsoid which prevents it from working for sizes greater than $n = 20$. It was already observed in the matrix case that a size limit existed and had no reason of improving with polynomials for which the conditioning of the problem is usually worse due to lesser number of degrees of freedom. In the case of HANSO, it appears to have more difficulties with polynomials than with matrices due to the penalty coefficient it uses in front of the root abscissa to enforce stability of the polynomials. Therefore it must repeat the process several times which increases substantially its computation time and sometimes prevents it to find an adequately stable polynomial as for $n = 32$. This drawback could maybe be corrected with another parametrization of the software but this is unsure.

In discrete-time,we observe in Figure 4.7c and Figure 4.7d that all methods achieve equally good solutions and are distinguishable only from their computation times. The MOSES algorithm appears to be the best one on these two examples but it must be stressed once again that it cannot handle complex polynomials.

	Time [s]			Gap (4.68)		
Size	ABPM	CRM	Hanso	ABPM	CRM	Hanso
4	2.2	0.2	4.0	0.00023	5.4e-016	6e-011
8	3.0	0.5	6.6	0.0028	4.3e-012	0.03
16	6.0	1.5	21	2.1	8.7e-008	2.1
32	-	5.5	-	-	2.6e-012	-

Table 4.1: Performance data of the continuous-time methods on the benchmark. The data is represented in Figures 4.6a and 4.6b.

	Time [s]			Rel. Gap (4.69)			$\|a - x_{so}\|$
Size	ABPM	CRM	Hanso	ABPM	CRM	Hanso	
4	0.4	0.2	8.2	1	1	1	1.4
8	1.2	0.2	13	0.9	0.88	0.9	1.5
16	5.2	1.4	12	1.1	1	1.4	4.2
32	-	2.2	-	-	1	-	5.3

Table 4.2: Performance data of the continuous-time methods on real random continuous-time polynomials. The data is represented in Figures 4.6c and 4.6d.

Conclusion on the results Comparing methods is challenging since every different example can lead to different results. Our experiments have put into light the advantage of using the CRM methods in continuous-time for solving the nearest polynomial problem. Its accuracy and complexity makes it the safest choice.

In discrete-time, the experiments described here tend to give an advantage to the Moses algorithm. Still, it is not always true and we have experimented cases (which we do not report here) where the Moses algorithm does not behave so well.

Still our observation is that the best results are not obtained by the algorithms which work in the coefficients space directly but by those which use alternative parametrization of the space such as the roots of the polynomial or the reflection coefficients to solve the nearest stable polynomial problem.

4.5 Introduction to weighted objective function

Finally, we now talk about a last feature that can be further added to the problem. In system identification, one needs to carry out numerous experiments and collect a significant amount of data to identify, and then validate, the model of a system. As often in that case, the variables of use are frequencies and the

	Time [s]			Gap (4.68)		
Size	ABPM	CRM	Hanso	ABPM	CRM	Hanso
4	2.0	0.2	2.4	0.0079	6.5e-005	1.3e-011
8	8.9	0.0	17	0.083	0.0003	0.0022
16	1.4e+003	0.1	71	6.7	0.11	33
32	-	0.3	2.6e+002	-	22	2.5e+004

Table 4.3: Performance data of the continuous-time methods on real discrete-time polynomials with imaginary unit roots. The data is represented in Figures 4.7a and 4.7b.

	Time [s]			Rel. Gap (4.69)			$\|a - x_{so}\|$
Size	ABPM	CRM	Hanso	ABPM	CRM	Hanso	
4	0.3	0.0	1.1	0.94	0.94	0.94	1.2
8	0.8	0.2	9.3	0.7	0.7	0.7	1.3
16	3.7	0.3	11	0.78	0.78	0.78	2.4
32	33	1.3	43	-	0.71	0.72	5

Table 4.4: Performance data of the continuous-time methods on real random discrete-time polynomials. The data is represented in Figures 4.7c and 4.7d.

response of the system is more relevant at some frequencies than at others. In this prospect, it is valuable to be able to add user-defined weights in the objective function in order to underline the importance of those frequencies. Such work was at the core of Tom D'haene's thesis [32, 33] who successfully implemented this kind of weighting to get

$$\min_{\theta} \frac{1}{m} \sum_{i=1}^{m} w_i^2 \|G(\Omega_i) - H(\Omega_i, \theta)\|^2 \tag{4.70}$$

where G is the transfer function of the original unstable system, H is the stable transfer function that we are seeking and can be modified through the vector θ of parameters and Ω_i is the excitation frequency. As weights are linked to frequencies, their introduction in our formulation would require several adaptations, the least being to work with the transfer function $H(\lambda)$ of the system rather than the state matrix A only,

$$H(\Omega) = C(\Omega I - A)^{-1} B + D \ ,$$

where Ω is either s or z depending on the type of problem we face.

In its current form the Augmented Barrier Projection Method is not able to deal with a weighted objective function and its design allows it to handle the Frobenius norm only. In the following we thus perform a first step towards the inclusion of a weighted objective function in our formulation. The goal is to see whether the $\mathcal{H}_2$-norm can be changed into a Frobenius norm in which case it is quite probable that the algorithm could be modified to handle such cases as well.

We will now try to develop the problem in some simplified setting. Let us assume that the problem is such that we only try to stabilize the continuous-time polynomial denominator $a(s)$ of the transfer function G over $m \geq n$ samples at various frequencies and that we use squared weights. As usual, let us denote by $x(s)$ the unknown denominator of H. Recall the $n+1$ dimensional vector

$$\pi_{j\omega_i}^n = \begin{bmatrix} (j\omega_i)^n & \dots & j\omega_i & 1 \end{bmatrix}^T \ .$$

Then for a vector of weights $w \in \mathbb{R}^m$ the objective function (4.70) can be written down as

$$\begin{aligned}
& \min_x \frac{1}{m} \sum_{i=1}^{m} w_i^2 \|\langle a - x, \pi_{j\omega_i}^n \rangle_{\mathbb{C}}\|^2 \\
= \ & \min_x \frac{1}{m} \sum_{i=1}^{m} w_i^2 \langle (\pi_{j\omega_i}^n)^T (a - x), (\pi_{j\omega_i}^n)^T (a - x) \rangle \\
= \ & \min_x \frac{1}{m} \sum_{i=1}^{m} w_i^2 \langle \bar{\pi}_{j\omega_i}^n (\pi_{j\omega_i}^n)^T (a - x), (a - x) \rangle \\
= \ & \min_x \langle (\frac{1}{m} \sum_{i=1}^{m} w_i^2 \bar{\pi}_{j\omega_i}^n (\pi_{j\omega_i}^n)^T)(a - x), (a - x) \rangle \\
= \ & \min_x \langle S(a - x), a - x \rangle \ ,
\end{aligned}$$

where we posed

$$S = \frac{1}{m}\sum_{i=1}^{m} w_i^2 \bar{\pi}_{j\omega_i}^n (\pi_{j\omega_i}^n)^T \ .$$

Our formulation of the above makes the Cholesky factorization of $S = LL^*$ obvious and yields the final formulation

$$\begin{aligned} &\min_{x\in\mathbb{C}^n} \ \|L^*(a-x)\|^2 \\ &s.t.\ x \in \mathrm{cl}(\mathbb{S}^n) \end{aligned} \tag{4.71}$$

The advantage of a formulation such as (4.71) is that it clearly has a similar form compared to the problems we have solved throughout this chapter. Therefore, it appears that minor changes in our resolution might enable us to solve (4.71) adequately.

The same principles apply if we now want to apply a filter to cut uninteresting frequencies. The difference with the above expressions is that we now treat the case where the weighting is continuous, under the form of a filter, rather than discrete where we specified some specific weights for some interesting frequencies.

In order to simplify our notations, we will note $e = a - x$ the vector of coefficients of the error. We are interested in minimizing the $\mathcal{H}_2$-norm of the error which is given in continuous-time by

$$\|e\|_{\mathcal{H}_2}^2 = \int_{s=j\mathbb{R}} e(s)e(s)^* d\mu(s) \ , \tag{4.72}$$

where $e(s) = e^T s$ and $\mu(s)$ is an appropriate measure function which incorporates the desired weighting. In discrete-time, (4.72) becomes

$$\|e\|_{\mathcal{H}_2}^2 = \oint_{|z|=1} e(z)e(z)^* d\mu(z) \ ,$$

with the same definition for μ. Let Γ denote the boundary of the stable set, i.e. $\Gamma = j\mathbb{R}$ in continuous-time and $\Gamma = e^{j\omega}$, $\omega \in [-\pi, \pi]$ in discrete-time. Accordingly, let π_λ^n denote either $\pi_{j\omega}^n$ or $\pi_{e^{j\omega}}^n$. Then, for some appropriate measure function $\mu(\lambda)$, we can reformulate the $\mathcal{H}_2$-norm into

$$\begin{aligned} \|e\|_{\mathcal{H}_2}^2 &= \int_\Gamma e^* \bar{\pi}_\lambda^n (\pi_\lambda^n)^T e d\mu(\lambda) \\ &= e^* \left(\int_\Gamma \bar{\pi}_\lambda^n (\pi_\lambda^n)^T d\mu(\lambda) \right) e \\ &= e^* M e \\ &= \|L^* e\|_F^2 \ . \end{aligned}$$

The matrix $M = LL^*$ does not have the same structure in continuous- and in discrete-time. In continuous-time, we have

$$\begin{aligned} M_{ik} &= \frac{1}{2\pi}\int_{-\infty}^{\infty} -(j\omega)^i (j\omega)^k d\mu(j\omega) \\ &= \int_{-\infty}^{\infty} -(j\omega)^{i+k} d\mu(j\omega) \end{aligned}$$

Provided the weight function is uniform on all frequencies, the matrix M appears to be a Hankel matrix, called the moment matrix

$$M = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_n \\ c_1 & \iddots & & \iddots & \vdots \\ c_2 & & \iddots & & \vdots \\ \vdots & \iddots & & & \vdots \\ c_n & \cdots & \cdots & \cdots & c_{2n} \end{bmatrix} .$$

In discrete-time, for a uniform weight function on all frequencies, the moment matrix takes the expression

$$\begin{aligned} M_{ik} &= \oint_{|z|=1} (z)^{-i} z^k dz \\ &= \frac{1}{2\pi} \oint_{-\pi}^{\pi} e^{j(k-i)\omega} d\omega \ . \end{aligned}$$

which yields a Toeplitz matrix

$$M = \begin{bmatrix} c_0 & c_1 & \cdots & c_n \\ \bar{c}_1 & \ddots & \ddots & \\ \vdots & \ddots & \ddots & c_1 \\ c_n & \cdots & \bar{c}_1 & c_0 \end{bmatrix} .$$

Clearly, $M \in \mathbb{K}_+^n$ both in continuous-time and in discrete-time since it is Hermitian and

$$e^* M e = \|e\|_{\mathcal{H}_2} \geq 0 \qquad \forall e \in \mathbb{C}^n \ .$$

We conclude that the formulation (4.71) also works for the $\mathcal{H}_2$-norm.

The same type of result can be obtained again for matrices as the following shows for the discrete-time case. Indeed, take $E = A - X$ and let $E(z) = E - Iz$ be the transfer function denominator.

$$\begin{aligned} \|E\|_{\mathcal{H}_2} &= \mathrm{Tr} \oint_{|z|=1} E(z)^* E(z) d\mu(z) \\ &= \mathrm{Tr} \oint_{|z|=1} (E - zI)^* (E - zI) d\mu(z) \\ &= \mathrm{Tr} \left(\begin{bmatrix} I & -E^* \end{bmatrix} \begin{bmatrix} c_0 I & c_1 I \\ \bar{c}_1 I & c_0 I \end{bmatrix} \begin{bmatrix} I \\ -E \end{bmatrix} \right) \ . \end{aligned}$$

Again, we have

$$\begin{bmatrix} c_0 I & c_1 I \\ \bar{c}_1 I & c_0 I \end{bmatrix} \in \mathbb{K}_+^{2n} \ , \tag{4.73}$$

which yields, using the Cholesky decomposition LL^* of (4.73),

$$\|E\|_{\mathcal{H}_2} = \left\| L^* \begin{bmatrix} I \\ -E \end{bmatrix} \right\|_F \ .$$

It is clear from our developments that the nearest stable system in the sense of the $\mathcal{H}_2$-norm, both in the matrix and the polynomial case, can be transposed in a form that fits our formulations. Therefore it is quite likely that our algorithm could process such cases provided some changes are brought to the algorithm to handle the extra matrices.

We conclude with a final comment about applying this technique to the full transfer function rather than just the denominator of the transfer function. Working with the transfer function would require that we fully revise the approach we adopted to solve the nearest stable system problem. Additional complications will occur which are as much opportunities. For instance, working with transfer functions imply taking pole-zero cancellations into account, but adding a certain number of zero would offer us the opportunity of using them to our advantage to cancel some undesired frequencies for example. Despite the fact that it is tractable in theory, we did not explore this problem further in this thesis but, once again, those results are available to some extent in [33].

Conclusions on chapter 4

This chapter presents our main achievements regarding the nearest stable polynomial problem.

We have first highlighted the existence of a number of techniques for the stabilization of polynomials. Those techniques are inherited from research in various areas : numerical linear algebra, speech processing and, naturally, control. Though they offer viable techniques for determining starting points, most of those methods cannot be extended into a method for finding the nearest stable polynomial to an unstable one.

We have thus analysed several new methods for finding the nearest stable polynomial to an unstable one and have compared them on numerical examples to illustrate their strength and weaknesses. Each of them has its particularities and we summarize them now.

The Augmented Barrier Projection Method is fully adaptable and could allow several enhancements such as additional constraints on the location of the roots of a polynomial. Another strength of the method is that it always produces solutions that are strictly inside the stability region. But its complexity harms its usefulness and only very small polynomials can be treated. Still the method levels up with others on simple cases such as random polynomials.

Methods based on an optimization of the roots proves to be the fastest methods discussed in this thesis. In continuous-time, the unconstrained formulation that is obtained is efficient both in time and accuracy due to the fact that a Newton-like approach can be applied. Unfortunately, the discrete-time equivalent cannot enjoy the same advantages and it uses a gradient scheme. Nonetheless, work on the complexity of the gradient computation yields satisfactory complexity results and the method behaves well in practice.

Other methods found in the literature behave well too. The method of Moses which uses the reflection coefficients of the polynomial has a step complexity which makes it very attractive but its convergence can be quite slow

in practice due to the use of an alternating optimization scheme. Still, the method behaves well on most cases. The main inconvenience of this method is that it is usable for real discrete-time polynomials only.

Finally, the use of the root abscissa and root radius in Hanso has proved to be a viable approach. The complexity and convergence are those of a BFGS scheme and thus make it one of the most interesting methods to solve the problem. As the root abscissa and radius have a matrix version, Hanso is fully adaptable. Still on polynomials the choice of a good value for the penalty coefficient is not easy and can sometimes hamper its efficiency.

We have concluded our research with a short discussion on the use of a weighted objective function which would offer a better control of the frequencies of the poles of the system. We have shown that the $\mathcal{H}_2$-norm of a polynomial could be easily reformulated as a weighted Frobenius norm thereof achieving a first step towards the resolution of such problems.

Conclusion

This thesis studies the nearest stable system problem to its core. We explore the complexity of the problem and use variational tools to report and analyze the types of nonsmoothnesses and nonconvexities that appear on the boundary of the stable set. We use the structural properties of self-concordant functions, construct several reformulations of the problem and exploit the unique structure of companion matrices to offer innovative algorithms to solve the problem. We prove their convergence when needed.

We sum up our contribution into three main topics : algorithms, reformulations and the set of stable systems.

Algorithms Our work contains the original description of two methods for matrices and two methods for polynomials which substantially expand the number of optimization methods available to the user for the resolution of the nearest stable system. These methods cover all the possible variations of the problem : for polynomials or for matrices, in continuous-time or in discrete-time, real or complex.

In all logic, finding *the* nearest stable system is not a realistic goal. Indeed, let us recall that the stable set is an open set and unless one takes into account the closure of the set there is no *nearest* system. Even when the closure of the stable set is taken into account, finding the nearest stable system requires a *global* solver for nonsmooth nonconvex optimization problems and there is no such an algorithm for the time being. Our algorithms thus find *a* nearest stable system or an approximate solution to a nearest stable system.

We have exploited the mathematical properties of self-concordant functions to build convex approximations of the nonconvex feasible set described by the Lyapunov inequalities. The resulting algorithm, the Augmented Barrier Projection method, has the ability to deal with any of the variations of the problem as demonstrated by many examples. In the future, the method could be adapted to suit specific needs such as the placement of the poles of the system in a specific region of the complex plane. This would be particularly useful to the identification community where such a feature is desirable. We proved the feasibility of our method and stated several convergence results which showed that the method performs at least as well as the damped Newton method described in [73] in theory. Still, the algorithmic complexity of one step of the method makes the method suitable for small problems only. Nonetheless, the iterates are always guaranteed to be strictly stable. This feature is certainly

one of the strong argument in favor of this method.

In order to reduce the algorithmic complexity, we successfully investigated a reformulation of the problem in continuous-time on which simple but efficient optimization methods are applied and provide fast convergence to an approximate solution. This was made possible by shifting the nonconvexity of the feasible set into the objective function and by seeking the best problem structure possible. We further developed one of the reformulations characterized by the simplicity of its feasible set. This simple feasible set offered an ideal framework for the gradient mapping method. The introduction of additional regularization terms in the reformulation enabled us to bound the level sets and prove convergence results.

Efficient algorithms for polynomials have to exploit the inherent structure associated with them. In a first step, we have modified the Augmented Barrier Projection Method to deal with the companion matrix associated with a polynomial. Though very attractive in theory, the difficulties associated with this approach were underlined and we have put into light the current superiority of alternative optimization methods that use the roots of the polynomial as optimization variables. The last approach proved to be very efficient both in continuous-time and in discrete-time.

The optimization methods that we developed for matrices and polynomials alike, have compared quite favorably with existing methods found in the literature.

Reformulations In chapter 3, we have developed several reformulations of the nearest stable matrix in continuous-time. Only one of those formulations has been fully developed so far. Other reformulations that are particularly suitable to a Newton scheme have been highlighted and promise to yield better algorithms in the near future. Those reformulations are an original contribution of this thesis and show the advantage that suitable reformulations can provide for the resolution of the specific case for which they have been developed. Further research could be directed towards the search of reformulations for other variations of the problem. The feasibility of this approach is not always guaranteed since the structure of the constraints that describe the stable set is not identical for continuous-time and discrete-time problems.

Many problems in optimization are nonconvex but in most cases researchers do not seek or do not find reformulations that could be easily handled. Our work can thus inspire people looking for such reformulations.

Stable set Our research led us to explore the geometry of the stable set and the intrinsic link between the eigenvalue structure of a matrix and the singularities of the stable set. The fact that results associated with this topic are little known prompted us to produce a short survey of the literature. The survey gives an insight of the complexity that surrounds the nearest stable system problem and contains multiple references for people interested in the topic. In particular, we have recalled bounds on the distance between two matrices and have illustrated how the bounds depends on the specific structures of the matrices. We have also reported various results that highlight some types of

singularities appearing on the boundary of the stable set due to its nonsmooth nonconvex nature. Finally, we have summarized necessary optimality conditions for the nearest matrix problem and have linked it to our work through a continuous-time benchmark. This benchmark is especially interesting since we have been able to use all the results in the survey to lay a conjecture on one of its locally optimal solution.

From an optimization point of view, many aspects could improve this work in the future. In the section on the continuous-time root method (CRM), we have seen the advantage that unconstrained minimization problems could provide since they are much more easy to handle than constrained ones. It should be possible to write an unconstrained minimization problem for discrete-time polynomials using the polar decomposition of the roots. In the same way, some matrix problems with a particular structure could be parametrized so as to yield unconstrained minimization problems as well. This would be the case if the expected solution is a normal matrix for example. Also, we have already mentioned that the research of a reformulation of the nearest discrete-time matrix problem such as the one we did for the continuous-time case in section 3.3 had the potential of yielding an efficient algorithm which would complement the already existing one for the continuous-time case. Further research could also be undertaken on other variations of the nearest stable system problem. The replacement of the Frobenius norm by other norms such as the 1-norm or the $\mathcal{H}_2$-norm is of particular interest. Even though it is not clear how it could be obtained, such an algorithm would tend to yield sparse solutions.

At a conceptual level, the work presented in this thesis could evolve to fit additional purposes. In particular, we did not focus on the control point of view. Further research could emphasize the usefulness of our methods by investigating efficient techniques for solving the nearest stable problem with additional constraints such as pole placement. We could also look at replacing the stability constraint by other ones; enforcing passivity of a system should be a realistic goal. We have introduced briefly the possibility of assigning weights into the objective function in order to improve the relevance of the solution. Following this idea, the quality of the solutions and the usefulness of the methods could also be improved by taking the zeros of the transfer function into account as well. This would be interesting to benefit from cancellation effects between the poles and the zeros of the transfer function and it would yield more relevant solutions.

In conclusion, this thesis contains an overview of the available techniques for solving a problem which has frequently been neglected due to its intrinsic complexity. The various approaches used throughout will provide new tools and inspiration for the resolution of closely related problems in a field in constant evolution and full of challenges.

Bibliography

[1] Pierre-Antoine Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

[2] G.S. Ammar, W.B. Gragg, and L. Reichel. Downdating of Szegö polynomials and data-fitting applications. *Linear Algebra and its Applications*, 172:315–336, July 1992.

[3] V.I. Arnold. On matrices depending on parameters. *Russian Mathematical Surveys*, 29, 1971.

[4] László Balogh and Rik Pintelon. Stable Approximation of Unstable Transfer Function Models. *IEEE Transactions on Instrumentation and Measurement*, 57(12):2720–2726, December 2008.

[5] Michael Bartholomew-Biggs and Steven Brown. Automatic differentiation of algorithms. *Journal of Computational and Applied Mathematics*, 124:171–190, 2000.

[6] Dennis S. Bernstein. Robust static and dynamic output-feedback stabilization: Deterministic and stochastic perspectives. *IEEE Transactions on Automatic Control*, 32(12):1076–1084, December 1987.

[7] Dimitri Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.

[8] A Betser, N. Cohen, and E. Zeheb. On solving the Lyapunov and Stein equations for a companion matrix. *Systems & Control Letters*, 25(3):211–218, June 1995.

[9] Rajendra Bhatia. *Perturbation bounds for matrix eigenvalues*. Longman Scientific and Technical, pitman res edition, 1987.

[10] Christian H. Bischof, H. Martin Bücker, Paul Hovland, Uwe Naumann, and Jean Utke. *Advances in automatic differentiation*. Springer, 2008.

[11] Yuval Bistritz. A circular stability test for general polynomials. *Systems & control letters*, 7(April):89–97, 1986.

[12] Yuval Bistritz. Reflections on schur-cohn matrices and jury-marden tables and classification of related unit circle zero location criteria. *Circuits, systems, and signal processing*, 15(88):111–136, 1996.

[13] Yuval Bistritz, Hanoch Lev-Ari, and Thomas Kailath. Immitance-Type Three-Term Schur and Levinson Recursions for Quasi-Toeplitz Complex Hermitian Matrices. *SIAM Journal on Matrix Analysis and Applications*, 12(3):497–520, July 1991.

[14] Vincent. D. Blondel, Mert Gurbuzbalaban, Alexander Megretski, and Michael L. Overton. Explicit Solutions for Root Optimization of a Polynomial Family With One Affine Constraint. *IEEE Transactions on Automatic Control*, 57(12):3078–3089, December 2012.

[15] Stephen Boyd and V Balakrishnan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its $L\infty$-norm. *Systems & Control Letters*, 15(1):1–7, July 1990.

[16] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[17] James V. Burke, Didier Henrion, Adrian S. Lewis, and Michael L. Overton. Stabilization via Nonsmooth, Nonconvex Optimization. *IEEE Transactions on Automatic Control*, 51(11):1760–1769, November 2006.

[18] James V. Burke, Adrian S. Lewis, and Michael L. Overton. Optimizing matrix stability. *Proceedings of the American Mathematical Society*, 129(6):1635–1642, 2000.

[19] James V. Burke, Adrian S. Lewis, and Michael L. Overton. Optimal Stability and Eigenvalue Multiplicity. *Foundations of Computational Mathematics*, 1(2):205–225, June 2001.

[20] James V. Burke, Adrian S. Lewis, and Michael L. Overton. Two numerical methods for optimizing matrix stability. *Linear Algebra and its Applications*, 351-352:117–145, August 2002.

[21] James V. Burke, Adrian S. Lewis, and Michael L. Overton. Variational analysis of functions of the roots of polynomials. *Mathematical Programming*, 104(2-3):263–292, July 2005.

[22] James V. Burke and Michael L. Overton. Stable perturbations of nonsymmetric matrices. *Linear Algebra and its Applications*, 171(July):249–273, July 1992.

[23] James V. Burke and Michael L. Overton. Differential properties of the spectral abscissa and the spectral radius for analytic matrix-valued mappings. *Nonlinear analysis*, 23(4):467–488, 1994.

[24] James V. Burke and Michael L. Overton. Variational analysis of non-Lipschitz spectral functions. *Mathematical Programming*, 351(January):317–351, June 2001.

[25] James V. Burke and Michael L. Overton. Variational Analysis of the Abscissa Mapping for Polynomials. *SIAM Journal on Control and Optimization*, 39(6):1651–1676, January 2001.

[26] C. Sidney Burrus, James W. Fox, Gary A. Sitton, and Sven Treitel. Horner's method for evaluating and deflating polynomials. Technical Report 1, Rice University, 2003.

[27] Mahmoud Chilali and Pascal Gahinet. H_∞ design with pole placement constraints: an LMI approach. *IEEE Transactions on Automatic Control*, 41(3):358–367, March 1996.

[28] Mahmoud Chilali, Pascal Gahinet, and Pierre Apkarian. Robust pole placement in LMI regions. *IEEE Transactions on Automatic Control*, 44(12):2257–2270, 1999.

[29] Chris Christensen. Newton's method for resolving affected equations. *The College Mathematics Journal*, 27:330–340, 1996.

[30] Carlos P. Coelho, Joel Phillips, and L. Miguel Silveira. A Convex Programming Approach for Generating Guaranteed Passive Approximations to Tabulated Frequency-Data. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(02):293–301, February 2004.

[31] Philippe Delsarte, Yves Genin, Yves Kamp, and Paul Van Dooren. Speech modeling and the trigonometric moment problem. *Philips Journal of Research*, 37:277–292, 1982.

[32] Tom D'haene. An Iterative Method to Stabilize a Transfer Function in the s- and z-Domains. *Instrumentation and Measurement, IEEE Transactions on*, 55(4):1192–1196, 2006.

[33] Tom D'haene. *Algorithms for identifying guaranteed stable and passive models from noisy data*. PhD thesis, Vrij Universiteit Brussels, 2008.

[34] Catherine Fraikin, Yurii Nesterov, and Paul Van Dooren. A gradient-type algorithm optimizing the coupling between matrices. *Linear Algebra and its Applications*, 429(5-6):1229–1242, September 2008.

[35] Pascal Gahinet and Arkadi Nemirovski. The Projective Method for solving linear matrix inequalities. *Mathematical Programming*, 77:163–190, 1997.

[36] Luca Gemignani. Computing a Hurwitz factorization of a polynomial. *Journal of Computational and Applied Mathematics*, 126(1-2):369–380, December 2000.

[37] Jose C. Geromel, Cid C. de Souza, and Robert E. Skelton. Static output feedback controllers: stability and convexity. *IEEE Transactions on Automatic Control*, 43(1):120–125, 1998.

[38] Andrea Griewank and Andrea Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, 2008.

[39] Andreas Griewank. On automatic differentiation. Technical Report November, Argonne National Laboratory, 1989.

[40] Andreas Griewank. A mathematical view of automatic differentiation. *Acta Numerica*, 2003.

[41] Yvan Hachez. *Convex Optimization over Non-Negative Polynomials : Structured Algorithms and Applications*. PhD thesis, Université catholique de Louvain, 2003.

[42] Sven Hammarling. Numerical solution of the discrete-time, convergent, non-negative definite Lyapunov equation. *Systems & Control Letters*, 17(2):137–139, August 1991.

[43] Laurent Hascoët and Valérie Pascual. *TAPENADE for C*. 2004.

[44] Georg Heinig and Uwe Jungnickel. Lyapunov equations for companion matrices. *Linear Algebra and its Applications*, 76:137–147, April 1986.

[45] Peter Henrici. Fast Fourier Methods in Computational Complex Analysis. *SIAM Review*, 21(4):481–527, October 1979.

[46] Nicholas J. Higham. *Matrix nearness problems and applications*. Applications of Matrix Theory. Oxford University Press, 1989.

[47] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 1996.

[48] Diederich Hinrichsen and Anthony J. Pritchard. *Mathematical Systems Theory I: Modelling, State Space Analysis, Stability and Robustness*. Springer, 2005.

[49] A. J. Hoffman and H. W. Wielandt. The variation of the spectrum of a normal matrix. *Duke Mathematical Journal*, 20(1):37–39, March 1953.

[50] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[51] Florian Jarre. Optimal ellipsoidal approximations around the analytic center. *Applied mathematics & optimization*, 19:15–19, 1994.

[52] Florian Jarre. An interior method for nonconvex semidefinite programs. *Optimization and Engineering*, pages 347–372, 2000.

[53] Stoyan Kanev, Carsten Scherer, Michel Verhaegen, and Bart De Schutter. Robust output-feedback controller design via local BMI optimization. *Automatica*, 40(7):1115–1127, July 2004.

[54] Tosio Kato. *Perturbation Theory for Linear Operators*. Springer-Verlag, 1980.

[55] Peter Kirrinnis. Partial fraction decomposition in $\mathbb{C}(F)$ and simultaneous newton iteration for factorization in $\mathbb{C}[z]$. *Journal of Complexity*, 444:378–444, 1998.

[56] Daniel Kressner, Emre Mengi, Ivica Nakic, and Ninoslav Truhar. Generalized Eigenvalue Problems with Specified Eigenvalues. *IMA Journal of Numerical Analysis*, (submitted), September 2010.

[57] Seth L. Lacy and Dennis S. Bernstein. Subspace Identification With Guaranteed Stability Using Constrained Optimization. *IEEE Trans. Autom. Control*, 48(07):1259–1263, 2003.

[58] Peter Lancaster. Explicit solutions of linear matrix equations. *Siam Review*, 12(4):544–566, 1970.

[59] Heinz Langer and Branko Najman. Remarks on the perturbation of analytic matrix functions II. *Integral Equations and Operator Theory*, 12(3):392–407, May 1989.

[60] Heinz Langer and Branko Najman. Remarks on the perturbation of analytic matrix functions III. *Integral Equations and Operator Theory*, 15(5):796–806, September 1992.

[61] Heinz Langer and Branko Najman. Leading coefficients of the eigenvalues of perturbed analytic matrix functions. *Integral Equations and Operator Theory*, 16(4):600–604, December 1993.

[62] L. V. Levantovskii. Singularities of the boundary of the stability domain. *Functional Analysis and Its Applications*, 16(1):34–37, 1982.

[63] Adrian S. Lewis and Michael L. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, (to appear), February 2012.

[64] V. B. Lidskii. Perturbation theory of non-conjugate operators. *USSR Computational Mathematics and Mathematical Physics*, 1:73–85, 1965.

[65] A.M. Lyapunov. *Probleme general de la stabilite du mouvement*. Princeton University Press, Princeton, 1947.

[66] Alexei A. Mailybaev. On the stability of polynomials depending on parameters. *Journal of Computer and Systems Sciences International*, 39(2):165–172, 2000.

[67] Alexei A. Mailybaev and Alexander P. Seiranyan. Singularities of the boundaries of stability domains. *Journal of applied mathematics and mechanics*, 62(6):984–995, 1998.

[68] Alexei A. Mailybaev and Alexander P. Seyranian. On Singularities of a Boundary of the Stability Domain. *SIAM Journal on Matrix Analysis and Applications*, 21(1):106–128, January 1999.

[69] J.D. Markel and A.H Gray. *Linear Prediction of Speech*. Springer-Verlag, Berlin, 1976.

[70] Julio Moro, James V. Burke, and Michael L. Overton. On the Lidskii–Vishik–Lyusternik Perturbation Theory for Eigenvalues of Matrices with Arbitrary Jordan Structure. *SIAM Journal on Matrix Analysis and Applications*, 18(4):793–817, October 1997.

[71] Randolph L. Moses and Duixian Liu. Determining the closest stable polynomial to an unstable one. *IEEE Transactions on Signal Processing*, 39(04):901–906, April 1991.

[72] Richard D. Neidinger. Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming. *SIAM Review*, 52(3):545–563, January 2010.

[73] Yurii Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, 2004.

[74] Yurii Nesterov. Gradient methods for minimizing composite objective function. *Core Discussion Paper*, 76(May), 2007.

[75] Ulö Nurges. New stability conditions via reflection coefficients of polynomials. *IEEE Transactions on Automatic Control*, 50(9):1354–1360, September 2005.

[76] Francois-Xavier Orbandexivry, Yurii Nesterov, and Paul Van Dooren. Gradient Method for the Nearest Stable Matrix in Continuous-time. *Optimization Methods and Software*, (Submitted), 2013.

[77] Francois-Xavier Orbandexivry, Yurii Nesterov, and Paul Van Dooren. Nearest stable system using successive convex approximations. *Automatica*, 49(5):1195–1203, May 2013.

[78] Michael L. Overton. Private communication, 2012.

[79] Michael L. Overton and Robert S. Womersley. On Minimizing the Spectral Radius of a Nonsymmetric Matrix Function: Optimality Conditions and Duality Theory. *SIAM Journal on Matrix Analysis and Applications*, 9(4):473–498, October 1988.

[80] Andy Packard and John Doyle. The Complex Structured Singular Value. *Automatica*, 29(1):71–109, 1993.

[81] Victor Y. Pan. Univariate Polynomials: Nearly Optimal Algorithms for Numerical Factorization and Root-finding. *Journal of Symbolic Computation*, 33(5):701–733, May 2002.

[82] Boris T. Polyak and Pavel S. Shcherbakov. Hard Problems in Linear Control Theory: Possible Approaches to Solution. *Automation and Remote Control*, 66(5):681–718, May 2005.

[83] Victor Puiseux. Recheche sur les fonctions algébriques. *Journal de mathématiques pures et appliquées*, 15:365–480, 1850.

[84] Victor Puiseux. Nouvelles recherches sur les fonctions algébriques. *Journal de mathématiques pures et appliquées*, 16:228–240, 1851.

[85] Mustapha Ait Rami and Didier Henrion. A hierarchy of LMI inner approximations of the set of stable polynomials. *Automatica*, 47:1455–1460, 2011.

[86] R. Tyrrell Rockafellar and Roger J-B. Wets. *Variational analysis*. Springer, 1998.

[87] Lahcen Saydy, André L. Tits, and Eyad H. Abed. Guardian maps and the generalized stability of parametrized families of matrices and polynomials. *Mathematics of Control, Signals, and Systems*, 3(4):345–371, December 1990.

[88] Carsten Scherer, Pascal Gahinet, and Mahmoud Chilali. Multiobjective output-feedback control via LMI optimization. *IEEE Transactions on Automatic Control*, 42(7):896–911, July 1997.

[89] Alexander P. Seyranian and Alexei A. Mailybaev. Interaction of eigenvalues in multi-parameter problems. *Journal of Sound and Vibration*, 267(5):1047–1064, November 2003.

[90] Gary A. Sitton, C. Sidney Burrus, James W. Fox, and Sven Treitel. Factoring very-high-degree polynomials. *IEEE Signal Processing Magazine*, 20(6):27–42, November 2003.

[91] Laurent Sorber, Marc Van Barel, and Lieven De Lathauwer. Unconstrained Optimization of Real Functions in Complex Variables. *SIAM Journal on Optimization*, 22(3):879–898, July 2012.

[92] J. Sreedhar, Paul Van Dooren, and André L. Tits. A fast algorithm to compute the real structured stability radius. In *International Series of Numerical Analysis*. 1996.

[93] Petre Stoica and Randolph L. Moses. On the unit circle problem: the Schur-Cohn procedure revisited. *Signal processing*, 26:95–118, 1992.

[94] Ji-guang Sun. On the variation of the spectrum of a normal matrix. *Linear Algebra and its Applications*, 246(1996):215–223, October 1996.

[95] Marc Van Barel. Private communication, 2013.

[96] Marc Van Barel, Raf Vandebril, Paul Van Dooren, and Katrijn Frederix. Implicit double shift QR-algorithm for companion matrices. *Numerische Mathematik*, 116(2):177–212, April 2010.

[97] Joris Vanbiervliet, Bart Vandereycken, Wim Michiels, Stefan Vandewalle, and Moritz Diehl. The smoothed spectral abscissa for robust stability optimization. *SIAM Journal on Optimization*, 20(1):156, 2009.

[98] M. I. Vishik and L. A. Lyusternik. The solution of some perturbation problems for matrices and selfadjoint or non-selfadjoint differential equations I. *Russian Mathematical Surveys*, 15(3):1–73, June 1960.

[99] Yinyu Ye. On affine scaling algorithms for nonconvex quadratic programming. *Mathematical Programming*, 56(1-3):285–300, August 1992.