Gran Sasso Science Institute
MATHEMATICS OF NATURAL, SOCIAL AND LIFE SCIENCES
DOCTORAL PROGRAMME
Cycle: XXXVI - AY 2020/2024

Low-rank properties in structured matrix nearness problems

PhD candidate:

Stefano Sicilia

Supervisor:

Prof. Dr. Nicola Guglielmi
Gran Sasso Science Institute

Thesis submitted for the degree of Doctor of Philosophy

Thesis Jury Members
Prof. Dr. Daniele Boffi (King Abdullah University of Science and Technology)
Prof. Dr. Dajana Conte (Università di Salerno)
Prof. Dr. Raffaele D'Ambrosio (Università dell'Aquila)
Prof. Dr. Volker Mehrmann (Technische Universität Berlin)
Prof. Dr. Vanni Noferini (Aalto University)

Thesis Referees
Prof. Dr. Volker Mehrmann (Technische Universität Berlin)
Prof. Dr. Vanni Noferini (Aalto University)

Abstract

In numerical linear algebra, the problem of computing the distance of a given matrix A from a given set $\mathcal{P}$ arises in different fields of matrix and control theory, where it is used to characterize the robustness of considered systems. Some examples include, but are not limited to, distance to singularity, matrix stability, measures in control theory, etc. (see e.g. [28, 33, 44, 45, 46, 53]). The problem consists in computing an element $B \in \mathcal{P}$ such that the distance between B and A is the smallest possible; under suitable assumptions on $\mathcal{P}$, the problem is always well defined. The most common version in this matrix nearness context concerns the *unstructured distance*, which means that the optimization problem introduced for the computation of the matrix B does not take into account any specific structure of the original matrix A, e.g. its sparsity pattern, its reality, a particular design of the entries etc.. Quite recently, an increasing interest has risen for the structured version of the distance (see e.g. [21, 31, 34, 37, 38, 49, 65]), where the optimizer sought is required to preserve a specific structure that the matrix A has. In this case we talk about *structured distance* between A and the set $\mathcal{P}$, which is clearly larger or equal than the *unstructured distance* between the same objects and it could have a different order of magnitude. Also in this case the problem may not be well-defined, for instance if the constraint forced by the structure is too strong and it makes impossible to find a matrix $B \in \mathcal{P}$ with the structure required, but again under reasonable assumptions it is possible to consider that the problem is solvable. The main motivation behind the introduction of the *structured distance* is that it allows to take into account some features of the original matrix A and it can be exploited to get a more appropriate matrix B that provides a more meaningful solution to the matrix nearness problem.

In this PhD thesis we study some examples of matrix nearness problems and we focus particularly on their structured version. We show a versatile two-level approach that can be used in this context and that can be adapted to many applications. In particular we discuss in detail the structured distance to stability of a Hurwitz or Schur matrix, the structured distance to singularity of an invertible matrix, the robustness of the spectral clustering of a graph and the structured stabilization of a matrix. All the unstructured versions of these problems possess an intrinsic low-rank feature, which is evident in their solution, but this remarkable fact seems less obvious for the structured case. We show how to uncover the low-rank property of the problem also in the structured case and we describe how it is possible to exploit this fact to get an efficient algorithm that computes the distance sought and the associated extremizer(s).

Acknowledgements

This thesis is the result of the four years spent in L'Aquila at GSSI for my PhD. During this period I had the chance to meet many people and this work would not have been possible without them. Thus I wish to acknowledge them all and to dedicate a few words to thank their support.

First and foremost I am very grateful to my supervisor, Prof. Nicola Guglielmi, for his guidance, support and encouragement during my PhD. His lectures on the first year attracted me and made me appassionate to the very interesting subject that then has become the topic of my PhD thesis. I really appreciated doing research with him and his advices have been very useful to me to understand many crucial facts in numerical analysis. I also thank him, together with Prof. Francesco Tudisco, for their financial support, which allowed me to attend many conferences also abroad.

I want to thank all the members of the jury for accepting to attend my PhD defence. I am very grateful to the two referees that reviewed my PhD thesis, Prof. Volker Mehrmann and Prof. Vanni Noferini. I thank them for the time spent to read the draft of the thesis and I appreciated their constructive comments and suggestions which allowed to improve a previous version of the thesis.

I really wish to thank Prof. Nicolas Gillis for the opportunity to visit him in Mons and for his financial support. The two months spent there gave me the possibility to share ideas with him and with his research group, in order to work on the topic of my thesis also from another perspective.

I would like to thank Prof. Dario Bini for all his advices during my Master's degree in Pisa and in particular I am grateful for suggesting me to follow my studies after the degree. Thanks to that, I decided to start the PhD at GSSI and I believe that this has been an excellent choice for my career.

During my stay at GSSI I had the chance to be part of the numerical analysis group and to attend many seminars organized in this environment. I want to thank all of its members for arranging very intersting workshops with many guests, so that I had the chance to learn more about their research field and to share ideas with them.

Now I would like to say a few words of gratitude also for the people I met during these years and with whom I shared some moments together also outside of the mathematical work.

During the conferences I had the chance to meet many other PhD students and young researchers and I wish to thank all of them for the nice moments spent together. In particular, I want to thank the numerical analysis group of Pisa that I frequently met in many workshops and schools and with which I kept in touch with these events.

I am really grateful to the research group I met in Mons. During my visit in Belgium they made me feel one of them and it has been a pleasure to travel together around the country and to try many belgian food specialties!

I would like to thank all the GSSI people, since they made me really feel part of a community. Especially I am grateful to all the colleagues with whom I shared the office.

Small breaks, chats and snacks gave a great boost to my working skills! Moreover I really want to thank Arturo and Martino for all the fantastic moments spent together, from the walks around the mountains in L'Aquila to the Italian Formula 1 Grand Prix.

I also want to thank all my friends from Pisa with whom I spent amazing moments during my Bachelor's and Master's degrees. I want to acknowledge Lorenzo and Marilena for all the amazing moments during the lessons' breaks at the university and for all the funny video calls of these years where we relive them. Moreover I wish to thank Alberto, Giacomo, Mattia P. and Mattia F. for all the trips we arranged in the last four years and for all the nice video calls that make me feel like if we were all still in the same place; they also deserve a special mention, for inviting and hosting me in Pisa and Trieste multiple times.

Finally, I would like to express my gratitude to my family and in particular to my parents. Their support in all these years constituted a solid base to build up my future and to let me continue my studies. I am really thankful that they always believe in me and that they encouraged me to do the same in all situations.

Contents

Chapter 1

Introduction

A matrix nearness problem is an optimization problem whose aim is to estimate how far a given matrix A is to fulfil a certain property. More precisely, given a matrix $A \in \mathbb{C}^{m\times n}$ and a property $\mathscr{P}$, the goal of the problem is to approximate the distance between A and the set

$$\mathcal{P} = \{M \in \mathbb{C}^{m\times n} : M \text{ fulfils the property } \mathscr{P}\}$$

and to compute the corresponding element $B \in \mathcal{P}$ that realizes it. This kind of problem arises in a wide range of topics, such as in graph theory (see [3, 37]), in dynamical systems (see [20]), in control theory (see e.g. [53]), in machine learning (see e.g. [25]) and in any field where it could be useful to consider the robustness of a certain tool, element or feature. In this PhD thesis we will focus on the case in which the property $\mathscr{P}$ is related to the spectrum of A (or to that of a matrix explicitly related to it, such as the Laplacian of A) and hence we always assume $n = m$; this setting leads to an eigenvalue optimization problem.

1.1 Matrix nearness problems

The most common matrix nearness problems may be divided into two classes: the *violating* type and the *recovering* type. The first group concerns robustness of a nice property of a matrix, while the second one is interested in the closest matrix to the given one that fulfils a desired property that the original matrix does not possess.

In some applications it is important that A does not have a certain property $\mathscr{P}$, and it useful to know how close A is to have this undesirable property. If the distance is small, then the source problem is likely to be ill-conditioned for A, and remedial action may need to be taken. We refer to this kind of applications as *violating* problems, since they concern the break of a desirable property. Some examples are the computation of the distance to singularity or to instability (see e.g. [34]), but also the computation of the ε-pseudospectrum of a matrix, i.e. the set of all eigenvalues of all the perturbations of A in the form $A + \Delta$ with $\|\Delta\| \leq \varepsilon$, belongs to this class (see e.g. [36, 65, 66]). A relevant instance of *violating* problem is the computation of the distance to instability of a Hurwitz matrix, that is a matrix with all its eigenvalues with negative real part. If we define the pseudoabscissa of the matrix $A \in \mathbb{C}^{n\times n}$ as

$$\alpha(A) := \max\{\mathrm{Re}(\lambda) : \lambda \text{ is an eigenvalue of } A\},$$

this optimization problem can be written as

$$\arg\min_{\Delta\in\mathbb{C}^{n\times n}}\{\|\Delta\| : \alpha(A + \Delta) = 0\}, \tag{1.1}$$

where $\|\cdot\|$ is a matrix norm and Δ is a matrix perturbation. In this case the property we want to violate is $\mathscr{P} = \{A \text{ is a Hurwitz-stable matrix}\}$, which only concerns the eigenvalue with largest real part. The aim of this problem is to verify whether the stability property of the given matrix A is robust or not. If the value of the distance, i.e. the minimum of the optimization problem, is not large, then it is possible to find a matrix $\Delta_\star$ with small norm such that $A + \Delta_\star$ is not Hurwitz-stable anymore. This means that the computations performed with the matrix A, that is theoretically stable, may not be reliable, since the matrix is close to be unstable. This issue is relevant in many applications, such as when the entries of the matrix A are given with an uncertainty error, which usually occurs while dealing with physical measurements: in this case, even though the unperturbed matrix A is guaranteed to be Hurwitz, there could be some small changes in its entries that destroy the stability of A, making it not suitable for performing reliable computations. If the Hurwitz matrix A is sparse, one may also be interested in the matrix nearness problem that concerns its structured stability robustness, that is

$$\arg\min_{\Delta \in \mathcal{S}} \{\|\Delta\| : \alpha(A + \Delta) = 0\}, \tag{1.2}$$

where $\mathcal{S} \subseteq \mathbb{C}^{n\times n}$ denotes the set of the matrices with the same sparsity pattern of A. This measure provides a more meaningful guarantee, since the perturbation Δ is only allowed to affect the non-zero entries of A, thus leaving its pattern unchanged.

The second class of matrix nearness problems refers to the situation where a complex square matrix $A \in \mathbb{C}^{n\times n}$ approximates a matrix B, and B is known to possess a certain property $\mathscr{P}$. Because of errors occurred during the computation and the approximation, the determined matrix A may not fulfil property $\mathscr{P}$. The most direct way to overcome this issue is to replace the matrix A with the nearest matrix B which instead satisfies the property $\mathscr{P}$. This kind of problems act as a *recovering* of the property $\mathscr{P}$ and thus we refer to them with this qualification. Nearness problems arising in this context involve, for example, the stabilization of a matrix (see [31, 38]), which is one of the most relevant instances of *recovering* problem. This is somehow the opposite of the previous discussed *violating* example and it concerns a non-Hurwitz matrix A such as $\alpha(A) > 0$ and the purpose is to move all its eigenvalues to the complex left half-plane. In other words we aim to solve the optimization problem

$$\arg\min_{\Delta \in \mathbb{C}^{n\times n}} \{\|\Delta\| : \alpha(A + \Delta) \leq 0\},$$

which involves all the eigenvalues of A with positive real part. An application of this problem concerns the discretization of a PDE with some known stability properties that are broken by a numerical scheme, because it computes a non-Hurwitz matrix A. In order to recover the stability property, we wish to make A Hurwitz by means of the smallest possible perturbation. Again, if the matrix A has a certain structure, we may further ask that also the optimal perturbation $\Delta_\star$ preserves this structure, so that the stabilized matrix $A + \Delta_\star$ has the same structure of the original matrix A. In this case, the result provided is more appropriate and it solves the structured matrix nearness problem

$$\arg\min_{\Delta \in \mathcal{S}} \{\|\Delta\| : \alpha(A + \Delta) \leq 0\},$$

which also involves all the eigenvalues of A with positive real part.

Both *recovering* and *violating* problems can be formalized in the same way as follows: given a square complex matrix $A \in \mathbb{C}^{n\times n}$ that fulfils a certain spectral property

$\mathscr{P}$, we consider the matrix nearness problem defined as

$$\mathcal{A} = \arg\min_{\Delta \in \mathbb{C}^{n\times n}} \{\|\Delta\| : A + \Delta \text{ does not fulfil the property } \mathscr{P}\}, \tag{1.3}$$

where $\|\cdot\|$ is a matrix norm and Δ is a matrix perturbation. A solution $\Delta_\star \in \mathcal{A}$ is a perturbation that added to the original matrix A implies that $A + \Delta_\star$ loses its property $\mathscr{P}$. Problem (1.3) is written in a *violating* form, but by considering the property $\mathscr{Q} = \neg\mathscr{P}$ it is also possible to represent a generic *recovering* problem.

Remark 1.1.1. *The use of the terms 'recovering' and 'violating' problems is not mathematically rigorous, but simply intuitive. Indeed the fact that a property $\mathscr{P}$ is desirable or not depends on a subjective opinion and it is also influenced by the application considered. Moreover both problems share the same formalization in* (1.3)*, making them hard to distinguish from a mathematical point of view.*

The purpose of this subdivision is rather to give an idea of the qualitative differences among matrix nearness problems involving the spectrum of a matrix. In particular the word 'recovering' is associated with a problem concerning the distance to stability, which in general involves all the eigenvalues of a matrix. In contrast, the word 'violating' refers to a distance to instability that generally concerns a single or a few eigenvalues of a matrix. According to this intuition, 'recovering' problems are generally harder to face than 'violating' problems, since it is easier to deal with few eigenvalues rather than with all the spectrum of the matrix.

As already mentioned, in many cases the matrix A has a certain structure, such as a prescribed sparsity pattern, a Toeplitz structure, it has real entries, etc. and one may ask that this feature is considered as a constraint in the matrix nearness problem, in order to maintain the structure also for the perturbed matrix. Formally, assume that A satisfies the property $\mathscr{P}$ and that it belongs to a certain subspace $\mathcal{S} \subseteq \mathbb{C}^{n\times n}$; the associated structured matrix nearness problem is

$$\mathcal{A}_{\mathcal{S}} = \arg\min_{\Delta \in \mathcal{S}} \{\|\Delta\| : A + \Delta \text{ does not fulfil the property } \mathscr{P}\}, \tag{1.4}$$

where the perturbation Δ must preserve the structure $\mathcal{S}$. The structured problems make sense even in the case that $A \notin \mathcal{S}$, but in many applications it is more relevant to assume that also A possesses the structure. In some contexts it is more appropriate to consider a structured matrix nearness problem like (1.4), rather than an unstructured one as (1.3). As already mentioned in the examples discussed, an important motivation behind the introduction of the structured problem (1.4) concerns the approximation of a sparse matrix A, that it is known by theory to fulfil a certain property, but because of errors and approximations occurred during the computation, the property is lost. By considering the subspace $\mathcal{S}$ as the description of the sparsity pattern of A, we can ensure that the perturbed matrix $\Delta_\star$ computed as a solution of the matrix nearness problem preserves the structure, so that the zero entries are not changed by the perturbation $\Delta_\star$.

Some unstructured matrix nearness problems have explicit solution that do not need the implementation of an involved algorithm to be computed. For instance, in the Frobenius norm metric, in order to find the closest real matrix to a given one it is sufficient to take the real part of its entries, while the closest symmetric matrix to a given one is its symmetric part. However, when we consider the structured version (1.4), it is generally unlikely to find an analytic expression for the solution, and even in the easiest examples there is not a known explicit formula. This generally makes non-trivial the computation of a solution of (1.4) and hence it is needed to approximate it through a numerical method.

In this PhD thesis we aim to generalize a two-level approach recently developed by Guglielmi and Lubich (see e.g. [31]) that has been originally designed for unstructured problems. We show how to adapt it to the structured case without compromising its remarkable properties, such as its low-rank features and the gradient system that it uses to compute the solution.

1.2 Outline of the thesis

The thesis is divided into three main parts:

1. Chapter 1 introduces the topic, while Chapter 2 and Chapter 3 describe the underlying technique used by the two-level approach, both for the unstructured and structured cases. We present some general theoretical results that can be used in all the three proposed applications.

2. Chapter 4, 5 and 6 are dedicated to the three applications mentioned in Section 1.5. They represent the steps of the development of the method from the chronological point of view: rank-1, fixed rank r and adaptive rank.

3. Chapter 7 draws the conclusions of the thesis, while the appendix concerns some basic but useful results and some examples.

More in detail, the outline of the thesis is the following:

- Chapter 1 contains the introduction to the topic of the thesis and it presents the method and the applications that will be discussed in the other chapters.

- Chapter 2 introduces the two-level approach for solving problem (1.3) and it discusses its low-rank features. It shows how to obtain a full-rank gradient system whose stationary points coincide with the solution of the optimization problem. Then it derives a low-rank ODE which shares many features with the full-rank gradient system.

- Chapter 3 generalizes the two-level approach presented in Chapter 2 to the structured case. It shows how to derive the structured gradient system as for the unstructured case and how to prove its properties. Then it introduces a low-rank ODE also in this setting and it describes all its features. At the end of the chapter, we state and prove the main theorem about the local convergence of the integration of the low-rank equation towards its stationary points, that are also associated to the solution of the matrix nearness problem.

- Chapter 4 presents a technique to compute the distance to instability of a matrix or the distance to singularity, by solving a *violating* matrix nearness problem through the integration of a rank-1 system.

- Chapter 5 concerns the robustness of the spectral clustering algorithm by means of a *violating* matrix nearness problem of distance to ambiguity that involves the integration of a rank-4 symmetric differential equation.

- In Chapter 6 we present a *recovering* matrix nearness problem for computing the closest stable matrix to a given one, also considering its structure, by means of a rank-adaptive system solved by an integrator well designed for this purpose.

- In Chapter 7 we draw the conclusions of the thesis and we discuss some future research directions.
- Appendix A contains some basic results about the fixed rank manifold.
- Appendix B provides some basic properties of projections and some examples related to this thesis.
- Appendix C shows two non-generic counterexamples associated to the applications studied.
- Appendix D contains some miscellaneous results.

1.3 A two-level approach

The method proposed in this thesis for solving the matrix nearness problems (1.3) and (1.4) relies on a two-level approach that splits the original problem into two nested sub-problems that are solved by an *inner iteration* and by an *outer iteration* (see e.g. [26, 27, 35]). In our setting and from now on we consider the metric induced by the Frobenius norm and the Frobenius inner product (see Section 1.6 for the formal definition).

Let us consider a subspace $\mathcal{S} \subseteq \mathbb{C}^{n\times n}$ (in the unstructured case $\mathcal{S} = \mathbb{C}^{n\times n}$), a real number $a_\star \in \mathbb{R}$ and a functional $\mathscr{F} : \mathcal{S} \to \mathbb{R}$ such that $\mathscr{F}(0) > a_\star$ and, for all $\Delta \in \mathbb{C}^{n\times n}$,

$$\mathscr{F}(\Delta) \le a_\star \iff A + \Delta \text{ does not fulfil the property } \mathscr{P},$$

which means that the functional $\mathscr{F}$ takes values non-greater than $a_\star$ if and only if Δ is admissible. In many applications $a_\star = 0$ or $a_\star = -1$, and the condition $\mathscr{F}(0) > a_\star$ simply means that the zero matrix is not an admissible perturbation and hence A does not fulfil the property $\mathscr{P}$. We rewrite the perturbation $\Delta = \varepsilon E$, where $\varepsilon > 0$ is the perturbation size and E has unit Frobenius norm and we define the functional F_ε as

$$F_\varepsilon(E) := \mathscr{F}(\varepsilon E).$$

The *inner iteration* minimizes the functional F_ε when the perturbation size ε is fixed, while the *outer iteration* aims in finding the smallest value $\varepsilon_\star$ such that it is possible to have $\mathscr{F}(\Delta) = a_\star$ for some perturbation Δ with Frobenius norm $\varepsilon_\star$. The outline of the two-level method is the following:

- *Inner iteration*: For a fixed ε, compute a matrix perturbation $E_\star(\varepsilon)$ such that

$$E_\star(\varepsilon) \in \arg\min_{\|E\|_F=1} F_\varepsilon(E), \tag{1.5}$$

  which for the structured case becomes

$$E_\star(\varepsilon) \in \arg\min_{\|E\|_F=1, E\in\mathcal{S}} F_\varepsilon(E) = \arg\min_{\|\Delta\|_F=\varepsilon,\ \Delta\in\mathcal{S}} \mathscr{F}(\Delta), \tag{1.6}$$

- *Outer Iteration*: Find the smallest value $\varepsilon_\star > 0$ such that

$$\varphi(\varepsilon) := F_\varepsilon(E_\star(\varepsilon)) = a_\star.$$

In almost all the applications we will consider, we have $a_\star = 0$, so we assume this fact unless something else is stated.

The *inner iteration* is the most elaborated procedure, while the *outer iteration* is theoretically easier to solve, since it consists of a one-dimensional root-finding problem quite simple to describe, even though it could be challenging. For instance, for the *violating* problem (1.1) concerning the computation of the distance to Hurwitz instability, the functional takes the form

$$\mathscr{F}(\Delta) = -\operatorname{Re}(\lambda_{\text{target}}(A+\Delta)),$$

where λ_{target} is the eigenvalue with largest real part, while for the *recovering* stabilization problem (1.2) for a non-Hurwitz matrix we have

$$\mathscr{F}(\Delta) = \frac{1}{2}\sum_{i=1}^{n} \left(\left(\operatorname{Re}\left(\lambda_i(A+\Delta)\right)\right)_+ \right)^2$$

where $\lambda_i(A+\Delta)$ denotes the i-th eigenvalue of $A+\Delta$ and $a_+ = \max(a,0)$.

To solve problem (1.5), the *inner iteration* introduces a perturbation matrix path $E(t)$ with $t \geq 0$ and it integrates the following matrix ODE

$$\dot{E} = -G_\varepsilon(E) + \operatorname{Re}\langle G_\varepsilon(E), E\rangle E, \tag{1.7}$$

where $G_\varepsilon(E)$ is the gradient of F_ε. The expression of $G_\varepsilon(E)$ is generally available and for instance, for the computation of the distance to Hurwitz instability, we have

$$G_\varepsilon(E) = -xy^*,$$

where x and y are, respectively, the unit left and right eigenvectors associated with the target eigenvalue $\lambda_{\text{target}}(A+\varepsilon E)$ so that $x^*y > 0$ (see Lemma 4.2.3). It is possible to prove that the stationary points of (1.7) corresponds to the local minima of F_ε and, in order to find them, we integrate the ODE (1.7) until we reach a sought stationary point. Since equation (1.7) is a gradient system, an integration of it will always lead to a stationary point and this is exactly what we are interested in for solving the optimization problem (1.5). In the structured case, the ODE integrated by the *structured inner iteration* is similar, but in its expressions it shows up the orthogonal projection with respect to the Frobenius inner product onto $\mathcal{S}$, denoted by $\Pi_{\mathcal{S}}$:

$$\dot{E} = -\Pi_{\mathcal{S}} G_\varepsilon(E) + \operatorname{Re}\langle \Pi_{\mathcal{S}} G_\varepsilon(E), E\rangle E. \tag{1.8}$$

Also equation (1.8) is a gradient system and hence its integration always leads to one of its stationary points, that again correspond to the local minimizers of (1.6) (see Theorems 3.1.3 and 3.1.4). Both in the unstructured and structured cases, it is possible to show that, up to non-generic events, the stationary points of the ODEs are of the form $E \propto G_\varepsilon(E)$ for equation (1.7) and $E \propto \Pi_{\mathcal{S}} G_\varepsilon(E)$ for equation (1.8).

Since in many applications the matrix $G_\varepsilon(E)$ has low-rank, say $\operatorname{rank}(G_\varepsilon(E)) = r \ll n$, it follows that in the unstructured case the stationary points have themselves low-rank. This motivates to introduce a different ODE whose trajectory belongs to the rank-r manifold $\mathcal{M}_r$:

$$\dot{E} = P_E\left(-G_\varepsilon(E) + \operatorname{Re}\langle G_\varepsilon(E), E\rangle E\right), \tag{1.9}$$

where P_E denotes the orthogonal projection with respect to the Frobenius inner product onto the tangent space $\mathcal{T}_E\mathcal{M}_r$ of $\mathcal{M}_r$ in E. Again ODE (1.9) is a gradient system and it is possible to show that equations (1.7) and (1.9) share the same stationary points, although the trajectories of their solutions do not coincide. Thus, since we focus our interest just on the stationary points and not on the whole trajectory of the solution of the ODE, we can integrate this new low-rank ODE instead of the original full-rank. This makes it possible to exploit the remarkable low-rank property that is naturally found in the unstructured problem, allowing us to get benefits in the numerical computations.

In contrast, for the structured problem (1.4), it is less obvious to prove this fact, since the projection $\Pi_{\mathcal{S}}$ generally destroys the low-rank property of the optimizers of the structured ODE (1.8). Thus it would be very appealing to recover the low-rank features of the unstructured matrix nearness problem also in the structured case.

1.4 The new structured-low-rank ODE

In this section we highlight the main theoretical novelties of this thesis. We observe that solutions of (1.8) can be rewritten as $E = \Pi_{\mathcal{S}} Z$, where Z solves the ordinary differential equation

$$\dot{Z} = -G_\varepsilon(\Pi_{\mathcal{S}} Z) + \mathrm{Re}\langle G_\varepsilon(\Pi_{\mathcal{S}} Z), \Pi_{\mathcal{S}} Z\rangle Z, \tag{1.10}$$

but this does not guarantee that $Z(t)$ is a low-rank matrix path. Thus we introduce a new perturbation that, for simplicity, we call again $E(t)$, although it does not coincide with the solution of (1.8), and we look for a low-rank matrix path $Y(t) \subseteq \mathcal{M}_r$ such that

$$E(t) = \Pi_{\mathcal{S}} Y(t), \qquad t \in [0, +\infty).$$

Taking inspiration from equation (1.10), we project the right hand side onto $\mathcal{T}_Y \mathcal{M}_r$ and we get

$$\dot{Y} = P_Y\left(-G_\varepsilon(\Pi_{\mathcal{S}} Y) + \mathrm{Re}\langle P_Y(G_\varepsilon(\Pi_{\mathcal{S}} Y)), \Pi_{\mathcal{S}} Y\rangle Y\right). \tag{1.11}$$

There exists an explicit one-to-one correspondence between the stationary points of (1.8) and those of (1.11), which ensures that we are not introducing nor losing solutions of problem (1.4) if we integrate the low-rank equation instead of the full-rank one. However in this case it occurs an issue that does not appear in the unstructured case: equation (1.11) is not a gradient system. This means that it is not guaranteed a priori that its integration leads to a stationary point, but, for instance, it may run into a periodic orbit or simply diverge. The latter fact is an important point, since it could potentially make integrating equation (1.11) useless.

Fortunately it is possible to show that, despite ODE (1.11) is not a gradient system, it is somehow close to that. In particular, by choosing a proper starting point sufficiently close to a stationary point, integrating equation (1.11) always leads to that stationary point. This just yields a local convergence result weaker than the global convergence property of a gradient system, but our numerical experiments show that this is enough to make the method work in practice. In this way, integrating (1.11) makes it possible to exploit the underlying low-rank features of the matrix nearness problem also in the structured case.

1.5 The applications studied

Chapters 4, 5 and 6 of this thesis concern three applications where the benefit of the low-rank insights can be exploited for a structured matrix nearness problem.

Structured distances to instability and singularity

Chapter 4 (see [34]) focuses on three problems that belong to the class of *violating* matrix nearness problems: the structured distance to instability (both in Hurwitz and Schur terms) and the structured distance to singularity.

- A Hurwitz-stable matrix is a matrix whose spectrum lies in the complex left half-plane. It is almost unstable if an eigenvalue is close to the imaginary axis.
- A Schur-stable matrix is a matrix whose spectrum lies in the complex unit disk. It is close to being unstable when an eigenvalue is not far from the border of the disk.
- An invertible matrix is close to being singular when one of its eigenvalues is close to the origin.

By exploiting the similarities between these problems, it is possible to solve them with the same approach that considers a single target eigenvalue λ_{target} for the definition of the functional $\mathscr{F}$ for the two-level approach: the one with largest real part (Hurwitz instability), the one with largest modulus (Schur instability) or the one with smallest modulus (distance to singularity). In all cases the expression of the functional to be minimized in the *structured inner iteration* for a fixed perturbation size ε is

$$F_\varepsilon(E) = f\left(\lambda_{\text{target}}\left(A+\varepsilon E\right), \overline{\lambda}_{\text{target}}\left(A+\varepsilon E\right)\right)$$

where $f(z,\overline{z}) = -\operatorname{Re}(z)$ for Hurwitz instability, $f(z,\overline{z}) = -|z|^2$ for Schur instability and $f(z,\overline{z}) = |z|^2$ for the distance to singularity. For Schur instability we have $a_\star = -1$, while $a_\star = 0$ in the other cases. These definitions of f imply that the gradient of the objective functional is a rank-1 matrix, which means that the trajectory of equation (1.11) belongs to the rank-1 manifold.

Spectral clustering robustness

Chapter 5 focuses on the problem of spectral clustering robustness. Given an undirected weighted graph, the spectral clustering is an algorithm that partitions the vertices into k-clusters, where the input k is a non negative integer much smaller than the cardinality of the vertex set. The robustness of the computed clustering is connected with the k-th and $(k+1)$-st eigenvalues λ_k and λ_{k+1} of the Laplacian of the weight matrix $L(W)$. In particular, if the spectral gap $g_k := \lambda_{k+1} - \lambda_k$ is small, this means that the cluster associated to k is not stable. However g_k provides an unstructured measure of robustness, that does not take into account the pattern of the original weight matrix, and hence it is not generally appropriate. We propose a different structured distance to instability, which corresponds to a *violating* matrix nearness problem, that can be computed by means of the two-level approach whose *structured inner iteration* minimizes the functional

$$F_\varepsilon(E) = \lambda_{k+1}(L(W+\varepsilon E)) - \lambda_k(L(W+\varepsilon E))$$

where ε is the perturbation size. In this case, the perturbation E considered is inside the Laplacian operator, but it is still possible to use the method proposed with some slight modifications in order to get an ODE analogous to (1.8). The gradient associated to the functional turns out to be a symmetric rank-4 matrix and hence the solution of the adaptation to this framework of equation (1.11) is symmetric and belongs to the rank-4 manifold.

Structured stabilization of a matrix

In Chapter 6 we consider a *recovering* problem: given a structured unstable matrix, in the Hurwitz sense, we look for the closest stable matrix with the same structure. In the *structured inner iteration* of the two-level approach, we construct an objective functional that has a variable number of summands that corresponds to the unstable eigenvalues of the perturbed matrix, whose expression is

$$F_\varepsilon(E) = \frac{1}{2}\sum_{i=1}^{n} \big((\mathrm{Re}\,(\lambda_i(A+\varepsilon E)) + \delta)_+\big)^2$$

where $a_+ = \max(a, 0)$ denotes the positive part of a, ε is the perturbation size and $0 < \delta \ll 1$ is a parameter. This means that the associated gradient does not have a fixed rank. However if we assume that the original matrix is almost stable, which is also the most interesting case in practice, then the gradient is low-rank. After adapting equation (1.11) to this setting, we use a rank-adaptive integrator, originally proposed in [13], that captures perfectly the features of the problem while allowing to exploit the low-rank properties.

1.6 Notation

Throughout the thesis we will use some standard mathematical notation that we list below.

- We denote the imaginary unit as $\mathrm{i} = \sqrt{-1}$. For a given complex number $z = x + \mathrm{i}y \in \mathbb{C}$ we define its real part $x := \mathrm{Re}(z)$ and its imaginary part $y := \mathrm{Im}(z)$. The complex conjugate is denoted as $\overline{z} := x - \mathrm{i}y$.

- The set of $n \times m$ complex matrices is denotes as $\mathbb{C}^{n\times m}$ and its subset of real matrices is $\mathbb{R}^{n\times m} \subseteq \mathbb{C}^{n\times m}$. A matrix $A \in \mathbb{C}^{n\times m}$ will be denoted from its entries by using the corresponding small letter as $A = (a_{i,j})$, where $1 \le i \le n$ and $1 \le j \le m$.

- The conjugate matrix of a complex matrix $A = (a_{i,j})$ is $A^* = (\overline{a}_{j,i})$, while its transpose is $A^\top = (a_{j,i})$.

- Given a matrix B, we denote by $\ker(B)$ and $\mathrm{range}(B)$ its kernel and range respectively.

- Given a subspace $\mathcal{S}$, we denote its dimension as $\dim(\mathcal{S})$. If it is ambiguous, we specify in the text if we mean the complex or the real dimension. The rank of a matrix A is defined as $\mathrm{rank}(A) := \dim(\mathrm{range}(A))$ and in some tables it is denoted as rk.

- The trace of a matrix $B = (b_{i,j})$ is $\mathrm{tr}(B) = \sum_{i=1}^{n} b_{ii}$. Throughout the thesis we often use the well-known properties

$$\mathrm{tr}(AB) = \mathrm{tr}(BA), \qquad \mathrm{tr}(A) = \mathrm{tr}(A^\top).$$

- We always consider as a matrix norm the Frobenius norm since it is induced by a matrix inner product. For all $A = (a_{i,j}) \in \mathbb{C}^{n\times n}$ and $B = (b_{i,j}) \in \mathbb{C}^{n\times n}$, the

Frobenius inner product is defined as

$$\langle A, B\rangle := \operatorname{tr}(A^*B) = \sum_{i,j=1}^{n} \overline{a}_{i,j} b_{i,j}.$$

The induced Frobenius norm is

$$\|A\|_F := \sqrt{\langle A, A\rangle} = \left(\sum_{i,j=1}^{n} |a_{i,j}|^2\right)^{\frac{1}{2}},$$

which usually is the most suitable norm in a matrix optimization context.

- For any vector $v \in \mathbb{C}^n$, we always consider the 2-norm $\|v\| = \sqrt{v^*v}$.
- Given a matrix $A \in \mathbb{R}^{n\times n}$, we denote its symmetric and skew-symmetric part, respectively, as

$$\operatorname{sym}(A) = \frac{A + A^\top}{2}, \qquad \operatorname{skew}(A) = \frac{A - A^\top}{2}.$$

- With the symbol I we denote the identity matrix, and if its dimension is unclear we write it as a subscript, e.g. I_n.
- We use the standard big O notation $\mathcal{O}(\delta)$ to denote asymptotic quantities to δ.
- Given a real number a, we denote by $(a)_+ := \max(a, 0)$ and $(a)_- := \min(a, 0)$ its positive and negative part respectively.

Chapter 2

Unstructured gradient system approach

In this chapter we focus on the unstructured problem (1.3) and we also give some general definitions used in the other chapters. More precisely, given a square complex matrix $A \in \mathbb{C}^{n\times n}$ that fulfils a certain property $\mathscr{P}$, we consider the problem

$$\mathcal{A} = \underset{\Delta\in\mathbb{C}^{n\times n}}{\arg\min}\{\|\Delta\|_F : A + \Delta \text{ does not fulfil the property } \mathscr{P}\}. \tag{2.1}$$

The choice of the Frobenius norm is motivated by the fact that it is induced by a scalar product, which makes it easier to formalize and tackle the problem. In our framework we always assume that $\mathcal{A}$ is non-empty, which implies that the minimum of (2.1) is attained, but the minimizer may not be unique. We look for an optimizer $\Delta_\star \in \mathcal{A}$ and so we introduce a suitable optimization problem, equivalent to (2.1), based on the minimization of an objective functional.

2.1 The objective functional

Let us consider a functional $\mathscr{F} : \mathbb{C}^{n\times n} \to \mathbb{R}$ such that, for a given real number $a_\star$ and for all $\Delta \in \mathbb{C}^{n\times n}$:

- the value of the functional in the zero matrix is greater than $a_\star$, that is
$$\mathscr{F}(0) > a_\star, \tag{2.2}$$
- the functional is non-greater than $a_\star$ for all the admissible perturbations, that is
$$\mathscr{F}(\Delta) \le a_\star \iff A + \Delta \text{ does not fulfil the property } \mathscr{P}, \tag{2.3}$$
- the functional is of the form
$$\mathscr{F}(\Delta) = f\left(\mathscr{H}(\Delta), \overline{\mathscr{H}(\Delta)}\right), \tag{2.4}$$
where $\mathscr{H} : \mathbb{C}^{n\times n} \to \mathbb{C}$ is holomorphic, and $f : \mathbb{C}\times\mathbb{C} \to \mathbb{C}$ is a smooth function such that
$$f(z, \overline{z}) = f(\overline{z}, z) \in \mathbb{R}, \qquad \forall z \in \mathbb{C}. \tag{2.5}$$

Assumption (2.2) ensures that the 0 matrix is not a solution of the problem, while assumption (2.3) implies that problem (2.1) can be reformulated as

$$\mathcal{A} = \underset{\Delta\in\mathbb{C}^{n\times n}}{\arg\min}\{\|\Delta\|_F : \mathscr{F}(\Delta) \le a_\star\}. \tag{2.6}$$

As already mentioned, in most applications $a_\star = 0$. Now we comment on the other features of the functional $\mathscr{F}$. Assumption (2.4) guarantees differentiability of the functional, which is a property that we need to use a gradient system approach for the optimization problem we will introduce. Concerning the function f, the two choices we consider are

$$f(z,w) = \frac{z+w}{2}, \qquad \text{and} \qquad f(z,w) = zw,$$

that for $w = \overline{z}$ corresponds to the real part and the squared absolute value of z, respectively. In some application it is also possible to use $-f$ in order to deal with the arg max case in (2.1).

For the applications we are interested in, we suppose that the function $\mathscr{H}$ is a linear combination of the eigenvalues of a function $\mathscr{L}$ of the perturbed matrix $A + \Delta$, that is

$$\mathscr{H}(\Delta) = \sum_{i=1}^{n} \gamma_i \lambda_i(\mathscr{L}(A+\Delta)), \tag{2.7}$$

where $\mathscr{L} : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ is a smooth linear operator, $\lambda_1(M), \dots, \lambda_n(M)$ denote the eigenvalues of M and $\gamma_i \in \mathbb{C}$ is a coefficient for $i = 1, \dots n$. In Chapters 4 and 6 $\mathscr{L}$ is simply the identity operator, while in Chapter 5 we consider the Laplacian operator

$$L(A) = \operatorname{diag}(A\mathbb{1}) - A, \qquad \mathbb{1} = (1,\dots,1)^\top \in \mathbb{R}^n.$$

so that $\mathscr{L}(A+\Delta) = L(A+\Delta)$. We always assume that the number of summands in the definition (2.7) is limited, say $r \ll n$, which means that many of the coefficients γ_i are 0 and thus the functional $\mathscr{F}$ depends on a small amount of eigenvalues of the perturbed matrix. This property implies a low-rank setting that is exploited in the solution of the optimization problem (2.6).

We also suppose that the eigenvalues involved in the expression of $\mathscr{H}$ are simple and this implies that $\mathscr{H}$ is holomorphic. A possible way to prove this is provided by [55, Theorem 2], where it is shown that, given a simple eigenvalue λ_0 of a matrix $Z_0 \in \mathbb{C}^{n\times n}$, it is always possible to define in a neighbourhood of Z_0 a smooth function $\lambda(Z)$ with $\lambda(Z_0) = \lambda_0$ such that $\lambda(Z)$ is an eigenvalue of Z.

Remark 2.1.1. *The assumptions on the functional $\mathscr{F}$ and on its components appear very natural for the setting we are interested in and we will always assume them throughout the thesis. We briefly collect here the two most important suppositions and their purpose:*

- *the eigenvalues involved in the definition (2.7) of $\mathscr{H}$ are simple, which guarantees regularity of the objective functional $\mathscr{F}$,*
- *the number of addends in the sum that defines $\mathscr{H}$ is $r \ll n$, which yields low-rank properties we aim to exploit.*

Both the assumptions are satisfied in many applications and in particular in those considered in this thesis.

In Chapters 4, 5 and 6 we consider different choices of the functional $\mathscr{F}$ which corresponds to different applications. In particular the expressions of $\mathscr{F}$ are of the form

§ 4: $\mathscr{F}(\Delta) = f(\lambda_{\text{target}}(A+\Delta), \overline{\lambda_{\text{target}}(A+\Delta)})$, where λ_{target} is the eigenvalue with largest real part or absolute value,

§ 5: $\mathscr{F}(\Delta) = \lambda_{k+1}(L(A+\Delta)) - \lambda_k(L(A+\Delta))$, where L is the Laplacian operator and $k < n$ is a non-negative integer,

§ 6: $\mathscr{F}(\Delta) = \frac{1}{2}\sum_{i=1}^{n}\left(\mathrm{Re}\,(\lambda_i(A+\Delta)+\delta)_+\right)^2$, where $\delta > 0$ and $a_+ = \max(a,0)$.

In Chapters 5 and 6 we have $a_\star = 0$ and $\mathscr{F}$ takes always non-negative values, while in Chapter 4 the functional can also take strictly negative values. In the dedicated chapters we describe in more detail these functionals, we give formal definitions of them and we highlight how the assumptions of Remark 2.1.1 are fulfilled.

2.2 Outline of the two-level approach

In general, the optimization problem (2.6) is highly non-convex and it is quite complicated to compute its global minima. Hence in this thesis we aim to find the local minima in the optimization problem (2.6) associated to $\mathscr{F}$ that may only provide an upper bound for the unstructured distance.

In order to do so, we introduce a two-level approach which splits the original problem into two different sub-problems, called *inner iteration* and *outer iteration.* Following the approach introduced by Guglielmi and Lubich (see e.g. [30]), we rewrite the perturbation $\Delta = \varepsilon E$, where $\varepsilon > 0$ is the perturbation size and E has unit Frobenius norm and we define the functional F_ε as

$$F_\varepsilon(E) := \mathscr{F}(\varepsilon E).$$

The *inner iteration* minimizes the functional F_ε when the perturbation size ε is fixed, while the *outer iteration* aims in finding the smallest value $\varepsilon_\star$ such that it is possible to annihilate the objective functional. The outline of the two-level method is the following:

- *Inner iteration*: For a fixed ε, compute a matrix perturbation $E_\star(\varepsilon)$ such that

$$E_\star(\varepsilon) \in \operatorname*{arg\,min}_{\|E\|_F=1} F_\varepsilon(E) = \operatorname*{arg\,min}_{\|\Delta\|=\varepsilon} \mathscr{F}(\Delta). \tag{2.8}$$

- *Outer Iteration*: Find the smallest value $\varepsilon_\star > 0$ such that

$$\varphi(\varepsilon) := F_\varepsilon(E_\star(\varepsilon)) = a_\star.$$

The *inner iteration* is the most elaborated procedure and we describe in Section 2.3 how to perform it. In contrast, the *outer iteration* is a theoretically simpler problem, once a solution of the *inner iteration* is available, and we describe it in detail in Section 2.4.

2.3 Inner Iteration: minimization with a fixed perturbation size

In this section we fix the perturbation size $\varepsilon > 0$ and we describe an ordinary differential equation that is used to solve problem (2.8). We follow a similar approach to that proposed in [32, 34, 37].

In order to find an optimal value of E that minimizes the objective functional $F_\varepsilon(E)$, we introduce a matrix differentiable path $E(t)$ of unit Frobenius norm matrices that depends on a real time variable $t \geq 0$. We denote by $\mathbb{S}_1$ the unit norm sphere in $\mathbb{C}^{n\times n}$

$$\mathbb{S}_1 = \left\{ M \in \mathbb{C}^{n\times n} : \|M\|_F = 1 \right\},$$

so that $E(t) \subseteq \mathbb{S}_1$. In this way it is possible to consider the continuous version $F_\varepsilon(E(t))$ of the objective functional, whose derivative is characterized by the following result.

Lemma 2.3.1. *Let $E(t) \subseteq \mathbb{S}_1$ be a differentiable path of matrices for $t \in [0, +\infty)$ and let ε be fixed. Then $F_\varepsilon(E(t))$ is differentiable in $[0, +\infty)$ with*

$$\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) = \varepsilon \operatorname{Re}\langle G_\varepsilon(E(t)), \dot{E}(t)\rangle,$$

where $G_\varepsilon(E)$ is the (rescaled) gradient of the objective functional $F_\varepsilon(E)$ and $\dot{E}(t) = \frac{\mathrm{d}E(t)}{\mathrm{d}t}$.

Proof. The assumption (2.4) on the definition of $\mathscr{F}$ implies that the gradient $\nabla\mathscr{H}(\Delta)$ is a well-defined matrix for all $\Delta \in \mathbb{C}^{n\times n}$. By using the abbreviations

$$f_z := \frac{\partial f(z, \overline{z})}{\partial z}, \qquad f_{\overline{z}} := \frac{\partial f(z, \overline{z})}{\partial \overline{z}},$$

property (2.5) and Proposition D.0.1 yield that $\overline{f_z} = f_{\overline{z}}$. Thus, denoting $E(t) = (e_{i,j}(t))$, yields

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) = \frac{\mathrm{d}}{\mathrm{d}t} \mathscr{F}(\varepsilon E(t)) = \frac{\mathrm{d}}{\mathrm{d}t} f\left(\mathscr{H}(\varepsilon E(t)), \overline{\mathscr{H}(\varepsilon E(t))}\right) = \\
&= f_z \cdot \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{H}(\varepsilon E(t)) + f_{\overline{z}} \cdot \frac{\mathrm{d}}{\mathrm{d}t}\overline{\mathscr{H}(\varepsilon E(t))} = f_z \cdot \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{H}(\varepsilon E(t)) + \overline{f_z} \cdot \overline{\frac{\mathrm{d}}{\mathrm{d}t}\mathscr{H}(\varepsilon E(t))} = \\
&= 2\varepsilon f_z \operatorname{Re}\left(\frac{\mathrm{d}}{\mathrm{d}t}\mathscr{H}(\varepsilon E(t))\right) = 2\varepsilon f_z \operatorname{Re}\left(\sum_{i,j=1}^{n} \frac{\mathrm{d}\mathscr{H}(\varepsilon E)}{\mathrm{d}e_{i,j}} \cdot \frac{\mathrm{d}e_{i,j}(t)}{\mathrm{d}t}\right) = \\
&= 2\varepsilon \operatorname{Re}\langle f_z \cdot \nabla\mathscr{H}(\varepsilon E(t)), \dot{E}(t)\rangle = \varepsilon \operatorname{Re}\langle G_\varepsilon(E(t)), \dot{E}(t)\rangle,
\end{aligned}$$

where $G_\varepsilon(E) = 2f_z\nabla\mathscr{H}(\varepsilon E)$ and hence, by definition, $\varepsilon G_\varepsilon(E) = \nabla F_\varepsilon(E)$. □

The matrix $G := G_\varepsilon(E(t))$ introduced in Lemma 2.3.1 gives the steepest descent direction for minimizing the objective functional F_ε. However this choice is generally not admissible, since it does not take into account the fact that the Frobenius norm of $E(t)$ must remain constantly equal to 1 for all $t \in [0, +\infty)$. In order to consider also this fact, we rewrite this unit norm restriction to get an equivalent formulation easy to deal with. Differentiating the constraint on the unit norm yields

$$0 = \frac{\mathrm{d}}{\mathrm{d}t}\|E(t)\|_F^2 = \frac{\mathrm{d}}{\mathrm{d}t}\langle E(t), E(t)\rangle = \langle \dot{E}(t), E(t)\rangle + \langle E(t), \dot{E}(t)\rangle = 2\operatorname{Re}\langle E(t), \dot{E}(t)\rangle,$$

which implies the equivalence

$$\|E(t)\|_F = 1 \quad \forall t \in [0, +\infty) \iff \|E(0)\| = 1 \text{ and } \operatorname{Re}\langle E(t), \dot{E}(t)\rangle = 0. \tag{2.9}$$

By taking advantage of relation (2.9), the next result shows how to select the best direction to follow in order to fulfil the unit Frobenius norm condition on $E(t)$.

Lemma 2.3.2. *Given $E \in \mathbb{S}_1$ and $G \in \mathbb{C}^{n\times n}$, the solution of the optimization problem*

$$\underset{Z\in\mathbb{S}_1,\ \operatorname{Re}\langle Z,E\rangle=0}{\arg\min} \quad \operatorname{Re}\langle G, Z\rangle$$

is

$$Z_\star = \frac{-G + \mathrm{Re}\langle G, E\rangle E}{\| - G + \mathrm{Re}\langle G, E\rangle E\|_F}.$$

Proof. Consider the real vectorized forms z, e and g in $\mathbb{R}^{2n^2}$ of the matrices Z, E and G. Since the real part of the Frobenius inner product in $\mathbb{C}^{n\times n}$ turns into the standard scalar product of $\mathbb{R}^{2n^2}$, the claim is equivalent to show that

$$\underset{\|z\|_2=1,\ z^\top e=0}{\arg\min}\ g^\top z = \frac{-g + (e^\top g)e}{\| - g + (e^\top g)e\|_2},$$

which follows from Proposition D.0.2. □

By giving to the variable Z the role of the derivative of E, Lemma 2.3.2 provides an ordinary differential equation for the perturbation E that guarantees the best admissible descent direction in order to minimize the time derivative of $F_\varepsilon(E(t))$. Omitting the normalization factor, which actually corresponds to a time rescaling, yields

$$\dot{E} = -G_\varepsilon(E) + \mathrm{Re}\langle G_\varepsilon(E), E\rangle E. \tag{2.10}$$

By construction of this ordinary differential equation, we have that $\mathrm{Re}\langle E, \dot{E}\rangle = 0$ along its solutions and so the unit Frobenius norm of E is conserved.

Now we investigate the properties of equation (2.10). We begin by showing that this ODE is a gradient system, meaning that along its trajectories the objective functional monotonically decreases.

Theorem 2.3.3. *Let $E(t)$ be a solution of* (2.10) *with starting value $E(0)$ of unit Frobenius norm. Then*

$$\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) \le 0.$$

Proof. We show the explicit rate of decay of the objective functional. Since $E(t)$ satisfies (2.10) and relation (2.9) implies that it has unit Frobenius norm, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) = \mathrm{Re}\langle G_\varepsilon(E(t)), \dot{E}(t)\rangle = -\|G_\varepsilon(E(t))\|_F^2 + (\mathrm{Re}\langle G_\varepsilon(E(t)), E(t)\rangle)^2 \le 0, \tag{2.11}$$

where the last inequality follows from the Cauchy-Schwarz inequality, since $\|E\|_F = 1$. □

The next result states that it is possible to characterize a stationary point $E_\star$ of equation (2.10), under the assumption that the gradient $G_\varepsilon(E_\star)$ does not vanish. In Chapters 4, 5 and 6 we discuss in detail the degenerate case when the gradient is 0 and we show that it is a non-generic event in the applications considered.

Theorem 2.3.4. *Let $E(t) \subseteq \mathbb{S}_1$ be a solution of equation* (2.10) *passing through $E_\star = E(t_\star)$ at time $t_\star > 0$ and assume that $G_\varepsilon(E_\star) \neq 0$. Then the following facts are equivalent:*

1. $\left.\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t))\right|_{t=t_\star} = 0$
2. *$E_\star$ is a stationary point of* (2.10)
3. *$E_\star$ is a non-zero real multiple of $G_\varepsilon(E_\star)$*

Proof. The implications 3. ⇒ 2. and 2. ⇒ 1. follow from equation (2.10) and Lemma 2.3.1 respectively. To conclude the proof we show that 1. ⇒ 3.. Assumption 1. implies that (2.11) is actually an equality, that is

$$\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E_\star) = -\|G_\varepsilon(E_\star)\|_F^2 + \left(\mathrm{Re}\langle G_\varepsilon(E_\star), E_\star\rangle\right)^2 = 0.$$

Since $G_\varepsilon(E_\star) \neq 0$, the Cauchy-Schwarz inequality in (2.11) would be strict unless $G_\varepsilon(E_\star)$ is a real multiple of $E_\star$, which implies 3.. □

Theorem 2.3.4 shows that, up to degenerate cases where the gradient vanishes, there is an equivalence between the stationary points of the gradient system (2.10) and the local minima of F_ε. Thanks to Theorem 2.3.3, for any starting point $E_0 = E(0)$ of unit Frobenius norm, the integration of (2.10) leads to a stationary point $E_\star$ and it is ensured that other scenarios like periodic orbits of the system are avoided, since the derivative of the objective functional is always non-positive and vanishes only in stationary points. Hence it is always guaranteed that integrating (2.10) provides a local minima of F_ε, no matter the starting point chosen.

2.3.1 Low-rank trajectory

In many applications, it turns out that the matrix $G_\varepsilon(E)$ is low-rank. In particular, as already assumed by Remark 2.1.1, the function $\mathscr{H}$ used in the definition of the functional $\mathscr{F}$ in (2.4) is a combination of few eigenvalues of the perturbed matrix $\mathscr{L}(A + \varepsilon E)$, say $r \ll n$, and then the gradient $G_\varepsilon(E)$ turns out to be a linear combination of the r outer products of the left and right eigenvectors associated to the eigenvalues considered in $\mathscr{H}$. More precisely, in Chapter 4 we have $r = 1$, in Chapter 5 we have $r = 4$, while in Chapter 6 the value of r changes during the trajectory.

When this situation arises, Theorem 2.3.4 ensures that the stationary points are proportional to a rank-r matrix. This fact suggests to design a new trajectory for the perturbation $E(t)$ contained in the rank-r manifold (see Proposition A.0.1 for more details about the rank-r manifold)

$$\mathcal{M}_r = \{M \in \mathbb{C}^{n\times n} \ : \ \mathrm{rank}(M) = r\}.$$

A fundamental tool used for building such a trajectory is the orthogonal projection P_E, with respect to the Frobenius inner product, at a point $E \in \mathcal{M}_r$ onto the tangent space $\mathcal{T}_E\mathcal{M}_r$ (see Proposition A.0.2 for more details). Before giving an explicit expression for P_E, we introduce decomposition that generalizes the well-known Singular Value Decomposition (SVD). Given $E \in \mathcal{M}_r$, an SVD-like is

$$E = USV^* \qquad U, V \in \mathbb{C}^{n\times r} \text{ such that } U^*U = V^*V = I_r, \quad S \in \mathbb{C}^{r\times r} \text{ invertible,} \tag{2.12}$$

where it is not required that the matrix S is diagonal with non-negative real entries, but only that it is invertible (see [51] for further details). Thanks to this decomposition of E, for all $M \in \mathbb{C}^{n\times n}$ it is possible to give the explicit formula for P_E (see Proposition B.0.3 for a proof)

$$P_E(M) = M - (I - UU^*)M(I - VV^*) = UU^*M + MVV^* - UU^*MVV^*.$$

We still call $E(t)$ the new low-rank trajectory, even though it is generally not a solution of equation (2.10). Imposing that $E(t)$ is contained in $\mathcal{M}_r$ requires that the time derivative $\dot{E}(t)$ belongs to the the tangent space $\mathcal{T}_{E(t)}\mathcal{M}_r$ for all t. This provides the idea for the definition of a new ODE for the low-rank trajectory: we project the right-hand side of equation (2.10) onto the tangent space and we get

$$\dot{E} = P_E\left(-G_\varepsilon(E) + \mathrm{Re}\langle G_\varepsilon(E), E\rangle E\right), \tag{2.13}$$

where Doležal's theorem (see [18]) guarantees that $E(t)$ can be decomposed as an analytic SVD-like (see e.g. [8])

$$E(t) = U(t)S(t)V(t)^*, \quad U(t), V(t) \in \mathbb{C}^{n\times r}, \quad S(t) \in \mathbb{C}^{r\times r} \text{ invertible},$$

where $U(t)$ and $V(t)$ have orthonormal columns. By observing that $P_E(E) = E$, we can rewrite equation (2.13) as

$$\dot{E} = -P_E(G_\varepsilon(E)) + \mathrm{Re}\langle G_\varepsilon(E), E\rangle E$$

and by using the decomposition of E and the formula for P_E we get

$$\dot{U}SV^* + U\dot{S}V^* + US\dot{V}^* = -UU^*G - GVV^* + UU^*GVV^* + \mu USV^*,$$

where $G = G_\varepsilon(E)$ and $\mu = \mathrm{Re}\langle G, E\rangle$. In order to fulfil the gauge conditions $U^*\dot{U} = V^*\dot{V} = 0$, we choose the matrices $\dot{U}, \dot{S}$ and $\dot{V}$ as the solutions of following system

$$\begin{cases} \dot{U} = -(I - UU^*)GVS^{-1} \\ \dot{S} = -U^*GV + \mu S \\ \dot{V} = (I - VV^*)G^*US^{-*} \end{cases}, \tag{2.14}$$

which is equivalent to (2.13). In this way it is also possible to determine uniquely the matrices $\dot{U}, \dot{S}$ and $\dot{V}$ from $\dot{E} = \dot{U}SV^* + U\dot{S}V^* + US\dot{V}^*$, since

$$\begin{cases} \dot{U} = (I - UU^*)\dot{E}VS^{-1} \\ \dot{S} = U^*\dot{E}V \\ \dot{V} = (I - VV^*)\dot{E}^*US^{-*} \end{cases}.$$

System (2.14) consists of two matrix ODEs of dimension $n \times r$ and one of dimension $r \times r$, which, from the computational point of view, are preferable to the $n \times n$ equation (2.10). The reason behind the choice of the SVD-like, rather than the standard SVD, is motivated by the fact that, in general, system (2.14) does not preserve the diagonal structure of S along the trajectory. We observe that a classical integration, e.g. by means of Euler's method, of system (2.14) is generally not suitable, because of the presence of the inverse of S that may cause numerical issues. Thus it is generally preferable to use a splitting method which overcomes this problem. We give more details about the integration of this low-rank system in the applications proposed in the next chapters.

After introducing the new low-rank ODE (2.13), we describe its features and how it is related with the original gradient system (2.10). First of all we show that also (2.13) is a gradient system that preserves the unit Frobenius norm of the solution $E(t)$.

Theorem 2.3.5. *Let $E(t)$ be a solution of* (2.13) *with starting value $E(0)$ of unit Frobenius norm. Then $E(t) \in \mathbb{S}_1$ and*

$$\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) \le 0.$$

Proof. Let $G = G_\varepsilon(E)$ for short. In order to show both the claims, we notice that

$$\operatorname{Re}\langle G, E\rangle = \operatorname{Re}\langle G, P_E(E)\rangle = \operatorname{Re}\langle P_E(G), E\rangle,$$

since P_E is an orthogonal projection such that $P_E(E) = E$. Thus, since $\|E(0)\|_F = 1$, (2.13) yields

$$\operatorname{Re}\langle E, \dot{E}\rangle = -\operatorname{Re}\langle E, P_E(G)\rangle + \operatorname{Re}\langle P_E(G), E\rangle E\rangle \|E\|_F^2 = 0,$$

that is the unit Frobenius norm of $E(t)$ is preserved along the trajectory. Finally the monotonicity of the objective functional is given again by the Cauchy-Schwarz inequality, since

$$\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) = \operatorname{Re}\langle G, \dot{E}\rangle = -\|P_E(G)\|_F^2 + (\operatorname{Re}\langle P_E(G), E(t)\rangle)^2 \leq 0,$$

where we have used that $\operatorname{Re}\langle G, P_E(G)\rangle = \|P_E(G)\|_F^2$. □

By using the same arguments of Theorem 2.3.5 and Theorem 2.3.4, it is also possible to extend in the same way the latter result for equation (2.13).

Theorem 2.3.6. *Let $E(t) \subseteq \mathbb{S}_1$ be a solution of equation* (2.13) *passing through $E_\star = E(t_\star)$ at time $t_\star > 0$ and assume that $P_{E_\star} G_\varepsilon(E_\star) \neq 0$. Then the following facts are equivalent:*

1. $\left.\dfrac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t))\right|_{t=t_\star} = 0$
2. *$E_\star$ is a stationary point of* (2.10)
3. *$E_\star$ is a non-zero real multiple of $P_{E_\star} G_\varepsilon(E_\star)$*

Proof. It follows the same steps of the proof of Theorem 2.3.4. □

Once shown that the original ODE and its projected version are both gradient systems that preserve the unit norm of the trajectory, we need to ensure that the integration of (2.13) leads to the same stationary points that are minimizers of the objective functional F_ε. The following result guarantees that the two equations (2.10) and (2.13) share the same stationary points, that is integrating the second equation does not introduce nor lose minimizers of the objective functional F_ε.

Theorem 2.3.7. *Let $E_\star \in \mathbb{S}_1$ be such that $G_\star = G_\varepsilon(E_\star) \in \mathcal{M}_r$. Then*

$$E_\star \text{ is a stationary point of (2.10)} \iff E_\star \text{ is a stationary point of (2.13).}$$

Proof. ⇒) It is trivial, since $P_E(0) = 0$.

⇐) Assume that $E_\star$ is a stationary point of (2.13). Thanks to Theorem 2.3.4, the claim is equivalent to show that $E_\star$ is a real multiple of $G_\star$. By Theorem 2.3.6, we know that there exists a matrix $W \in \mathbb{C}^{n\times n}$ with $P_E(W) = 0$ such that, for some non-zero $\mu \in \mathbb{R}$,

$$E_\star = \mu G_\star + W. \tag{2.15}$$

By the definition of P_E, we have

$$W = (I - U_\star U_\star^*) W (I - V_\star V_\star^*)$$

where we have considered an SVD-like of $E_\star \in \mathcal{M}_r$ as $E_\star = U_\star S_\star V_\star^*$, where $S_\star \in \mathbb{C}^{r\times r}$. Pre-multiplying equation (2.15) by $U_\star^*$ and post-multiplying it by $V_\star$ yields

$$S_\star V_\star^* = \mu U_\star^* G_\star, \qquad U_\star S_\star = \mu G_\star V_\star,$$

which shows that $E_\star, G_\star \in \mathcal{M}_r$ have the same kernel and range and thus we are in the hypothesis of Proposition D.0.4, which guarantees $G_\star = U_\star U_\star^* G_\star$. We pre-multiply by $U_\star U_\star^*$ equation (2.15) to get

$$E_\star = \mu U U_\star^* G_\star = \mu G_\star,$$

and hence $E_\star$ is a non-zero real multiple of $G_\star$ and the claim follows from Theorem 2.3.5. □

The implementation of the *inner iteration* is discussed more in detail for each application in the next chapters. Indeed the expression of the gradient takes a more important role in this case and hence it is needed to differentiate the cases.

2.4 Outer iteration: tuning the perturbation size

Once that a computation of the optimizers is available for a given $\varepsilon > 0$, we need to determine an optimal value for the perturbation size $\varepsilon_\star$. Let $E_\star(\varepsilon)$ be a solution of the optimization problem (2.8) and consider the function

$$\varphi(\varepsilon) := F_\varepsilon(E_\star(\varepsilon)).$$

This function is non-negative and we define $\varepsilon_\star$ as the smallest zero of $\varphi - a_\star$. Assuming that the eigenvalues of $\mathscr{L}(A + \varepsilon E_\star(\varepsilon))$ that appear in the definition (2.7) of $\mathscr{H}$ are simple, for $0 \le \varepsilon < \varepsilon_\star$, yields that φ is a differentiable function in the interval $[0, \varepsilon_\star)$. The aim of the *outer iteration* is to estimate $\varepsilon_\star$, which is the solution of the optimization problem (1.3) and hence an approximation of the distance sought. In order to solve this problem, we use a combination of the well-known Newton and bisection methods, which provides an approach similar to [24, 28, 33] or [34]. If the current approximation ε is smaller than $\varepsilon_\star$, it is possible to exploit Newton's method, since φ is differentiable there (see Lemma 2.4.1); otherwise, if $\varepsilon > \varepsilon_\star$, we use the bisection method. The following result provides a simple formula for the first derivative of φ, which is cheap to compute, making the Newton method easy to apply.

Lemma 2.4.1. *For $0 \le \varepsilon < \varepsilon_\star$ we have*

$$\varphi'(\varepsilon) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} F_\varepsilon(E_\star(\varepsilon)) = \langle G_\varepsilon(E_\star(\varepsilon)), E_\star(\varepsilon)\rangle = -\|G_\varepsilon(E_\star(\varepsilon))\|_F \le 0.$$

Proof. As shown in Lemma 2.3.1, we get

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon} F_\varepsilon(E_\star(\varepsilon)) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} \mathscr{F}(\varepsilon E_\star(\varepsilon)) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} f\left(\mathscr{H}(\varepsilon E_\star(\varepsilon)), \overline{\mathscr{H}(\varepsilon E_\star(\varepsilon))}\right) =$$

$$= f_z \cdot \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathscr{H}(\varepsilon E_\star(\varepsilon)) + f_{\overline{z}} \cdot \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\overline{\mathscr{H}(\varepsilon E_\star(\varepsilon))} = f_z \cdot \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathscr{H}(\varepsilon E_\star(\varepsilon)) + \overline{f_z} \cdot \overline{\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathscr{H}(\varepsilon E_\star(\varepsilon))} =$$

$$= 2 f_z \operatorname{Re}\left(\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathscr{H}(\varepsilon E_\star(\varepsilon))\right) = 2 f_z \operatorname{Re}\left(\sum_{i,j=1}^{n} \frac{\mathrm{d}\mathscr{H}(\varepsilon E_\star)}{\mathrm{d}e_{i,j}} \cdot \left(e_{i,j} + \varepsilon \frac{\mathrm{d}e_{i,j}(\varepsilon)}{\mathrm{d}\varepsilon}\right)\right) =$$

$$= 2\,\mathrm{Re}\langle f_z \cdot \nabla \mathscr{H}(\varepsilon E_\star(\varepsilon)), E_\star(\varepsilon) + \varepsilon E_\star'(\varepsilon)\rangle = \mathrm{Re}\langle G_\varepsilon(E_\star(\varepsilon)), E_\star(\varepsilon) + \varepsilon E_\star'(\varepsilon)\rangle,$$

where $E_\star'(\varepsilon)$ is the derivative with respect to ε of $E_\star(\varepsilon)$. Since $E_\star(\varepsilon)$ is a unit norm stationary point of (2.10) and (2.13), and a zero of the derivative of the objective functional F_ε, then $G_\varepsilon(E_\star(\varepsilon))$ is a negative multiple of $E_\star$. Thus $G_\varepsilon(E_\star(\varepsilon)) = -\|G_\varepsilon(E_\star(\varepsilon))\|_F \; E_\star(\varepsilon)$ and, since $\|E_\star(\varepsilon)\|_F = 1$ for all ε, we have

$$\mathrm{Re}\langle G_\varepsilon(E_\star(\varepsilon)), E_\star'(\varepsilon)\rangle = -\frac{\|G_\varepsilon(E_\star(\varepsilon))\|_F}{2}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\|E_\star(\varepsilon)\|_F^2 = 0,$$

which yields the claim. □

Remark 2.4.2. *The assumption on the simplicity of the eigenvalues of $\mathscr{L}(A+\varepsilon E_\star(\varepsilon))$ can be supported by similar reasons as the ones stated in Remark 4.2.2. In any case the root-finding technique used in this context relies also on a bisection method that does not need differentiability to hold. In this way Newton's method is replaced when the differentiability is lost.*

Algorithm 1 provides the outline of the *outer iteration* in order to solve problem (2.1).

Algorithm 1 Outer iteration

Input: A matrix A, an interval and an initial guess $\varepsilon_0 \in [\varepsilon_{\mathrm{lb}}, \varepsilon_{\mathrm{ub}}]$ for $\varepsilon_\star$, a tolerance τ_{out} and a maximum number of iterations niter

Output: The value $\varepsilon_\star$ solution of problem (2.1) and the associated minimizer $E_\star(\varepsilon_\star)$

```
 1: Compute a stationary point $E_\star(\varepsilon_0)$ of (2.10) and (2.13) (inner iteration).
 2: Set $\ell = 0$.
 3: while $\ell <$ niter and $\varepsilon_{\mathrm{ub}} - \varepsilon_{\mathrm{lb}} > \tau_{\mathrm{out}}$ do
 4:     if $\varphi(\varepsilon_\ell) - a_\star <$ toler then
 5:         Set $\varepsilon_{\mathrm{ub}} := \min(\varepsilon_{\mathrm{ub}}, \varepsilon_\ell)$.
 6:         Set $\varepsilon_{\ell+1} := \frac{\varepsilon_{\mathrm{lb}}+\varepsilon_{\mathrm{ub}}}{2}$ (bisection step).
 7:     else
 8:         Set $\varepsilon_{\mathrm{lb}} := \max(\varepsilon_{\mathrm{lb}}, \varepsilon_\ell)$.
 9:         Compute $\varphi(\varepsilon_\ell)$ and $\varphi'(\varepsilon_\ell)$.
10:         Update $\varepsilon_{\ell+1} := \varepsilon_\ell - \frac{\varphi(\varepsilon_\ell)}{\varphi'(\varepsilon_\ell)}$ (Newton step).
11:     end if
12:     if $\varepsilon_{\ell+1} \notin [\varepsilon_{\mathrm{lb}}, \varepsilon_{\mathrm{ub}}]$ then
13:         Set $\varepsilon_{\ell+1} := \frac{\varepsilon_{\mathrm{lb}}+\varepsilon_{\mathrm{ub}}}{2}$.
14:     end if
15:     Set $\ell := \ell + 1$.
16:     Compute $E_\star(\varepsilon_\ell)$ by integrating problem (2.8) with starting value $E_\star(\varepsilon_{\ell-1})$.
17: end while
18: Return $\varepsilon_\star := \varepsilon_\ell$ and $E_\star(\varepsilon_\star)$.
```

The integration of the *inner iteration* in steps 1 and 16 in Algorithm 1 is discussed for the different applications in Chapters 4,5 and 6.

Chapter 3

Structured gradient system approach

In this chapter we focus on the structured problem (1.4), which is an extension of problem (1.3). In this framework we consider a subset $\mathcal{S} \in \mathbb{C}^{n\times n}$ that describes the admissible set of perturbation allowed, i.e. the structure. Given a square complex matrix $A \in \mathbb{C}^{n\times n}$ that fulfils a certain property $\mathscr{P}$, we consider the problem

$$\mathcal{A}_{\mathcal{S}} = \arg\min_{\Delta\in\mathcal{S}}\{\|\Delta\|_F : A+\Delta \text{ does not fulfil the property } \mathscr{P}\}. \tag{3.1}$$

Although the problem is well-defined also for all $A \in \mathbb{C}^{n\times n}$, we usually assume that $A \in \mathcal{S}$. Indeed this is the most interesting case where the perturbation is asked to preserve the same structure of A.

From now on we suppose that $\mathcal{A}_{\mathcal{S}}$ is not empty, that is the minimum of (2.1) is attained, and we look for a minimizer $\Delta_\star \in \mathcal{A}_{\mathcal{S}}$. This assumption is relevant, since in general the constraint on the structure that appears in problem (1.4) and (3.1) may be too restrictive, especially if the dimension of $\mathcal{S}$ is small. Example C.0.1 in the appendix shows a case where problem (3.1) does not have a solution and neither an infimum, since $A+\Delta$ fulfils the property $\mathscr{P}$ for all $\Delta \in \mathbb{C}^{n\times n}$.

As done for the unstructured case, we consider a real number $a_\star$ and a functional $\mathscr{F} : \mathbb{C}^{n\times n} \to \mathbb{R}$ that satisfies properties (2.2), (2.3), (2.4) and (2.5) and we reformulate problem (3.1) as

$$\mathcal{A}_{\mathcal{S}} = \arg\min_{\Delta\in\mathcal{S}}\{\|\Delta\|_F : \mathscr{F}(\Delta) \le a_\star\}.$$

We make use of a two-level approach similar to that introduced for the unstructured problem (2.1). We rewrite the perturbation $\Delta = \varepsilon E$, where $\varepsilon > 0$ is the perturbation size and E is a unit Frobenius norm matrix that is

$$E \in \mathcal{S}_1 := \{M \in \mathcal{S} \;:\; \|M\|_F = 1\}$$

and we define again the functional F_ε as

$$F_\varepsilon(E) := \mathscr{F}(\varepsilon E).$$

The outline of the structured-two-level method is the following:

- *Structured Inner iteration*: For a fixed ε, compute a matrix perturbation $E_\star(\varepsilon) \in \mathcal{S}_1$ such that
$$E_\star(\varepsilon) \in \arg\min_{E\in\mathcal{S}_1} F_\varepsilon(E) = \arg\min_{\Delta\in\mathcal{S},\ \|\Delta\|=\varepsilon} \mathscr{F}(\Delta). \tag{3.2}$$

- *Structured Outer Iteration*: Find the smallest value $\varepsilon_\star > 0$ such that

$$\varphi(\varepsilon) := F_\varepsilon(E_\star(\varepsilon)) = a_\star.$$

Also in this case the *structured inner iteration* is the most elaborated procedure while the *structured outer iteration* is theoretically simpler to solve, once a solution of (3.2) is available (see Section 3.2). In Section 3.1 we retrace the ideas behind the solution of the *inner iteration* and we generalize them for the *structured inner iteration.* This extension is straightforward for some results, but it requires more effort for others. For instance the introduction of a low-rank ODE analogous to that of (2.13) for the structured case is not trivial and it is one of the main novelties of this thesis. In particular it shows the intrinsic low-rank property of the problem also in the structured case.

Throughout this chapter and the next ones, we make use of the orthogonal projection $\Pi_\mathcal{S}$, with respect to the Frobenius inner product, onto the subspace $\mathcal{S}$. The function $\Pi_\mathcal{S}$ is linear and, for all $M, N \in \mathbb{C}^{n\times n}$, it fulfils the condition

$$\operatorname{Re}\langle \Pi_\mathcal{S}(M), N\rangle = \operatorname{Re}\langle \Pi_\mathcal{S}(M), \Pi_\mathcal{S}(N)\rangle = \operatorname{Re}\langle M, \Pi_\mathcal{S}(N)\rangle, \tag{3.3}$$

which follows directly from the definition of an orthogonal projection and it is equivalent to impose that $\Pi_\mathcal{S}(M)$ is the element of $\mathcal{S}$ that is the nearest matrix to a given matrix M in the distance induced by the Frobenius norm (see Proposition B.0.2). We always assume that an explicit expression for $\Pi_\mathcal{S}$ is available. For instance in the case $\mathcal{S} = \mathbb{R}^{n\times n}$, then it is easy to show that $\Pi_\mathcal{S}(M) = \operatorname{Re}(M)$, while when $\mathcal{S}$ is defined by the sparsity pattern of the given matrix $A = (a_{i,j})$, then, for all $M = (m_{i,j}) \in \mathbb{C}^{n\times n}$, we have (see Proposition B.0.4)

$$(\Pi_\mathcal{S}(M))_{i,j} = \begin{cases} m_{i,j} & \text{if } a_{i,j} \neq 0 \\ 0 & \text{otherwise} \end{cases}.$$

In Propositions B.0.6, B.0.7 and B.0.8 it is possible to find the expression of $\Pi_\mathcal{S}$ in some other interesting choices of the subspace $\mathcal{S}$. For convenience, we sometimes omit the parenthesis when writing the projection, meaning that $\Pi_\mathcal{S} M = \Pi_\mathcal{S}(M)$.

3.1 Structured inner iteration

In this section, given a fixed perturbation size $\varepsilon > 0$, we solve problem (3.2) by generalizing the approach presented in Section 2.3, where we introduced a matrix ordinary differential equation whose stationary points coincide with the minimizers sought. In particular we describe how to deal with the structure constraint on $\mathcal{S}$ as proposed in [34].

We introduce a matrix differentiable path $E(t) \subseteq \mathcal{S}_1$ of structured unit Frobenius norm matrices that depends on a real time variable $t \geq 0$ so that we can obtain a differentiation formula for the objective functional.

Lemma 3.1.1. *Let $E(t) \subseteq \mathcal{S}_1$ be a differentiable path of matrices for $t \in [0, +\infty)$ and let ε be fixed. Then $F_\varepsilon(E(t))$ is differentiable in $[0, +\infty)$ with*

$$\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) = \varepsilon \operatorname{Re}\langle G_\varepsilon(E(t)), \dot{E}(t)\rangle = \varepsilon \operatorname{Re}\langle \Pi_\mathcal{S} G_\varepsilon(E(t)), \dot{E}(t)\rangle,$$

where $G_\varepsilon(E)$ *is the (rescaled) gradient of the objective functional* $F_\varepsilon(E)$ *and* $\dot{E}(t) = \frac{\mathrm{d}E(t)}{\mathrm{d}t}$.

Proof. It is identitical to the proof of Lemma 2.3.1. The second equality follows form the property (3.3), since the assumption that $E(t)$ is contained in $\mathcal{S}$ implies that $\dot{E} = \Pi_{\mathcal{S}}\dot{E} \in \mathcal{S}$. □

The next lemma provides a result for detecting the best direction to follow in order to minimize the objective functional F_ε, to fulfil the unit Frobenius norm condition on $E(t)$ and to preserve the structure $\mathcal{S}$.

Lemma 3.1.2. *Given* $E \in \mathcal{S}_1$ *and* $P \in \mathcal{S}$, *the solution of the optimization problem*

$$\underset{Z\in\mathcal{S}_1,\ \mathrm{Re}\langle Z,E\rangle=0}{\arg\min} \quad \mathrm{Re}\langle P, Z\rangle$$

is

$$Z_\star = \frac{-P + \mathrm{Re}\langle P, E\rangle E}{\| -P + \mathrm{Re}\langle P, E\rangle E\|_F}.$$

Proof. It follows the same approach of Lemma 2.3.2, since the assumptions yield that $Z_\star \in \mathcal{S}_1$ and hence it is admissible. □

By considering $P = \Pi_{\mathcal{S}}G_\varepsilon(E)$ in Lemma 3.1.2, we can write an ordinary differential equation for the perturbation E that guarantees the best admissible descent direction in order to minimize the time derivative of $F_\varepsilon(E(t))$ and that also preserves the structure constraint on the trajectory and its unit norm. Omitting the normalization factor, which actually corresponds to a time rescaling, yields

$$\dot{E} = -\Pi_{\mathcal{S}}G_\varepsilon(E) + \mathrm{Re}\langle \Pi_{\mathcal{S}}G_\varepsilon(E), E\rangle E. \tag{3.4}$$

It turns out that equations (2.10) and (3.4) have many features in common, which are highlighted in the next section.

3.1.1 The structured gradient system properties

Similarly to Theorem 2.3.3, the next result shows that also equation (3.4) is a gradient system.

Theorem 3.1.3. *Let* $E(t)$ *be a solution of* (3.4) *with starting value* $E(0) \in \mathcal{S}_1$. *Then*

$$\frac{\mathrm{d}}{\mathrm{d}t}F_\varepsilon(E(t)) \le 0.$$

Proof. As in the unstructured case, we show the explicit rate of decaying of the objective functional. Since $E(t)$ satisfies (3.4) and relation (2.9) implies that it has unit Frobenius norm, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}F_\varepsilon(E(t)) = \mathrm{Re}\langle \Pi_{\mathcal{S}}G_\varepsilon(E), \dot{E}\rangle = -\|\Pi_{\mathcal{S}}G_\varepsilon(E)\|_F^2 + (\mathrm{Re}\langle \Pi_{\mathcal{S}}G_\varepsilon(E), E\rangle)^2 \le 0 \tag{3.5}$$

and the Cauchy-Schwarz inequality ensures that the derivative is non-positive. □

Also in this case it is possible to characterize the stationary points of the gradient system (3.4) similarly to Theorem 2.3.4.

Theorem 3.1.4. *Let $E(t) \subseteq \mathcal{S}_1$ be a solution of equation* (3.4) *passing through $E_\star = E(t_\star)$ at time $t_\star > 0$ and assume that $\Pi_{\mathcal{S}} G_\varepsilon(E_\star) \neq 0$. Then the following facts are equivalent:*

1. $\left.\dfrac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t))\right|_{t=t_\star} = 0,$

2. *$E_\star$ is a stationary point of* (3.4)*,*

3. *$E_\star$ is a non-zero real multiple of $\Pi_{\mathcal{S}} G_\varepsilon(E_\star)$.*

Proof. The implications 3. ⇒ 2. and 2. ⇒ 1. follow from equation (3.4) and Lemma 3.1.1 respectively. To conclude the proof we show that 1. ⇒ 3.. Assumption 1. implies that (3.5) is actually an equality, that is

$$\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E_\star) = -\|\Pi_{\mathcal{S}} G_\varepsilon(E_\star)\|_F^2 + \left(\mathrm{Re}\langle \Pi_{\mathcal{S}} G_\varepsilon(E_\star), E_\star\rangle\right)^2 = 0.$$

Since $\Pi_{\mathcal{S}} G_\varepsilon(E_\star) \neq 0$, the Cauchy-Schwarz inequality in (3.5) is strict unless $\Pi_{\mathcal{S}} G_\varepsilon(E_\star)$ is a real multiple of $E_\star$, which implies 3.. □

Also for the structured setting, the characterization of the stationary points of the gradient system relies on the case where the gradient does not vanish, but here it is further required that also the projection onto the structure of the gradient does not vanish. When the dimension of $\mathcal{S}$ is high enough, the annihilation of the gradient and its projection is non-generic, since $G_\varepsilon(E_\star) \neq 0$ generally guarantees $\Pi_{\mathcal{S}} G_\varepsilon(E_\star) \neq 0$. However, if the dimension of $\mathcal{S}$ is too small, it may happen that $G_\varepsilon(E_\star) \neq 0$, while $\Pi_{\mathcal{S}} G_\varepsilon(E_\star) = 0$, which means that the problem may be unsolvable due to the excessive strict constraint given by $\mathcal{S}$, for instance as shown in Example C.0.1. In the next chapters we take into account this issue and we specify when some further assumptions on the non-vanishing projected gradient are needed.

A difference between the unstructured and structured characterization of the stationary point $E_\star$ (that is Theorem 2.3.4 and Theorem 3.1.4 respectively) concerns the low-rank properties associated to the gradient $G_\varepsilon(E_\star)$. While in the unstructured case it is evident that $E_\star$ has the same rank as $G_\varepsilon(E_\star)$, in the structured case we can only say that this is true up to the projection $\Pi_{\mathcal{S}}$. Unfortunately this means that $E_\star$ is generically full-rank, since the projection onto the structure usually destroys the low-rank property. The recovery of the low-rank feature is still possible, but it is more complicated and we describe how to deal with it in the following sections.

3.1.2 A low-rank ODE for the structured problem

Until this point, the generalization of the *inner iteration* to the *structured inner iteration* is quite simple, but the extension of how to exploit the low-rank insights is less straightforward. In this section we retrace a similar idea to what is done for the unstructured case, but we introduce an auxiliary new low-rank perturbation matrix Y that will take the role of the pre-projection of the full-rank perturbation E.

As said for the unstructured case, in many applications it turns out that the matrix $G_\varepsilon(E)$ is low-rank and it is a linear combination of the r outer products of the left and right eigenvectors associated to the eigenvalues considered in the function $\mathscr{H}$ in the objective function $\mathscr{F}$. In order to exploit this fact, we wish to introduce a low-rank matrix path whose projection onto $\mathcal{S}$ consists of a suitable matrix path that acts as the matrix perturbation.

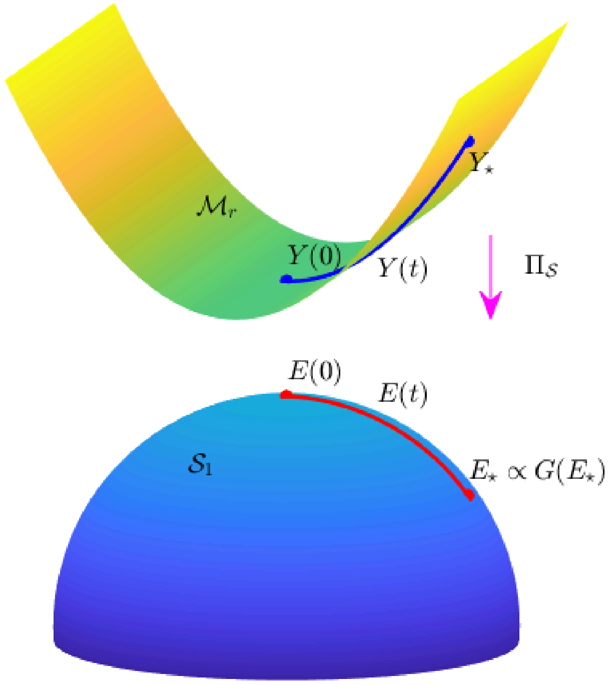


FIGURE 3.1: Representation of the low-rank trajectory $Y(t)$ solution of (3.9) and of its associated structured projection $E(t) = \Pi_{\mathcal{S}} Y(t)$.

We observe that solutions of (3.4) can be rewritten as $E(t) = \Pi_{\mathcal{S}} Z(t)$, where $Z(t)$ solves the ordinary differential equation

$$\dot{Z} = -G_\varepsilon(\Pi_{\mathcal{S}} Z) + \mathrm{Re}\langle G_\varepsilon(\Pi_{\mathcal{S}} Z), \Pi_{\mathcal{S}} Z\rangle Z, \tag{3.6}$$

but this does not guarantee that $Z(t)$ is a low-rank matrix path. Thus, we introduce a new perturbation that, for simplicity, we call $E(t)$ (but actually does not coincide with the solution of (3.4)) and we look for a low-rank matrix path $Y(t) \subseteq \mathcal{M}_r$ such that

$$E(t) = \Pi_{\mathcal{S}} Y(t), \qquad t \in [0, +\infty).$$

Taking inspiration from equation (3.6), we project the right hand side and we get

$$\dot{Y} = P_Y\left(-G_\varepsilon(\Pi_{\mathcal{S}} Y) + \mathrm{Re}\langle P_Y(G_\varepsilon(\Pi_{\mathcal{S}} Y)), \Pi_{\mathcal{S}} Y\rangle Y\right), \tag{3.7}$$

where P_Y denotes the orthogonal projection onto $\mathcal{T}_Y \mathcal{M}_r$. By highlighting the role of the perturbation $E = \Pi_{\mathcal{S}} Y$, we can rewrite equation (3.7) in a more compact way

$$\dot{Y} = -P_Y(G_\varepsilon(E)) + \mathrm{Re}\langle P_Y G_\varepsilon(E), E\rangle Y,$$

since $P_Y Y = Y$ by definition. Figure 3.1 shows a visualization of the trajectory of the solution of equation (3.9) and of its associated structured perturbation.

The following result describes the main properties of the solutions of equation (3.7).

Lemma 3.1.5. *Let $Y(t)$ be a solution of equation (3.7) for $t \in [0, +\infty)$ with starting value $Y(0) = Y_0 \in \mathcal{M}_r$. Then $Y(t) \in \mathcal{M}_r$ for all t. Moreover, if $\|\Pi_{\mathcal{S}} Y_0\|_F = 1$, then $\|\Pi_{\mathcal{S}} Y(t)\|_F = 1$ for all t.*

Proof. From the right-hand side of (3.7) we notice that $\dot{Y} \in \mathcal{T}_Y\mathcal{M}_r$, which means that the whole trajectory $Y(t)$ belongs to the rank-r manifold. Finally the unit norm of $E = \Pi_{\mathcal{S}}Y$ is conserved along the solution of the ODE, since

$$\frac{\mathrm{d}}{\mathrm{d}t}\|E(t)\|_F^2 = \mathrm{Re}\langle \Pi_{\mathcal{S}}Y, \dot{Y}\rangle = -\mathrm{Re}\langle E, P_Y(G_\varepsilon(E))\rangle + \mathrm{Re}\langle P_Y(G_\varepsilon(E)), \Pi_{\mathcal{S}}Y\rangle\|E\|_F^2 = 0,$$

where we have used the properties of the projection $\Pi_{\mathcal{S}}$ and that $\|E\|_F^2 = \mathrm{Re}\langle E, Y\rangle$. □

Lemma 3.1.5 ensures that, if we consider a starting point $Y_0 \in \mathcal{M}_r$ such that $\Pi_{\mathcal{S}}Y_0$ has unit Frobenius norm, then the matrix path $Y(t)$ solution of the ODE (3.7) is low-rank and it is associated with an admissible perturbation path $E(t) \subseteq \mathcal{S}_1$. This new perturbation determined in general is not a solution of (3.4), but we show that it shares the same stationary points. Before doing this, we give a preliminary characterization of the stationary points of (3.7).

Theorem 3.1.6. *Let $Y(t) \subseteq \mathcal{M}_r$ be a solution of equation (3.4) passing through $Y_\star = Y(t_\star)$ at time $t_\star > 0$ and such that $\Pi_{\mathcal{S}}Y(t)$ has unit Frobenius norm. Assume that $P_{Y_\star}G_\varepsilon(E_\star) \neq 0$. Then the following facts are equivalent:*

1. $\left.\frac{\mathrm{d}}{\mathrm{d}t}F_\varepsilon(\Pi_{\mathcal{S}}Y(t))\right|_{t=t_\star} = 0$
2. *$Y_\star$ is a stationary point of (2.10)*
3. *$Y_\star$ is a non-zero real multiple of $P_{Y_\star}G_\varepsilon(E_\star)$*

Proof. It uses the same arguments of Theorem 2.3.4 and Theorem 2.3.6. □

The following result retraces Theorem 2.3.7 in order to show that there exists an explicit connection between the stationary points of the two ODEs (3.4) and (3.7).

Theorem 3.1.7. *Consider the two matrix ordinary differential equations*

$$\dot{E} = -\Pi_{\mathcal{S}}G_\varepsilon(E) + \mathrm{Re}\langle \Pi_{\mathcal{S}}G_\varepsilon(E), E\rangle E, \tag{3.8}$$

$$\dot{Y} = -P_Y G_\varepsilon(E) + \mathrm{Re}\langle P_Y G_\varepsilon(E), E\rangle Y. \tag{3.9}$$

1. *Let $E_\star \in \mathcal{S}_1$ of unit Frobenius norm be a stationary point of (3.8) and assume that $G_\varepsilon(E_\star)$ has rank r. Then $E_\star = \Pi_{\mathcal{S}}Y_\star$ for a certain matrix $Y_\star \in \mathcal{M}_r$ that is a stationary point of (3.9).*
2. *Conversely, let $Y_\star \in \mathcal{M}_r$ be a stationary point of (3.9) such that $E_\star = \Pi_{\mathcal{S}}Y_\star$ has unit Frobenius norm and $P_{Y_\star}G_\star \neq 0$, where $G_\star = G_\varepsilon(E_\star)$. Then $P_{Y_\star}G_\star = G_\star$, $Y_\star$ is a non-zero real multiple of $G_\star$ and $E_\star$ is a stationary point of (3.8).*

Proof. We start with the first statement. Theorem 3.1.4 states that the assumption is equivalent to say that there exists a non-zero $\mu \in \mathbb{R}$ such that $E_\star = \mu^{-1}\Pi_{\mathcal{S}}G_\star$, where $G_\star := G_\varepsilon(E_\star) \in \mathcal{M}_r$. Let us introduce $Y_\star = \mu^{-1}G_\star \in \mathcal{M}_r$ and we show that it is the stationary point of (3.9) sought. It is clear that $E_\star = \Pi_{\mathcal{S}}Y_\star$ and $P_{Y_\star}G_\star = \mu P_{Y_\star}Y_\star = \mu Y_\star = G_\star$, which implies

$$\mathrm{Re}\langle P_{Y_\star}G_\star, E_\star\rangle = \mathrm{Re}\langle G_\star, E_\star\rangle = \mu\,\mathrm{Re}\langle Y_\star, E_\star\rangle = \mu\|E_\star\|_F^2 = \mu$$

and similarly $\mathrm{Re}\langle \Pi_{\mathcal{S}}G_\star, E_\star\rangle = \mu$. Hence the left-hand side of (3.9) becomes

$$-P_{Y_\star}G_\star + \mathrm{Re}\langle P_{Y_\star}G_\star, E_\star\rangle Y_\star = -G_\star + \mu Y_\star = 0,$$

which means that $Y_\star$ is a stationary point of (3.9).

For the second statement we begin by showing that $Y_\star$ is a non-zero real multiple of $G_\star$. Since $Y_\star$ is a stationary point, Theorem 3.1.6 yields $Y_\star = \nu^{-1} P_{Y_\star} G_\star \neq 0$ for some $\nu \in \mathbb{R} \setminus \{0\}$, that is

$$G_\star = \nu Y_\star + W, \tag{3.10}$$

where $W \in \mathbb{C}^{n\times n}$ satisfies $P_{Y_\star} W = 0$. Let $Y_\star = U_\star S_\star V_\star^*$ where $U_\star, V_\star \in \mathbb{C}^{n\times r}$ has orthonormal columns and $S_\star \in \mathbb{C}^{r\times r}$ is invertible. Then

$$W = (I - U_\star U_\star^*) W (I - V_\star V_\star^*)$$

and equation (3.10) becomes

$$G_\star = \nu U_\star S_\star V_\star^* + (I - U_\star U_\star^*) W (I - V_\star V_\star^*).$$

By multiplying from the right by $U_\star$ we get

$$G_\star U_\star = \nu U_\star S_\star,$$

which means that $G_\star, Y_\star$ and $U_\star$ have the same range. Then Proposition D.0.4 implies

$$G_\star = G_\star U_\star U_\star^* = \nu U_\star S_\star V_\star^* = \nu Y_\star,$$

which shows that $Y_\star$ is a non-zero multiple of $G_\star$. Thus $P_{Y_\star} G_\star = G_\star$ and the properties of the projection imply

$$\langle P_{Y_\star} G_\star, E_\star\rangle = \langle G_\star, E_\star\rangle = \nu\langle Y_\star, E_\star\rangle = \nu \|E_\star\|_F^2 = \nu,$$

since $E_\star$ has Frobenius unit norm by assumption. Finally the left-hand side of (3.8) becomes

$$-\Pi_{\mathcal{S}} G_\star + \langle \Pi_{\mathcal{S}} G_\star, E_\star\rangle \Pi_{\mathcal{S}} Y_\star = -\nu \Pi_{\mathcal{S}} Y_\star + \langle G_\star, E_\star\rangle \Pi_{\mathcal{S}} Y_\star = 0,$$

which shows that $E_\star = \Pi_{\mathcal{S}} Y_\star$ is a stationary point of (3.8). □

Theorem 3.1.7 shows that there exists a one-to-one correspondence between the stationary points of (3.4) and (3.7), but this is not enough to guarantee that integrating equation (3.7) leads to them. Indeed it is not possible to prove a result, analogous to Theorem 3.1.3, for the monotonicity of the objective functional evaluated in the perturbation described by $\Pi_{\mathcal{S}} Y(t)$, because actually it is not true. Indeed there exist cases where the functional increases along the trajectory of the solution of equation (3.9) (see Figure 4.1).

The main motivation behind this issue is that the projections $\Pi_{\mathcal{S}}$ and P_Y do not commute. For instance, if $\mathcal{S} \subseteq \mathbb{C}^{2\times 2}$ is the subspace of real symmetric matrices, we can consider the rank-1 matrices

$$Y = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix}, \qquad G = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$$

and we observe that

$$\Pi_{\mathcal{S}}(P_Y G) = \Pi_{\mathcal{S}}\left(\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \right) = \Pi_{\mathcal{S}}\left(\begin{pmatrix} 1 & 2 \\ 0 & 4 \end{pmatrix} \right) = \begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix},$$

while

$$P_Y(\Pi_{\mathcal{S}}G) = \begin{pmatrix}1 & 2\\ 2 & 4\end{pmatrix} - \begin{pmatrix}0 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & 2\\ 2 & 4\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} = \begin{pmatrix}1 & 2\\ 0 & 4\end{pmatrix}.$$

Also for larger dimension it is usually false that $P_Y(\Pi_{\mathcal{S}}M)$ and $\Pi_{\mathcal{S}}(P_YM)$ are equal for all $M \in \mathbb{C}^{n\times n}$, since $\operatorname{rank}(P_Y(\Pi_{\mathcal{S}}(M))) \leq 2\operatorname{rank}(Y)$ by the definition of P_Y, while the latter matrix is generally full-rank. This shows that in general

$$\operatorname{Re}\langle \Pi_{\mathcal{S}}G, P_YM\rangle = \operatorname{Re}\langle \Pi_{\mathcal{S}}G, \Pi_{\mathcal{S}}P_YM\rangle \neq \operatorname{Re}\langle \Pi_{\mathcal{S}}G, P_Y\Pi_{\mathcal{S}}M\rangle$$

with $G = G_\varepsilon(E)$, $M \in \mathbb{C}^{n\times n}$ and hence, for the derivative formula in Lemma (3.1.1) with $\dot{E} = \Pi_{\mathcal{S}}\dot{Y}$, we have that

$$\frac{\mathrm{d}}{\mathrm{d}t}F_\varepsilon(E(t)) = \operatorname{Re}\langle G, \Pi_{\mathcal{S}}\dot{Y}\rangle =$$

$$= -\operatorname{Re}\langle \Pi_{\mathcal{S}}G, P_YG\rangle + \operatorname{Re}\langle P_YG, E\rangle \operatorname{Re}\langle G, E\rangle \neq -\|\Pi_{\mathcal{S}}P_YG\|^2 + (\operatorname{Re}\langle P_YG, E\rangle)^2,$$

which, if it were true, together with the Cauchy-Schwarz inequality, would have ensured the monotonicity of the objective functional.

However, the monotonicity of $F_\varepsilon(\Pi_{\mathcal{S}}Y(t))$ is related to the choice of the starting point for the integration of (3.9), which plays a crucial role. In the next paragraph we aim to overcome this problem by asking a weaker requirement about the monotonic decrease of the objective functional.

3.1.3 Local convergence to the low-rank stationary points

Unlike the ODE (3.4), equation (3.7) is not a gradient system, but somehow it is close to. In particular we show that, given a suitable starting point, the value of $F_\varepsilon(\Pi_{\mathcal{S}}Y(t))$ decreases exponentially to 0 as $t \to +\infty$. Before doing this, we need a preliminary lemma.

Lemma 3.1.8. *Assume that the gradient $G_\varepsilon(E)$ in the ODE* (3.7) *has rank r and it consists of a linear combination of eigenvectors corresponding to simple eigenvalues of $\mathscr{L}(A+\varepsilon E)$. Let $Y_\star \in \mathcal{M}_r$ be a stationary point of equation* (3.7) *such that $E_\star = \Pi_{\mathcal{S}}Y_\star \in \mathcal{S}_1$. Then there exists $\delta_\star > 0$ such that, for all $\delta \in (0, \delta_\star]$ and all $\hat{Y} \in \mathcal{M}_r$ with $\|Y_\star - \hat{Y}\|_F \leq \delta$ and $\Pi_{\mathcal{S}}\hat{Y} \in \mathcal{S}_1$, we have*

$$\|P_{\hat{Y}}G_\varepsilon(\Pi_{\mathcal{S}}\hat{Y}) - G_\varepsilon(\Pi_{\mathcal{S}}\hat{Y})\| \leq C\delta^2,$$

where C is a constant independent of δ.

Proof. The assumptions on the gradient imply that it is possible to write

$$G_\star := G_\varepsilon(E_\star) = B_\star\Lambda_\star C_\star^*,$$

where $B_\star, C_\star \in \mathbb{C}^{n\times r}$ have orthonormal columns containing the unit left and right eigenvectors, respectively, of $\mathscr{L}(A+\varepsilon E_\star)$ and $\Lambda_\star \in \mathbb{C}^{r\times r}$ contains the associated eigenvalues on the diagonal. Let $Y_\star = U_\star S_\star V_\star^*$ be the SVD-like (see (2.12)) of a stationary point of (3.7) that fulfils the assumptions. Then, as shown in Theorem 3.1.7, there exists $\nu \in \mathbb{R}\setminus\{0\}$ such that

$$Y_\star = U_\star S_\star V_\star^* = \nu^{-1}G_\star = \nu^{-1}B_\star\Lambda_\star C_\star^*,$$

meaning that $U_\star = \nu^{-1} B_\star \Lambda_\star C_\star^* V_\star S_\star^{-1}$ and $V_\star^* = \nu^{-1} S_\star^{-1} U_\star^* B_\star \Lambda_\star C_\star^*$. Thus

$$Y_\star = B_\star \left(\nu^{-2} \Lambda_\star C_\star^* V_\star S_\star^{-1} U_\star^* B_\star \Lambda_\star \right) C_\star^*$$

and hence it is not restrictive to assume for the SVD-like of $Y_\star$ that $U_\star = B_\star$ and $V_\star = C_\star$. We introduce the matrix paths

$$\widetilde{Y}(\tau) = \widetilde{U}(\tau)\widetilde{S}(\tau)\widetilde{V}(\tau)^*, \qquad \widetilde{G}(\tau) = \widetilde{B}(\tau)\widetilde{\Lambda}(\tau)\widetilde{C}(\tau)^*, \qquad \tau \in [0, \delta]$$

which are guaranteed to be differentiable by the assumption on the simple eigenvalues, and such that

$$Y_\star = \widetilde{Y}(0), \qquad G_\star = \widetilde{G}(0), \qquad \hat{Y} = \widetilde{Y}(\delta), \qquad G_\varepsilon(\hat{E}) = \widetilde{G}(\delta).$$

Since any matrix $\hat{Y}$ that satisfies the hypothesis can be written as $\hat{Y} = \widetilde{Y}(\delta)$, for instance

$$\widetilde{Y}(\tau) = \hat{Y} + \frac{\tau - \delta}{\delta} \left(\hat{Y} - Y_\star \right),$$

it is enough to study these straight paths in order to conclude.

We will denote, for brevity, by U, V, B, C, Λ and later $\dot{U}, \dot{V}, \dot{B}, \dot{C}, \dot{\Lambda}$ the associated function (equipped with the $\sim$) evaluated at $\tau = 0$. The derivatives of $\widetilde{B}$ and of the other matrix functions, are well defined in a right-neighbourhood of $\tau = 0$. Given the left and right unit eigenvectors $\tilde{x}(\tau)$ and $\tilde{y}(\tau)$ of $\mathscr{L}(A + \varepsilon E(\tau))$ (where $E(\tau) = \Pi_{\mathcal{S}} \widetilde{Y}(\tau)$) associated to the eigenvalue $\lambda(\tau)$, the eigenvectors' derivative formulas are (see [34, 58] for more details)

$$\frac{1}{\varepsilon} \frac{\mathrm{d}}{\mathrm{d}\tau} \tilde{x}(\tau)^* = -x(\tau)^* \mathscr{L}\left(\dot{E}(\tau) \right) Z(\tau) + \mathrm{Re}\left(x(\tau)^* \mathscr{L}\left(\dot{E}(\tau) \right) Z(\tau) x(\tau) \right) x(\tau)^*,$$

$$\frac{1}{\varepsilon} \frac{\mathrm{d}}{\mathrm{d}\tau} \tilde{y}(\tau)^* = -Z(\tau) \mathscr{L}\left(\dot{E}(\tau) \right) y(\tau) + \mathrm{Re}\left(y(\tau)^* Z(\tau) \mathscr{L}\left(\dot{E}(\tau) \right) y(\tau) \right) y(\tau),$$

where

$$Z(\tau) = \left(\mathscr{L}\left(W + \varepsilon E\left(\widetilde{Y}(\tau) \right) \right) - \lambda I \right)^\sharp$$

is the group inverse of the perturbed matrix, that is bounded since the eigenvalues are simple. This shows that the first derivative of $\tilde{x}(\tau)$ and $\tilde{y}(\tau)$ and hence also $\dot{U}, \dot{V}, \dot{B}, \dot{C}$ and $\dot{\Lambda}$ are well defined, since their columns are exactly the derivatives of the eigenvectors, and this allows to expand until the first order the matrices $\widetilde{G}$ and $P_{\tilde{Y}} \widetilde{G}$ for $0 \le \tau \le \delta$. Recalling that $U(0) = U_\star = B_\star = B(0)$, $V(0) = V_\star = C_\star = C(0)$ and that $U^* U = V^* V = I_r$ yields

$$P_{\tilde{Y}(\tau)} \widetilde{G}(\tau) = \widetilde{U}(\tau)\widetilde{U}(\tau)^* \widetilde{G}(\tau) + \widetilde{G}(\tau)\widetilde{V}(\tau)\widetilde{V}(\tau)^* - \widetilde{U}(\tau)\widetilde{U}(\tau)^* \widetilde{G}(\tau)\widetilde{V}(\tau)\widetilde{V}(\tau)^* =$$

$$= B\Lambda C^* + \tau \left(\dot{U}U^* B\Lambda C^* + U\dot{U}^* B\Lambda C^* + UU^* \dot{B}\Lambda C^* + UU^* B\dot{\Lambda} C^* + UU^* B\Lambda \dot{C}^* \right) +$$

$$+\tau \left(\dot{B}\Lambda C^* VV^* + B\dot{\Lambda} C^* VV^* + B\Lambda \dot{C}^* VV^* + B\Lambda C^* \dot{V} V^* + B\Lambda C^* V\dot{V}^* \right) +$$

$$+\tau \left(-\dot{U}U^* B\Lambda C^* VV^* - U\dot{U}^* B\Lambda C^* VV^* - UU^* \dot{B}\Lambda C^* VV^* - UU^* B\dot{\Lambda} C^* VV^* \right) +$$

$$+\tau \left(-UU^* B\Lambda \dot{C}^* VV^* - UU^* B\Lambda C^* \dot{V} V^* - UU^* B\Lambda C^* V\dot{V}^* \right) + \mathcal{O}(\tau^2) =$$

$$= B\Lambda C^* + \tau \left(\dot{U}\Lambda V^* + U\dot{U}^* U\Lambda V^* + UU^* \dot{B}\Lambda V^* + U\dot{\Lambda} V^* + U\Lambda \dot{C}^* \right) +$$

$$+\tau\left(\dot{B}\Lambda V^* + U\dot{\Lambda}V^* + U\Lambda\dot{C}^*VV^* + U\Lambda V^*\dot{V}V^* + U\Lambda\dot{V}^*\right) +$$
$$+\tau\left(-\dot{U}\Lambda V^* - U\dot{U}^*U\Lambda V^* - UU^*\dot{B}\Lambda V^* - U\dot{\Lambda}V^*\right) +$$
$$+\tau\left(-U\Lambda\dot{C}^*VV^* - U\Lambda V^*\dot{V}V^* - U\Lambda\dot{V}^*\right) + \mathcal{O}(\tau^2) =$$
$$= B\Lambda C^* + \tau\left(\dot{B}\Lambda V^* + U\dot{\Lambda}V^* + U\Lambda\dot{C}^*\right) + \mathcal{O}(\tau^2),$$

while

$$\widetilde{G}(\tau) = B\Lambda C^* + \tau(\dot{B}\Lambda C^* + B\dot{\Lambda}C^* + B\Lambda\dot{C}^*) + \mathcal{O}(\tau^2),$$

which proves the claim. □

Lemma 3.1.8 is crucial for providing a local convergence result. It states that, in a neighbourhood of width δ of a stationary point $Y_\star$, the gradient and its projection onto $\mathcal{T}_{Y_\star}\mathcal{M}_r$ coincide up to quadratic terms in δ, meaning that the two matrices are very close. For stating the main theorem of the section we need the following definition.

Definition 3.1.9. *A strict local minimum of a smooth function $F : \mathcal{S}_1 \to [0, +\infty)$ is a matrix E such that the Hessian matrix $H(E)$ of F defines a positive definite bilinear form when restricted to $\mathcal{T}_E\mathcal{S}_1$, that is there exists $\alpha > 0$ such that, for all $Z \in \mathcal{T}_E\mathcal{S}_1$,*

$$\langle H(E)Z, Z\rangle \geq \alpha\|Z\|_F^2.$$

The next result shows the local convergence of equation (3.7) towards a stationary point $Y_\star$ under two main assumptions:

- $\Pi_\mathcal{S}$ is smooth and bijective on the image when restricted to $\mathcal{M}_r$,
- the matrix $\Pi_\mathcal{S} Y_\star$ is a strict local minimum of F_ε.

After the proof of the theorem we comment on these assumptions.

Theorem 3.1.10. *Let $Y_\star \in \mathcal{M}_r$ be a stationary point of the projected differential equation* (3.7) *such that $E_\star = \Pi_\mathcal{S} Y_\star \in \mathcal{S}_1$ and $P_{Y_\star}G_\varepsilon(E_\star) \neq 0$. Suppose that $E_\star$ is a strict local minimum of the functional F_ε on $\mathcal{S}_1$ and assume that*

$$\Pi_\mathcal{S}|_{\mathcal{M}_r} : \mathcal{M}_r \to \Pi_\mathcal{S}(\mathcal{M}_r) \subseteq \mathcal{S}$$

is a diffeomorphism. Then, for an initial datum $Y(0)$ sufficiently close to $Y_\star$, the solution $Y(t)$ of (3.7) *converges to $Y_\star$ exponentially as $t \to +\infty$. Moreover $F_\varepsilon(\Pi_\mathcal{S} Y(t))$ decreases monotonically with t and converges exponentially to the local minimum value $F(E_\star)$ as $t \to +\infty$.*

Proof. By applying $\Pi_\mathcal{S}$ to both sides of (3.7) and by recalling the properties of the projections we get the equivalent form

$$\dot{E} = \Pi_\mathcal{S}\left(-P_Y(G) + \langle P_Y(G), E\rangle Y\right) = -\Pi_\mathcal{S} P_Y(G) + \langle \Pi_\mathcal{S} P_Y(G), E\rangle E,$$

where $E(t) = \Pi_\mathcal{S} Y(t) \in \mathcal{S}_1$ and $G = G_\varepsilon(E)$ for short. By means of Lemma 3.1.8, this equation can be rewritten as a perturbation of the original gradient system of E, that is

$$\dot{E} = -\Pi_\mathcal{S}(G_\varepsilon(E)) + \langle \Pi_\mathcal{S}(G_\varepsilon(E)), E\rangle E + D := -\widehat{\Pi}_E^\mathcal{S}(G_\varepsilon(E)) + D,$$

where $\|D(t)\| = \mathcal{O}(\|Y(t) - Y_\star\|_F^2)$ and the orthogonal projection of $B \in \mathbb{R}^{n\times n}$ onto the tangent space $\mathcal{T}_E\mathcal{S}_1$ is defined as (see Proposition B.0.9)

$$\widehat{\Pi}_E^{\mathcal{S}}(B) = \Pi_{\mathcal{S}}(B) - \langle \Pi_{\mathcal{S}}(B), E\rangle E.$$

For $\delta := \|E(t) - E_\star\|_F$, which is supposed to be sufficiently small, the assumptions yield

$$\|D(t)\| = \mathcal{O}(\|Y(t) - Y_\star\|_F^2) = \mathcal{O}(\|E(t) - E_\star\|_F^2) = \mathcal{O}(\delta^2)$$

and

$$E - E_\star = \widehat{\Pi}_{E_\star}^{\mathcal{S}}(E - E_\star) + \mathcal{O}(\delta^2).$$

Since $E_\star$ is a strict local minimum, by definition (see 3.1.9) there exists $\alpha > 0$ such that the Hessian matrix $H_\varepsilon(E_\star)$ of F_ε at $E_\star$ satisfies

$$\langle Z, H_\varepsilon(E_\star)Z\rangle \geq \alpha Z, \qquad \forall Z \in \mathcal{T}_{E_\star}\mathcal{S}_1.$$

By the definition of $\widehat{\Pi}_E^{\mathcal{S}}$ it also follows that

$$\widehat{\Pi}_{E_\star}^{\mathcal{S}}(G_\varepsilon(E)) = \widehat{\Pi}_{E_\star}^{\mathcal{S}}(G_\varepsilon(E)) - \widehat{\Pi}_{E_\star}^{\mathcal{S}}(G_\varepsilon(E_\star)) = \widehat{\Pi}_{E_\star}^{\mathcal{S}} H_\varepsilon(E_\star)\widehat{\Pi}_{E_\star}^{\mathcal{S}}(E - E_\star) + \mathcal{O}(\delta^2),$$

since Theorem 3.1.4 implies $\widehat{\Pi}_{E_\star}^{\mathcal{S}}(G_\varepsilon(E_\star)) = \Pi_{\mathcal{S}} G_\varepsilon(E_\star) - \langle \Pi_{\mathcal{S}} G_\varepsilon(E_\star), E_\star\rangle E_\star = 0$. Thus

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|E(t) - E_\star\|_F^2 &= \langle E - E_\star, -\widehat{\Pi}_{E_\star}^{\mathcal{S}}(G_\varepsilon(E)) + D\rangle = \\ = \langle \widehat{\Pi}_{E_\star}^{\mathcal{S}}(E - E_\star) + \mathcal{O}(\delta^2), &-\widehat{\Pi}_{E_\star}^{\mathcal{S}} H_\varepsilon(E_\star)\widehat{\Pi}_{E_\star}^{\mathcal{S}}(E - E_\star) + \mathcal{O}(\delta^2)\rangle = \\ &= \langle \widehat{\Pi}_{E_\star}^{\mathcal{S}}(E - E_\star), -H_\varepsilon(E_\star)\widehat{\Pi}_{E_\star}^{\mathcal{S}}(E - E_\star)\rangle + \mathcal{O}(\delta^3) \leq \\ &\leq -\alpha\|\widehat{\Pi}_{E_\star}^{\mathcal{S}}(E - E_\star)\|_F^2 + \mathcal{O}(\delta^3) \leq -\frac{\alpha}{2}\|E - E_\star\|_F^2,\end{aligned}$$

where we have used that $\widehat{\Pi}_{E_\star}^{\mathcal{S}} H_\varepsilon(E_\star)\widehat{\Pi}_{E_\star}^{\mathcal{S}}(E - E_\star) = \mathcal{O}(\delta)$, Proposition B.0.10 for the projection $\widehat{\Pi}_{E_\star}^{\mathcal{S}}$ and that δ is sufficiently small. This proves the exponential convergence of $E(t)$ towards $E_\star$ as $t \to +\infty$ and a similar approach shows that $F(E(t))$ converges monotonically towards $F(E_\star)$ as $t \to +\infty$:

$$\begin{aligned}\frac{1}{\varepsilon}\frac{\mathrm{d}}{\mathrm{d}t}F_\varepsilon(E(t)) &= \mathrm{Re}\langle G_\varepsilon(E), \dot{E}\rangle = \mathrm{Re}\langle \widehat{\Pi}_{E_\star}^{\mathcal{S}} G_\varepsilon(E), \dot{E}\rangle = \\ = \mathrm{Re}\langle \widehat{\Pi}_{E_\star}^{\mathcal{S}} G_\varepsilon(E), -\widehat{\Pi}_{E_\star}^{\mathcal{S}} G_\varepsilon(E) + D\rangle &= \|\widehat{\Pi}_{E_\star}^{\mathcal{S}} H_\varepsilon(E_\star)\widehat{\Pi}_{E_\star}^{\mathcal{S}}(E - E_\star)\|_F^2 + \mathcal{O}(\delta^3) \leq \\ &\leq -\alpha^2\|\widehat{\Pi}_{E_\star}^{\mathcal{S}}(E - E_\star)\|_F^2 + \mathcal{O}(\delta^3) \leq -\frac{\alpha^2}{2}\|E - E_\star\|_F^2.\end{aligned}$$

□

The first assumption on the regularity of $\Pi_{\mathcal{S}}$ is usually fulfilled when the dimension of $\mathcal{S}$ is large enough. If this is not the case, then it is also less attractive to consider equation (3.7) instead of (3.4), since there would not be a gain in computational terms, meaning that the assumption on the fact that $\Pi_{\mathcal{S}}$ is a local diffeomorphism is not restrictive. The second assumption is mainly technical and in practice we experienced that it is verified. We give more specific details about this facts in Chapter 4.

3.2 Structured outer iteration

As done for the *outer iteration* in the unstructured case, it is possible to follow the same approach for the *structured outer iteration*. Let $E_\star(\varepsilon)$ be the minimizer computed by the *structured inner iteration*, that is a stationary point of (3.4) or the projection onto $\mathcal{S}$ of a stationary point of (3.7), and define

$$\varphi(\varepsilon) := F_\varepsilon(E_\star(\varepsilon)).$$

We are interested in the smallest zero $\varepsilon_\star$ of $\varphi - a_\star$, that we compute by means of the Newton-bisection method described in Algorithm 1. The following lemma provides the formula for the derivative of φ in $\varepsilon < \varepsilon_\star$ for the structured case.

Lemma 3.2.1. *For* $0 \leq \varepsilon < \varepsilon_\star$ *we have*

$$\varphi'(\varepsilon) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} F_\varepsilon(E_\star(\varepsilon)) = \langle \Pi_{\mathcal{S}} G_\varepsilon(E_\star(\varepsilon)), E_\star(\varepsilon)\rangle = -\|\Pi_{\mathcal{S}} G_\varepsilon(E_\star(\varepsilon))\|_F \leq 0.$$

Proof. With the same steps of Lemma 2.4.1 we have

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon} F_\varepsilon(E_\star(\varepsilon)) = \mathrm{Re}\langle G_\varepsilon(E_\star(\varepsilon)), E_\star(\varepsilon) + \varepsilon E_\star'(\varepsilon)\rangle = \mathrm{Re}\langle \Pi_{\mathcal{S}} G_\varepsilon(E_\star(\varepsilon)), E_\star(\varepsilon) + \varepsilon E_\star'(\varepsilon)\rangle,$$

where $E_\star'(\varepsilon)$ is the derivative with respect to ε of $E_\star(\varepsilon)$. Theorem 2.3.6 yields that the unit norm stationary point $E_\star(\varepsilon)$ of (3.4) is a real multiple of $\Pi_{\mathcal{S}} G_\varepsilon(E_\star(\varepsilon))$ and, since the objective functional F_ε is monotonically decreasing, then $E_\star$ is a negative multiple of $\Pi_{\mathcal{S}} G_\varepsilon(E_\star(\varepsilon))$. Thus $\Pi_{\mathcal{S}} G_\varepsilon(E_\star(\varepsilon)) = -\|\Pi_{\mathcal{S}} G_\varepsilon(E_\star(\varepsilon))\|_F \; E_\star(\varepsilon)$ and, since $\|E_\star(\varepsilon)\|_F = 1$ for all ε, we have

$$\mathrm{Re}\langle \Pi_{\mathcal{S}} G_\varepsilon(E_\star(\varepsilon)), E_\star'(\varepsilon)\rangle = -\frac{\|\Pi_{\mathcal{S}} G_\varepsilon(E_\star(\varepsilon))\|_F}{2} \frac{\mathrm{d}}{\mathrm{d}\varepsilon} \|E_\star(\varepsilon)\|_F^2 = 0.$$

□

Hence, the algorithm for performing the *structured outer iteration* is the same as Algorithm 1, but in steps 1 and 16 the *structured inner iteration* is solved instead of its unstructured version.

Chapter 4

Rank-1 structured eigenvalue optimization

In this chapter, that is mainly based on the results from [34], we show how the structured two-level method introduced in Chapter 2 and Chapter 3 can be exploited in a matrix stability framework.

We propose and discuss a new approach for solving eigenvalue optimization problems for large structured matrices, where it is required to control a single target eigenvalue of a given matrix $A \in \mathbb{C}^{n\times n}$. The class of optimization problems considered is related to compute structured pseudospectra and their extremal points, but it is also suitable to deal with structured matrix nearness problems such as computing the distance to instability or to singularity under structured perturbations. The structure consists of a linear subspace of the matrix set and generally it denotes a property of the original matrix that we want to preserve after perturbing it. In particular, we focus on the practically important cases of large matrices with a given sparsity pattern and on perturbation matrices with given range and co-range. It is known that analogous eigenvalue optimization for unstructured complex matrices favorably works with rank-1 matrices. The novelty presented in this chapter is that structured eigenvalue optimization can still be performed with rank-1 matrices, which yields a significant reduction of storage and in some cases of the computational cost. Optimizers are shown to be rank-1 matrices orthogonally projected onto the given structure and this fact is used in the numerical algorithms designed to solve the problem.

4.1 Introduction

We describe an approach to solve structured eigenvalue optimization problems that uses constrained gradient flows and the underlying rank-1 property of the optimizers. We illustrate basic techniques on a class of problems that arise in computing structured pseudospectra or their extremal points and appear as the essential algorithmic building block in structured matrix nearness problems. For example, we determine the largest possible spectral abscissa or radius of a given matrix under perturbations of a prescribed norm that preserve its structure, or - in other words - the structured pseudospectral abscissa or radius. This is an important subtask in the computation of structured stability radii (or structured distance to instability in another terminology). In the literature these quantities are extensively studied with the purpose of analyzing stability properties and robustness of linear dynamical systems (see, e.g., [47]). Similarly, if one is interested in the distance of a matrix to singularity, the unstructured distance is the smallest singular value. However, if the matrix is structured, having a small singular value does not imply the existence of a small structured perturbation that makes it singular, and the structured distance to singularity is not readily obtained.

The structures considered in this chapter are general complex- or real-linear structures, that is, the perturbation matrices are restricted to lie in a structure space $\mathcal{S}$, which can be an arbitrary linear subspace of $\mathbb{C}^{n\times n}$ or $\mathbb{R}^{n\times n}$. We will put the focus on two very different classes of major interest in applications:

(i) perturbation matrices with a given sparsity pattern,

(ii) perturbation matrices with given range and co-range.

Instead of a direct discrete approach to solve the optimization problems, we present a continuous approach using structure- and norm-constrained gradient flows, which reveals the underlying rank-1 property of optimizers, on which we build our discrete optimization method. The rank-1 property is well-known for unstructured problems (see e.g. [67]) and has been exploited for developing suitable algorithms (see e.g. [29, 36, 52]). The rank-1 differential equation is finally fully discretized, using an appropriate time discretization (here chosen beyond mere gradient descent) and an adaptive, line search-type stepsize selection. We mention that there are several situations previously addressed in the literature where considering a time-continuous approach provides new insights, e.g. [1, 7, 16, 42, 67] and references therein. This list is far from exhaustive.

In previous works, structured eigenvalue optimization problems were addressed for some specific structures. For example when the matrices are required to be real (the unstructured problem would consider them as complex), it has been proved that the optimizers have a rank-2 structure [62] and indeed are obtained as real parts of an underlying rank-1 matrix [30]. Similarly, Hamiltonian eigenvalue optimization has been studied in detail in [57] and [2], in the setting of robust passivity analysis of linear control systems, where eigenvalues of Hamiltonian matrices have to be bounded away from the imaginary axis. In that case it is possible to show that for a real Hamiltonian matrix, extremal perturbations have rank 4 [28]. However, when considering for example a sparse matrix, the low-rank property of optimizers seems to be irremediably lost. *It is a basic goal of this chapter to uncover the underlying rank-*1 *property and to show how it can be used in algorithms for structured eigenvalue optimization.*

The chapter is organized as follows. In Section 4.2 we set up the framework and present our approach, which is based on a structure- and norm-constrained gradient system. We show that optimizers are orthogonal projections of rank-1 matrices onto the given structure. We discuss the possibilities and difficulties of using a gradient system for structure-projected rank-1 matrices. This works well for the case (ii) of prescribed range and co-range, but it is not feasible for the case (i) of a prescribed sparsity pattern. In Section 4.3 we introduce instead a differential equation on the manifold of rank-1 matrices of unit Frobenius norm, for which the stationary points are shown to be in a bijective correspondence with the stationary points of the structure- and norm-constrained gradient system. In Section 4.4 we prove local convergence to strict minima under an assumption that appears to be generically satisfied in case (i) of sparse matrices, but that is not satisifed in case (ii) of perturbation matrices with prescribed range and co-range. A basic observation, valid for all cases, is that near a local minimizer, the rank-1 tangent projection is very close to the identity map, and so the computationally favorable rank-1 projected system behaves locally like the gradient system. In Section 4.5 we discretize the rank-1 differential equation by a splitting method. This leads us to a fully discrete algorithm that updates rank-1 matrices in every step. Then, in Section 4.6 we describe a two-level approach to compute the structured stability radius (or structured distance to instability), used to characterize robustness of spectral stability properties. This is an important use of the considered class of eigenvalue optimization problems for solving structured matrix nearness problems. The structured distance to singularity is computed in an analogous way. In Section 4.7 we present some numerical examples showing that the rank-1 system is well-suited for the efficient computation of optimizers. Finally, in Section 4.8 we show how the alternative approach of using the gradient system for structure-projected rank-1 matrices can be used for the case (ii) of prescribed range and co-range.

4.2 Structured constrained gradient flows

In this section we formulate and discuss a class of eigenvalue optimization problems that are related to structured pseudospectra. We derive and study structure- and norm-constrained gradient systems and their stationary points, which turn out to be structure-projected rank-1 matrices.

4.2.1 Problem formulation and motivation

For a matrix $A \in \mathbb{C}^{n\times n}$, let $\lambda(A) \in \mathbb{C}$ be a target eigenvalue of A, for example:

- eigenvalue of minimal or maximal real part;
- eigenvalue of minimal or maximal modulus;
- closest eigenvalue to a given set in the complex plane.

We note that here the eigenvector associated with the target eigenvalue may not depend continuously on the matrix A when several eigenvalues are simultaneously extremal, but it depends continuously on A when the extremal eigenvalue is unique. Let $\mathcal{S}$ be a subspace of the vector space of complex or real $n \times n$ matrices, e.g., a space of matrices with a prescribed sparsity pattern, or matrices with given range and co-range. We let

$$f : \mathbb{C}^2 \to \mathbb{C} \quad \text{with} \quad f\left(z, \overline{z}\right) = f\left(\overline{z}, z\right) \in \mathbb{R} \quad \text{for all } \ z \in \mathbb{C} \tag{4.1}$$

be a given smooth function that will be minimized over target eigenvalues $\lambda(A + \Delta)$ for structured perturbations $\Delta \in \mathcal{S}$ to a given matrix A. While our theory applies to general functions f with (4.1), in our examples we consider specific cases where f or $-f$ evaluated at $(z, \overline{z})$ equals

$$\mathrm{Re}(z) = \frac{z + \overline{z}}{2} \quad \text{or} \quad |z|^2 = z\overline{z}.$$

As it is dicussed in more detail in Section 4.6, the real part function is used in studying the distance to instability (or stability radius) of a Hurwitz matrix, that is with all eigenvalues in the left complex half-plane. The interest is in computing the nearest matrix $A + \Delta$ to A for which the rightmost eigenvalue is on the imaginary axis. Here, the perturbation Δ will be constrained to be in the structure space $\mathcal{S}$, and “nearest” will refer to the Frobenius norm $\|\Delta\|_F$. Similarly, the squared modulus function is used when A is a Schur matrix, that is with all eigenvalues in the unit disk, to compute the nearest matrix $A + \Delta$ to A for which the eigenvalue with largest absolute value is on the unit circle. The squared modulus function is also used to compute the structured distance to singularity of an invertible matrix.

We consider the following *structured eigenvalue optimization problem*, which turns out to be a particular case of problem (3.1): for a given perturbation size $\varepsilon > 0$, find

$$\underset{\Delta\in\mathcal{S},\ \|\Delta\|_F=\varepsilon}{\arg\min} \quad f\left(\lambda\left(A+\Delta\right), \overline{\lambda}\left(A+\Delta\right)\right), \tag{4.2}$$

where $\|\Delta\|_F$ is the Frobenius norm of the structured matrix $\Delta \in \mathcal{S}$ and $\lambda(A+\Delta)$ is the considered target eigenvalue of the perturbed matrix $A+\Delta$. The arg max case is treated analogously, replacing f by $-f$. This problem arises in computing extremal points of the *structured ε-pseudospectrum*

$$\Lambda_\varepsilon^{\mathcal{S}}(A) = \{\lambda \in \mathbb{C} \,:\, \lambda \text{ is an eigenvalue of } A+\Delta \text{ for some } \Delta \in \mathcal{S} \text{ with } \|\Delta\|_F \le \varepsilon\}.$$

For $f(z,\overline{z}) = -\operatorname{Re}(z)$, (4.2) yields the structured pseudospectral abscissa

$$\alpha_\varepsilon^{\mathcal{S}}(A) = \max\left\{\operatorname{Re}(z) \,:\, z \in \Lambda_\varepsilon^{\mathcal{S}}(A)\right\}$$

and for $f(z,\overline{z}) = -|z|^2$ it yields the structured pseudospectral radius

$$\rho_\varepsilon^{\mathcal{S}}(A) = \max\left\{|z| \,:\, z \in \Lambda_\varepsilon^{\mathcal{S}}(A)\right\}.$$

The structured distance to instability is then obtained by finding the smallest $\varepsilon > 0$ such that $\alpha_\varepsilon^{\mathcal{S}}(A) = 0$ (for a Hurwitz matrix A) or $\rho_\varepsilon^{\mathcal{S}}(A) = 1$ (for a Schur matrix A).

In the following, as done in Chapter 2 and Chapter 3, we write

$$\Delta = \varepsilon E \quad \text{with } \|E\|_F = 1 \qquad \text{and} \qquad F_\varepsilon(E) = f\left(\lambda\left(A+\varepsilon E\right), \overline{\lambda}\left(A+\varepsilon E\right)\right),$$

so that Problem (4.2) is equivalent to the problem of finding

$$\underset{E\in\mathcal{S},\ \|E\|_F=1}{\arg\min} \quad F_\varepsilon(E). \tag{4.3}$$

Problem (4.2) and Problem (4.3) are nonconvex, nonsmooth optimization problems and, as far as we know, generally there is no analytic formula for their solution.

4.2.2 Minimizing the objective functional

In order to deal with problem (4.3), we use the projection onto the subspace introduced in Chapter 3 and we briefly recall its main features. Let $\Pi_{\mathcal{S}}$ be the orthogonal projection (with respect to the Frobenius inner product) onto $\mathcal{S}$ as defined in Definition B.0.1: for every $Z \in \mathbb{C}^{n\times n}$,

$$\Pi_{\mathcal{S}} Z \in \mathcal{S} \quad \text{and} \quad \operatorname{Re}\langle \Pi_{\mathcal{S}} Z, W\rangle = \operatorname{Re}\langle Z, W\rangle \qquad \forall\, W \in \mathcal{S}.$$

For a complex-linear subspace $\mathcal{S}$, taking the real part of the complex inner product can be omitted (because with $W \in \mathcal{S}$, then also $\mathrm{i}W \in \mathcal{S}$), but taking the real part is needed for real-linear subspaces. Note that for $\mathcal{S} = \mathbb{R}^{n\times n}$, we then have $\Pi_{\mathcal{S}} Z = \operatorname{Re}(Z)$ for all $Z \in \mathbb{C}^{n\times n}$, while Proposition B.0.4 and Proposition B.0.5 provide, respectively, the explicit formula for the orthogonal projection onto a given sparsity pattern and onto the set of prescribed range and co-range matrices.

As done in Chapter 2 and Chapter 3, we introduce a differentiable matrix path $E(t)$ of unit Frobenius norm matrices. To get the gradient of the functional $F_\varepsilon(E(t))$, we need the derivative of the target eigenvalue $\lambda(A+\varepsilon E(t))$ for t in some interval $\mathcal{I}$, for instance $\mathcal{I} = [0, +\infty)$. In the case of a simple eigenvalue, which is the situation we will consider in the following, this derivative is obtained from the following well-known result (see e.g. or [50], [23, Theorem 1] or [48]).

Lemma 4.2.1 (Derivative of simple eigenvalues). *Consider a continuously differentiable path of square complex matrices $M(t)$ for t in an interval $\mathcal{I}$. For $t \in \mathcal{I}$, let $\lambda(t)$ be a continuous path of simple eigenvalues of $M(t)$ and let $x(t)$ and $y(t)$ be left and* *right eigenvectors, respectively, of $M(t)$ associated with the eigenvalue $\lambda(t)$. Then, $x(t)^*y(t) \neq 0$ for all $t \in \mathcal{I}$ and λ is continuously differentiable on $\mathcal{I}$ with*

$$\dot{\lambda} = \frac{x^* \dot{M} y}{x^* y}, \tag{4.4}$$

where the $\cdot$ indicates (entrywise) differentiation with respect to t.

Remark 4.2.2. *We mention some situations where the assumption of a smoothly evolving simple eigenvalue is violated. As such situations are either non-generic or can happen generically only at isolated times t, they do not affect the computation after discretization of the differential equation.*

- *Along a trajectory $E(t)$, the target eigenvalue $\lambda(t) = \lambda(A + \varepsilon E(t))$ may become discontinuous. For example, in the case of the eigenvalue of largest real part, a different branch of eigenvalues may get to have the largest real part. In such a case of discontinuity, the differential equation is further solved, with descent of the largest real part until finally a stationary point is approximately reached.*
- *A multiple eigenvalue $\lambda(t)$ may occur at some finite t because of a coalescence of eigenvalues. Even if some continuous trajectory runs into a coalescence, this is non-generic to happen after discretization of the differential equation, and so the computation will not be affected.*
- *A multiple eigenvalue may appear in a stationary point, in the limit $t \to +\infty$. The computation will stop before, and items 1.-3. in Theorem 4.2.8 will then be satisfied approximately.*

Although the situations above do not affect the time-stepping of the gradient system, close-to-multiple eigenvalues do impair the accuracy of the computed left and right eigenvectors that appear in the gradient.

Hence, assuming that the target eigenvalue is simple appears quite natural in this context. A further intuition that motivates this fact is that the set of matrices with multiple eigenvalues has zero measure in $\mathbb{C}^{n\times n}$ (see e.g. [4] for more details about this fact), even though this does not guarantee that matrices with large Jordan blocks do not occur with probability zero in the algorithm's dynamics. In addition to this theoretical motivations, we have never experienced in practice the issue of a multiple target eigenvalue. Thus, we will always assume the setting where equation (4.4) *holds.*

Since Theorem 4.2.1 ensures $x(t)^*y(t) \neq 0$, for all t we can apply the normalization

$$\|x(t)\| = 1, \quad \|y(t)\| = 1, \quad x(t)^*y(t) \text{ is real and positive.} \tag{4.5}$$

In this chapter we always assume that normalization (4.5) holds. We observe that a pair of left and right eigenvectors x and y fulfilling this property may be replaced by μx and μy for any complex μ of modulus 1 without changing the property (4.5). With this normalization it is always possible to define the eigenvalue condition number, that is

$$\kappa(t) = \frac{1}{x(t)^*y(t)} > 0,$$

which gives the idea of how close a simple eigenvalue is to become multiple.

The following lemma, which adapts Lemma 2.3.1 to this setting, allows to compute the steepest descent direction of the functional F_ε in $\mathbb{C}^{n\times n}$, which means neglecting any structural constraint. For this reason, we refer to it as the *free gradient* of the functional.

Lemma 4.2.3. *Let $E(t) \in \mathbb{C}^{n\times n}$, for t near t_0, be a continuously differentiable path of matrices, with the derivative denoted by $\dot{E}(t)$. Assume that $\lambda(t)$ is a simple eigenvalue of $A+\varepsilon E(t)$ depending continuously on t, with associated eigenvectors $x(t)$ and $y(t)$ satisfying* (4.5)*, and let the eigenvalue condition number be*

$$\kappa(t) = \frac{1}{x(t)^* y(t)} > 0.$$

Then, $F_\varepsilon(E(t)) = f\big(\lambda(t), \overline{\lambda(t)}\big)$ is continuously differentiable with respect to t and we have

$$\frac{1}{\varepsilon\kappa(t)} \frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) = \mathrm{Re}\langle G_\varepsilon(E), \dot{E}(t)\rangle, \tag{4.6}$$

*where the (rescaled) gradient of F_ε is the rank-*1 *matrix*

$$G_\varepsilon(E) = 2 f_{\overline{\lambda}}\, x y^* \in \mathbb{C}^{n\times n} \qquad \text{with} \quad f_{\overline{\lambda}} = \frac{\partial f}{\partial \overline{\lambda}}(\lambda, \overline{\lambda}).$$

Proof. It follows directly from Lemma 4.2.1, Lemma 2.3.1 and Lemma 3.1.1, but we also show the direct computation:

$$\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon\left(E(t)\right) = f_\lambda\, \dot{\lambda} + f_{\overline{\lambda}}\, \dot{\overline{\lambda}} = \frac{\varepsilon}{x^* y}\left(f_\lambda\, x^* \dot{E} y + f_{\overline{\lambda}} \overline{x^* \dot{E} y}\right) = \frac{\varepsilon}{x^* y}\, 2\,\mathrm{Re}\left(f_\lambda\, x^* \dot{E} y\right).$$

□

It is easy to compute the coefficient of the free gradient introduced in Lemma 4.2.3 in the cases we are considering. When the target eigenvalue is the one with largest real part we have:

$$f(\lambda, \overline{\lambda}) = -\mathrm{Re}(\lambda), \quad 2 f_{\overline{\lambda}} = -1, \quad G_\varepsilon(E) = -x y^*,$$

which is non-zero for all λ. Instead if the target eigenvalue is the one with largest absolute value we have:

$$f(\lambda, \overline{\lambda}) = -|\lambda|^2, \quad 2 f_{\overline{\lambda}} = -2\lambda, \quad G_\varepsilon(E) = -2\lambda x y^*,$$

which is non-zero whenever $\lambda \neq 0$, i.e. if the original matrix has at least a non-zero eigenvalue. In Lemma 4.2.6 we discuss the assumption made in Chapter 2 which states that in general $G_\varepsilon(E) \neq 0$.

The next step we need to focus on is the preservation of the structure and the unit Frobenius norm of the perturbation. Let us consider a differentiable path of structured matrices $E(t)$ in the linear space $\mathcal{S}$. Since the subspace does not depend on t, then also $\dot{E}(t) \in \mathcal{S}$ and Lemma 4.2.3 and the properties of the projection $\Pi_{\mathcal{S}}$ yield

$$\frac{1}{\varepsilon\kappa(t)} \frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) = \mathrm{Re}\langle \Pi_{\mathcal{S}} G_\varepsilon(E(t)), \dot{E}(t)\rangle, \tag{4.7}$$

where the free gradient has been replaced by the rescaled structured gradient $\Pi_{\mathcal{S}} G_\varepsilon(E)$, which is the projection onto $\mathcal{S}$ of a rank-1 matrix. To fulfil the constraint

$$E(t) \subseteq \mathcal{S}_1 := \{M \in \mathcal{S} \ : \ \|M\|_F = 1\},$$

we must have

$$0 = \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|E(t)\|_F^2 = \mathrm{Re}\langle E(t), \dot{E}(t)\rangle.$$

In view of equation (4.7) we are thus led to the following constrained optimization problem for the admissible direction of steepest descent.

Lemma 4.2.4. *Given $E \in \mathcal{S}_1$ and $G \in \mathbb{C}^{n\times n}$, the solution of the optimization problem*

$$\arg\min_{Z\in\mathcal{S}_1,\ \mathrm{Re}\langle Z,E\rangle=0} \quad \mathrm{Re}\langle \Pi_{\mathcal{S}} G, Z\rangle$$

is

$$Z_\star = \frac{-\Pi_{\mathcal{S}} G + \mathrm{Re}\langle \Pi_{\mathcal{S}} G, E\rangle E}{\| -\Pi_{\mathcal{S}} G + \mathrm{Re}\langle \Pi_{\mathcal{S}} G, E\rangle E\|_F}.$$

Proof. It follows directly from Lemma 3.1.2. □

Lemmas 4.2.3 and 4.2.4 show that the admissible direction of steepest descent of the functional F_ε at a matrix $E \in \mathcal{S}$ of unit Frobenius norm is given by the positive multiples of the matrix $-\Pi_{\mathcal{S}} G_\varepsilon(E) + \mathrm{Re}\langle \Pi_{\mathcal{S}} G_\varepsilon(E), E\rangle E$. This leads us to consider the (rescaled) gradient flow on the manifold $\mathcal{S}_1$ of matrices in the structure space $\mathcal{S}$ of unit Frobenius norm:

$$\dot{E} = -\Pi_{\mathcal{S}} G_\varepsilon(E) + \mathrm{Re}\langle \Pi_{\mathcal{S}} G_\varepsilon(E), E\rangle E. \tag{4.8}$$

By construction of this ordinary differential equation, we have that $\dot{E} \in \mathcal{S}$ for $E \in \mathcal{S}$ and $\mathrm{Re}\langle E, \dot{E}\rangle = 0$ along its solutions, and so both the structure $\mathcal{S}$ and the Frobenius norm 1 are conserved.

As we follow the admissible direction of steepest descent of the functional F_ε along solutions $E(t)$ of the ODE (4.8), we obtain the following.

Theorem 4.2.5. *Assume that $\lambda(t)$ is a simple eigenvalue of $A + \varepsilon E(t)$ and that $\lambda(\cdot)$ is continuous at t. Let $E(\cdot)$ of unit Frobenius norm satisfy the differential equation* (4.8). *Then,*

$$\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) \le 0.$$

Proof. We write $G = G_\varepsilon(E)$ for short and take the inner product of (4.8) with $\dot{E}$. Using that $\mathrm{Re}\langle E, \dot{E}\rangle = 0$, we find

$$\|\dot{E}\|_F^2 = -\mathrm{Re}\langle G - \mathrm{Re}\langle G, E\rangle E, \dot{E}\rangle = -\mathrm{Re}\langle G, \dot{E}\rangle$$

and hence Lemma 4.2.3 and (4.8) yield

$$\frac{1}{\varepsilon\kappa}\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) = \mathrm{Re}\langle G, \dot{E}\rangle = -\|\dot{E}\|_F^2 = -\|G - \mathrm{Re}\langle G, E\rangle E\|_F^2 \le 0,$$

which gives the precise rate of decay of F_ε along a trajectory $E(t)$ of (4.8). □

Theorem 4.2.5 shows that the ODE (4.8) is a gradient system; in order to find its stationary points we need the following important result that states the non-vanishing property of the structured gradient.

Lemma 4.2.6. *Let $A, E \in \mathcal{S}$ and $\varepsilon > 0$, and let λ be a simple target eigenvalue of $A + \varepsilon E$.*

(i) Complex case: let $\mathcal{S}$ be a complex-linear subspace of $\mathbb{C}^{n\times n}$. Then,

$$\Pi_{\mathcal{S}} G_\varepsilon(E) \ne 0 \quad \text{if} \quad \overline{\lambda} f_{\overline{\lambda}} \ne 0.$$

(ii) Real case: let $\mathcal{S}$ be a real-linear subspace of $\mathbb{R}^{n\times n}$. Then,

$$\Pi_{\mathcal{S}} G_\varepsilon(E) \neq 0 \quad \text{if} \quad \mathrm{Re}(\overline{\lambda} f_{\overline{\lambda}}) \neq 0.$$

Proof. We give the proof for the real case. The complex case is analogous but slightly simpler. We take the real inner product of $\Pi_{\mathcal{S}} G_\varepsilon(E)$ with $A+\varepsilon E \in \mathcal{S}$ and use the definition of $\Pi_{\mathcal{S}} G_\varepsilon(E)$:

$$\begin{aligned}\langle \Pi_{\mathcal{S}} G_\varepsilon(E), A+\varepsilon E\rangle &= \mathrm{Re}\langle \Pi_{\mathcal{S}}(2 f_{\overline{\lambda}} xy^*), A+\varepsilon E\rangle = \mathrm{Re}\langle 2 f_{\overline{\lambda}} xy^*, A+\varepsilon E\rangle \\ &= \mathrm{Re}\big(2 f_\lambda\, x^*(A+\varepsilon E)y\big) = \mathrm{Re}\big(2 f_\lambda \lambda\, x^* y\big) = 2\ \mathrm{Re}\big(f_{\overline{\lambda}}\overline{\lambda}\big)\ (x^*y),\end{aligned}$$

where $x^*y > 0$ by (4.5). This yields the claim. □

Remark 4.2.7. *If the identity matrix I is in $\mathcal{S}$, then the condition for $\Pi_{\mathcal{S}} G_\varepsilon(E) \neq 0$ can be weakened:*

(i) In the complex case, it then suffices to have $f_{\overline{\lambda}} \neq 0$. This is seen by taking the inner product with $A+\varepsilon E - \mu I \in \mathcal{S}$ for an arbitrary $\mu \in \mathbb{C}$.

(ii) In the real case, if λ is real, then it suffices to have $\mathrm{Re}\, f_{\overline{\lambda}} \neq 0$. If λ is non-real, then it even suffices to have $f_{\overline{\lambda}} \neq 0$. In both cases this is seen by taking the inner product with $A+\varepsilon E - \mu I \in \mathcal{S}$ for an arbitrary $\mu \in \mathbb{R}$.

In the rest of the chapter we implicitly assume that the structured gradient does not vanish, as this is a generic property. Moreover we choose the time interval to be $\mathcal{I} = [0,+\infty)$.

We have the following characterization of stationary points of the norm- and structure-constrained gradient system (4.8) on $\mathcal{S}_1$.

Theorem 4.2.8. *Let $E(t) \subseteq \mathcal{S}_1$ be a solution of equation (4.8) passing through $E_\star = E(t_\star)$ at time $t_\star > 0$ and assume that $\Pi_{\mathcal{S}} G_\varepsilon(E_\star) \neq 0$. Then the following facts are equivalent:*

1. $\displaystyle \frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t))\Big|_{t=t_\star} = 0,$
2. *$E_\star$ is a stationary point of (4.8),*
3. *$E_\star$ is a non-zero real multiple of $\Pi_{\mathcal{S}} G_\varepsilon(E_\star)$.*

Proof. Follows from Theorem 3.1.4. □

Since minimizers of the optimization problem (4.3) are stationary points of the norm- and structure-constrained gradient system (4.8), Theorem 4.2.8 immediately yields the following corollary.

Corollary 4.2.9. *Optimizers $E_\star$ of (4.3) are projections onto $\mathcal{S}$ of rank-1 matrices.*

This provides the motivation to search for a differential equation that retains the rank-1 property along its solutions. We describe a first, seemingly obvious approach in the next subsection and then turn to a less obvious alternative in Section 4.3 on which we focus in Sections 4.3 to 4.7.

4.2.3 Constrained gradient flow for structure-projected rank-1 matrices

Let $\mathcal{M}_1$ be the manifold of complex $n \times n$ rank-1 matrices, and let $\mathcal{M}_1^{\mathcal{S}} = \Pi_{\mathcal{S}}\mathcal{M}_1$ be the set of $\mathcal{S}$-projected rank-1 matrices. We note that $\mathcal{M}_1^{\mathcal{S}}$ need not be a manifold. For example, for $\mathcal{S} = \mathbb{R}^{n\times n} = \mathrm{Re}\,\mathbb{C}^{n\times n}$ we have $\mathcal{M}_1^{\mathcal{S}} = \mathrm{Re}\,\mathcal{M}_1$, which is the union of $\{0\}$ and the two manifolds of real rank-1 and rank-2 matrices. Let us suppose in this short subsection that $\mathcal{M}_1^{\mathcal{S}}$ is a manifold, at least locally in a neighbourhood of interest. For $E \in \mathcal{M}_1^{\mathcal{S}}$ (in such a neighbourhood), we then let $\mathcal{T}_E\mathcal{M}_1^{\mathcal{S}}$ be the tangent space at E of $\mathcal{M}_1^{\mathcal{S}}$. We further suppose that the orthogonal projection $P_E^{\mathcal{S}}$ onto the tangent space $\mathcal{T}_E\mathcal{M}_1^{\mathcal{S}}$ is computationally readily available. This is the case when the structure space $\mathcal{S}$ consists of matrices of prescribed range and co-range, as will be discussed in Section 4.8. However, this is not the case when the structure is given by a sparsity pattern, for which we therefore propose the alternative approach of Section 4.3.

We consider the projected gradient system on the manifold $\mathcal{M}_1^{\mathcal{S}}$:

$$\dot{E} = -P_E^{\mathcal{S}}\Pi_{\mathcal{S}}G_\varepsilon(E) + \mathrm{Re}\,\langle P_E^{\mathcal{S}}\Pi_{\mathcal{S}}G_\varepsilon(E), E\rangle E. \tag{4.9}$$

We note that $P_E^{\mathcal{S}}E = E$ for $E \in \mathcal{M}_1^{\mathcal{S}}$, because the fact that scalar multiples of E are again in $\mathcal{M}_1^{\mathcal{S}}$ implies that $E \in \mathcal{T}_E\mathcal{M}_1^{\mathcal{S}}$. Therefore, the right-hand side of (4.9) is in the tangent space $\mathcal{T}_E\mathcal{M}_1^{\mathcal{S}}$, and so we have a differential equation on $\mathcal{M}_1^{\mathcal{S}}$. Since $\mathrm{Re}\langle E, \dot{E}\rangle = 0$, the unit Frobenius norm is preserved. Moreover, we again have the monotonicity property of Theorem 4.2.5 by the same argument as before. Under the non-degeneracy condition $P_E^{\mathcal{S}}\Pi_{\mathcal{S}}G_\varepsilon(E_\star) \neq 0$, we have that $E_\star \in \mathcal{M}_1^{\mathcal{S}}$ is a stationary point of (4.9) if and only if $E_\star$ is a real multiple of $P_{E_\star}^{\mathcal{S}}\Pi_{\mathcal{S}}G_\varepsilon(E_\star)$. Clearly, every stationary point $E_\star$ of (4.8) is also a stationary point of (4.9). In fact if the right-hand side $R(E)$ of (4.8) vanishes, then also $P_E^{\mathcal{S}}R(E)$ vanishes, which is the right-hand side of (4.9). However, in general we cannot exclude that (4.9) may have additional, spurious stationary points $E_\odot$ that are not a real multiple of $G_\varepsilon^{\mathcal{S}}(E_\odot)$.

In Section 4.8 we show how the projected gradient system (4.9) can actually be used in computations when the structure space $\mathcal{S}$ consists of complex matrices with prescribed range and co-range. Moreover, we find that in this particular case no spurious stationary points are possible.

4.3 A rank-1 matrix differential equation

When the structure space $\mathcal{S}$ consists of matrices with a prescribed sparsity pattern, where the tangent space projection $P_E^{\mathcal{S}}$ is not readily available, and the projected gradient system (4.9) of the previous subsection can apparently not be used in a computationally efficient way. As a more accessible alternative, we consider a differential equation on the manifold $\mathcal{M}_1$ of rank-1 matrices, which uses only the known and computationally very simple orthogonal projections $\Pi_{\mathcal{S}}$ onto the structure and P_Y onto the tangent space $\mathcal{T}_Y\mathcal{M}_1$ at $Y \in \mathcal{M}_1$ (note that in general $P_E^{\mathcal{S}} \neq \Pi_{\mathcal{S}}P_Y$ for $E = \Pi_{\mathcal{S}}Y$ and $Y \in \mathcal{M}_1$, since $\Pi_{\mathcal{S}}$ and P_Y do not commute). The alternative differential equation is shown to lead to the same stationary points as the structure- and norm-constrained gradient flow (4.8), without any spurious stationary points (under a non-degeneracy condition). However, this differential equation is not a gradient system, and the monotonicity property of Theorem 4.2.5 is therefore not guaranteed (though it is usually observed in numerical experiments). Also for this alternative differential equation, reformulated for the factors of the rank-1 matrices, we will numerically approximate its stationary points.

4.3.1 Formulation and properties of the rank-1 differential equation

Solutions of (4.8) can be written as $E(t) = \Pi_{\mathcal{S}} Z(t)$, where $Z(t)$ solves

$$\dot{Z} = -G_\varepsilon(\Pi_{\mathcal{S}} Z) + \operatorname{Re}\langle G_\varepsilon(\Pi_{\mathcal{S}} Z), \Pi_{\mathcal{S}} Z\rangle Z, \tag{4.10}$$

as it is immediately seen by projecting both sides onto $\mathcal{S}$ with $\Pi_{\mathcal{S}}$ and comparing with (4.8). We note that if $E(t) = \Pi_{\mathcal{S}} Z(t)$ has unit Frobenius norm, then

$$\operatorname{Re}\langle E, \dot{E}\rangle = -\operatorname{Re}\langle E, G_\varepsilon(E)\rangle + \operatorname{Re}\langle G_\varepsilon(E), E\rangle, \operatorname{Re}\langle E, E\rangle = 0.$$

Therefore, the unit Frobenius norm of $E(t) = \Pi_{\mathcal{S}} Z(t)$ is conserved for all t. Since $G_\varepsilon(E)$ is of rank 1 (unless $G_\varepsilon(E) = 0$, which we exclude), every stationary point $Z_\star$ of the differential equation (4.10) is of rank 1. We therefore project the right-hand side onto the tangent space $\mathcal{T}_Y \mathcal{M}_1$ at $Y \in \mathcal{M}_1$ and consider instead the projected differential equation with solutions of rank 1:

$$\dot{Y} = -P_Y G_\varepsilon(\Pi_{\mathcal{S}} Y) + \operatorname{Re}\langle P_Y G_\varepsilon(\Pi_{\mathcal{S}} Y), \Pi_{\mathcal{S}} Y\rangle Y. \tag{4.11}$$

Here, $P_Y : \mathbb{C}^{n\times n} \to \mathcal{T}_Y \mathcal{M}_1$ is the orthogonal projection onto the tangent space $\mathcal{T}_Y \mathcal{M}_1$, which for a rank-1 matrix $Y = \sigma u v^*$ with $\|u\| = \|v\| = 1$ is given as (see [51] and Proposition A.0.2)

$$P_Y(Z) = Z - (I - uu^*)Z(I - vv^*). \tag{4.12}$$

It is useful to note that $P_Y(Y) = Y$. For $E = \Pi_{\mathcal{S}} Y$ of unit Frobenius norm in (4.11), we find

$$\operatorname{Re}\langle E, \dot{E}\rangle = \operatorname{Re}\langle E, \dot{Y}\rangle = -\operatorname{Re}\langle E, P_Y G_\varepsilon(E)\rangle + \operatorname{Re}\langle P_Y G_\varepsilon(E), E\rangle \operatorname{Re}\langle E, Y\rangle = 0,$$

where we used that $\operatorname{Re}\langle E, Y\rangle = \operatorname{Re}\langle \Pi_{\mathcal{S}} E, Y\rangle = \operatorname{Re}\langle E, \Pi_{\mathcal{S}} Y\rangle = \operatorname{Re}\langle E, E\rangle = \|E\|_F^2 = 1$. So, for all t, we have that $E = \Pi_{\mathcal{S}} Y$ has unit Frobenius norm.

4.3.2 Stationary points

The following theorem states that the differential equations (4.8) and (4.11) yield the same stationary points.

Theorem 4.3.1. *There exists a one-to-one correspondence between the stationary points of the original gradient system and those of the rank-*1 *matrix ODE.*

(a) *Let $E_\star \in \mathcal{S}$ of unit Frobenius norm be a stationary point of the gradient system* (4.8)*. Then, $E_\star = \Pi_{\mathcal{S}} Y_\star$ for some matrix $Y_\star \in \mathcal{M}_1$ that is a stationary point of the differential equation* (4.11)*.*

(b) *Conversely, let $Y_\star \in \mathcal{M}_1$ be a stationary point of the differential equation* (4.11) *such that $E_\star = \Pi_{\mathcal{S}} Y_\star$ has unit Frobenius norm and $P_{Y_\star} G_\varepsilon(E_\star) \neq 0$. Then, $P_{Y_\star} G_\varepsilon(E_\star) = G_\varepsilon(E_\star)$, $Y_\star$ is a non-zero real multiple of $G_\varepsilon(E_\star)$, and $E_\star$ is a stationary point of the gradient system* (4.8)*.*

Proof. It is a direct consequence of Theorem 3.1.7. However we report it for this specific case, since the proof is easier in the rank-1 setting. Let $G_\star = G_\varepsilon(E_\star)$ in this proof for short.

(a) By Theorem 4.2.8, $E_\star = \mu^{-1} \Pi_{\mathcal{S}} G_\star$ for some non-zero real μ. Then, $Y_\star := \mu^{-1} G_\star$ is of rank 1 and we have $E_\star = \Pi_{\mathcal{S}} Y_\star$. We further note that $P_{Y_\star} G_\star = \mu P_{Y_\star} Y_\star = \mu Y_\star = G_\star$. Thus

$$-P_{Y_\star} G_\star + \operatorname{Re}\langle P_{Y_\star} G, E_\star\rangle Y_\star = -G_\star + \operatorname{Re}\langle G, E_\star\rangle Y_\star,$$

and hence

$$\mathrm{Re}\langle G_\star, E_\star\rangle = \mathrm{Re}\langle \Pi_{\mathcal{S}} G_\star, E_\star\rangle = \mathrm{Re}\langle \mu E_\star, E_\star\rangle = \mu \|E_\star\|_F^2 = \mu.$$

So we have

$$-G_\star + \mathrm{Re}\langle G_\star, E_\star\rangle Y_\star = -G_\star + \mu Y_\star = 0$$

by the definition of $Y_\star$. This shows that $Y_\star$ is a stationary point of (4.11).

(b) We show that $Y_\star$ is a non-zero real multiple of $G_\star$. By Theorem 4.2.8, $E_\star$ is then a stationary point of the differential equation (4.8). For a stationary point $Y_\star$ of (4.11), we have that $P_{Y_\star}(G_\star)$ is a non-zero real multiple of $Y_\star$. Hence, in view of $P_{Y_\star}(Y_\star) = Y_\star$, we can write $G_\star$ as

$$G_\star = \mu Y_\star + W, \qquad \text{where } \mu \neq 0 \text{ is real and } P_{Y_\star}(W) = 0.$$

Writing the rank-1 matrix $Y_\star = \rho u v^*$ with $\rho \neq 0$ and $\|u\| = \|v\| = 1$, we then have by (4.12) that

$$W = W - P_{Y_\star}(W) = (I - uu^*) W (I - vv^*).$$

On the other hand, $G_\star = 2\overline{f_\lambda} x y^*$ is also of rank 1. So we have

$$2\overline{f_\lambda} x y^* = \mu u v^* + (I - uu^*) W (I - vv^*).$$

Multiplying from the right with v yields that x is a complex multiple of u, and multiplying from the left by u^* yields that y is a complex multiple of v. Hence, $G_\star$ is a complex multiple of $Y_\star$. Since we already know that $P_{Y_\star}(G_\star)$ is a non-zero real multiple of $P_{Y_\star}(Y_\star) = Y_\star$, it follows that $G_\star$ is the same real multiple of $Y_\star$. Thus stationary points $Y_\star \in \mathcal{M}_1$ of the differential equation (4.11) are characterized as real multiples of $G_\star$. Hence, $E_\star = \Pi_{\mathcal{S}} Y_\star$ is a real multiple of $\Pi_{\mathcal{S}} G_\star$, and by Theorem 4.2.8, $E_\star = \Pi_{\mathcal{S}} Y_\star$ is a stationary point of (4.8). □

4.3.3 Possible loss of monotonicity

Since the projections $\Pi_{\mathcal{S}}$ and P_Y do not commute, along solutions of (4.11) we cannot guarantee the monotonicity property of Theorem 4.2.5 that we have for the constrained gradient system (4.8). However, in all our numerical experiments we observed that starting with an initial datum given by the negative free gradient of the considered functional, i.e. $Y(0) = -G_\varepsilon(0)$, we always obtained a monotone convergence behaviour to a (local) optimum. Only in very few cases, by starting from a randomly chosen initial datum, we were able to observe a non-monotonic convergence. However the loss of monotonicity occurred only once, after the first step, and monotonicity was recovered from the following step onwards (see Figure 4.1). In the following section we will explain this behaviour locally near a stationary point, but we have no theoretical explanation for the favourable numerically observed monotonic behaviour far from stationary points.

4.3.4 Differential equations for the factors of rank-1 matrices

Equation (4.11) is an abstract differential equation on the rank-1 manifold $\mathcal{M}_1$. We write any matrix $Y \in \mathcal{M}_1$ in a non-unique way as

$$Y = \rho u v^*,$$

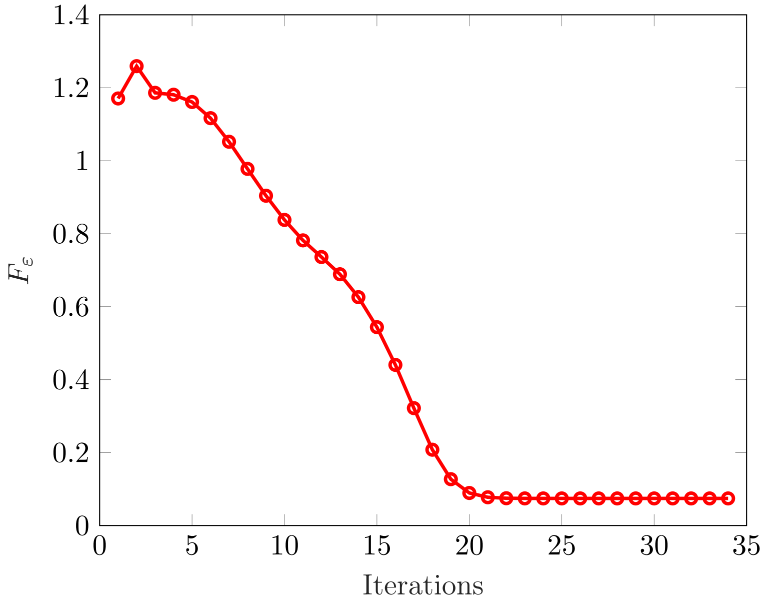


FIGURE 4.1: Non-monotonic decrease of the objective functional F_ε during the integration of equation (4.8).

where $\rho \in \mathbb{R}$ with $\rho > 0$ and $u, v \in \mathbb{C}^n$ have unit norm. The following lemma shows how we can rewrite the rank-1 differential equation (4.11) in terms of differential equations for the factors u, v and an explicit formula for ρ.

Lemma 4.3.2. *Let $Y(t) \subseteq \mathcal{M}_1$ be a solution of the rank-1 differential equation (4.11) such that $\|\Pi_{\mathcal{S}} Y(t)\|_F = 1$. Then it is always possible to decompose it as*

$$Y(t) = \rho(t)u(t)v(t)^*,$$

where $\rho(t)$ fulfils the unit norm constraint on $E = \Pi_{\mathcal{S}} Y$, that is

$$\rho = \frac{1}{\|\Pi_{\mathcal{S}}(uv^*)\|_F},$$

and the unit norm factors $u(t)$ and $v(t)$ are solutions of the ODEs

$$\begin{aligned} \rho\dot{u} &= -(I - uu^*)Gv - \frac{\mathrm{i}}{2}\operatorname{Im}(u^*Gv)u, \\ \rho\dot{v} &= -(I - vv^*)G^*u + \frac{\mathrm{i}}{2}\operatorname{Im}(u^*Gv)v, \end{aligned}$$

where $G = G_\varepsilon(E)$.

Proof. The equation for ρ is obvious because $1 = \|E\|_F = \rho\|\Pi_{\mathcal{S}}(uv^*)\|_F$. We write the right-hand side of (4.11) and use (4.12) to obtain for $Y = \rho uv^*$

$$\dot{Y} = -P_Y G + \operatorname{Re}\langle P_Y G, E\rangle Y =$$

$$= -(I - uu^*)Gvv^* - uu^*G(I - vv^*) - uu^*Gvv^* + \operatorname{Re}\langle P_Y G, E\rangle Y =$$

$$= -\Big((I - uu^*)Gvv^* + \omega u\Big)v^* - u\Big(u^*G(I - vv^*) + \omega v^*\Big) - \Big(\zeta + \operatorname{Re}\langle P_Y G, E\rangle \rho\Big)uv^*,$$

where $\zeta = \mathrm{Re}(u^*Gv)$ and $\omega = \frac{\mathrm{i}}{2}\,\mathrm{Im}(u^*Gv)$ are such that $u^*Gv = \zeta + 2\omega$. Since it is also possible to write

$$\dot{Y} = (\rho\dot{u})v^* + u(\rho\dot{v}^*) + \dot{\rho}uv^*,$$

we can read off $\rho\dot{u}$, $\rho\dot{v}^*$ and $\dot{\rho}$ as the three terms in big brackets in the former expression for $\dot{Y}$. This yields the stated differential equations for u and v (and another one for ρ, which will not be needed). Moreover, since $\mathrm{Re}(\omega) = 0$, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 = 2\,\mathrm{Re}(u^*\dot{u}) = \frac{2}{\rho}\left(\mathrm{Re}(-u^*(I-uu^*)Gv) + \mathrm{Re}(\omega)\|u\|^2\right) = 0$$

and analogously for v, so that the unit norm of u and v is conserved. □

The positive factor ρ on the left-hand sides of the differential equations for u and v only determines the speed with which the trajectory is traversed, but has no influence on the trajectory itself. Since Lemma 4.2.3 provides the explicit expression for $G = G_\varepsilon(E) = 2f_{\overline{\lambda}}xy^*$, we can rewrite the differential equations for u and v as

$$\begin{cases} \rho\dot{u} = (\alpha\overline{\beta}\gamma)u - (\overline{\beta}\gamma)x - \dfrac{\mathrm{i}}{2}\,\mathrm{Im}(\alpha\overline{\beta}\gamma)u, \\[2ex] \rho\dot{v} = (\overline{\alpha}\beta\overline{\gamma})v - (\overline{\alpha\gamma})y - \dfrac{\mathrm{i}}{2}\,\mathrm{Im}(\overline{\alpha}\beta\overline{\gamma})v, \end{cases} \tag{4.13}$$

where $\alpha = u^*x$, $\beta = v^*y$ and $\gamma = 2f_{\overline{\lambda}}$.

4.3.5 Cases of interest for the rank-1 ODE

The real dimension of the manifold of complex $n \times n$ rank-1 matrices of unit norm is $4n-2$. Integrating (4.11) instead of (4.8) is very appealing in those cases where $\dim(\mathcal{S})$ is significantly larger than $4n-2$. An important example is given by sparse matrices with a sparsity pattern with a number of non-zero elements of order cn with $c > 4$ (and ideally much larger than 4). In the case of a real target eigenvalue the dimension of the manifold of real $n \times n$ rank-1 matrices of unit norm is $2n-1$ so that for structured matrices it is meaningful to make use of (4.11) if $c > 2$. Similarly, when considering matrices with prescribed range and co-range,

$$\mathcal{S} = \{B\Delta C : \Delta \in \mathbb{R}^{k\times l}\}, \tag{4.14}$$

where $B \in \mathbb{R}^{n\times k}$ and $C \in \mathbb{R}^{l\times n}$ with $k, l < n$, replacing the unknown matrix Δ, which is a full $k \times l$ real matrix, by a rank-1 matrix, significantly reduces the memory requirements when k and l are large. As for the computational cost, we may argue that the reduced number of variables may lead to a faster convergence of the method.

4.4 Local convergence to the rank-1 stationary points

In this section we show that solutions of the rank-1 projected differential equation (4.11) converge locally to strong (or strict) local minima of the functional F_ε, that is the solution converges to a local minimum E for which the Hessian matrix $H_\varepsilon(E)$ of F_ε at E yields a positive definite quadratic form when restricted to the tangent space $\mathcal{T}_E\mathcal{S}_1$ of the manifold $\mathcal{S}_1$ at E (see also Definition 3.1.9). Here, $\mathcal{S}_1$ is the manifold of matrices in $\mathcal{S}$ of unit Frobenius norm. We formulate and prove a key lemma, analogous to Lemma 3.1.8, we discuss some assumptions needed for the main result and then we state the local convergence result.

Lemma 4.4.1. *Let $Y_\star \in \mathcal{M}_1$ with $E_\star = \Pi_{\mathcal{S}} Y_\star \in \mathcal{S}$ of unit Frobenius norm. Let $Y_\star$ be a stationary point of the rank-1 projected differential equation (4.11), with an associated target eigenvalue λ of $A + \varepsilon E_\star$ that is simple. Then, there exists $\bar\delta > 0$ such that, for all positive $\delta \le \bar\delta$ and all $Y \in \mathcal{M}_1$ with $\|Y - Y_\star\|_F \le \delta$ and $\Pi_{\mathcal{S}} Y$ of unit norm, we have*

$$\|P_Y G_\varepsilon(\Pi_{\mathcal{S}} Y) - G_\varepsilon(\Pi_{\mathcal{S}} Y)\|_F \le C\delta^2$$

with $C > 0$ independent of δ.

Proof. The proof is the same of that of Lemma 3.1.8, but in the rank-1 setting; we report it for completeness. Let us consider a smooth regular path $Y(\tau) = u(\tau)v(\tau)^* \in \mathcal{M}_1$ (with non-zero $u(\tau), v(\tau) \in \mathbb{C}^n$) such that $E(\tau) = \Pi_{\mathcal{S}} Y(\tau)$ is of unit Frobenius norm and

$$Y(0) = Y_\star = \alpha G_\star \quad \text{for some real } \alpha, \text{ where } G_\star = G_\varepsilon\left(E(0)\right) = 2\overline{f}_\lambda x y^*,$$

where (λ, x, y) is the eigentriplet of $A + \varepsilon E(0)$ associated with the target eigenvalue λ. Similarly, for $\tau \in [0, \delta]$ with δ such that $\lambda(\tau)$ remains simple, we have

$$G(\tau) = G_\varepsilon(E(\tau)) = 2\overline{f}_\lambda(\tau) x(\tau) y(\tau)^*.$$

We may assume that the path is parametrized such that $\|\dot Y(\tau)\|_F = 1$ and hence we have $\|Y(\tau) - Y_\star\|_F \sim \tau$ for small τ. By the given assumptions all quantities are smooth with respect to τ. In particular, for a simple eigenvalue, under a smooth matrix perturbation, the derivatives $\dot x(\tau)$ and $\dot y(\tau)$ of the associated eigenvectors, under the assumed normalization (4.5), are given by (see e.g. [31, 58])

$$\frac{1}{\varepsilon}\dot x(\tau)^* = -x(\tau)^* \dot E(\tau) N(\tau) + \mathrm{Re}\left(x(\tau)^* \dot E(\tau) N(\tau) x(\tau)\right) x(\tau)^*,$$

$$\frac{1}{\varepsilon}\dot y(\tau) = -N(\tau) \dot E(\tau) y(\tau) + \mathrm{Re}\left(y(\tau)^* N(\tau) \dot E(\tau) y(\tau)\right) y(\tau),$$

where $N(\tau)$ is the group inverse of $A + \varepsilon E(\tau) - \lambda(\tau) I$ and the last terms on the right-hand side of both differential equations account for the unit norm preservation for both eigenvectors and for the positivity of their inner product. Note that by the simplicity of $\lambda(\tau)$, the group inverse $N(\tau)$ and thus also $\dot x(\tau)$ and $\dot y(\tau)$ as well as their derivatives are bounded. With the formula (4.12) for the projection P_Y, we thus have the following first order expansion near $\tau = 0$. Here we indicate by u, v, x, y and $f_{\overline\lambda}$ (and further $\dot u, \dot v, \dot x, \dot y$ and $\dot f_{\overline\lambda}$) the associated functions of τ at $\tau = 0$, i.e. corresponding to the stationary point. We have

$$
\begin{aligned}
P_{Y(\tau)} G(\tau) = P_{Y(\tau)}\Big(f_{\overline\lambda}(\tau) x(\tau) y(\tau)^*\Big) = \\
= \Big(f_{\overline\lambda} + \tau \dot f_{\overline\lambda}\Big) \cdot \Bigg(\Big(xx^* + \tau\left(\dot u x^* + x \dot u^*\right)\Big)\Big(xy^* + \tau\left(\dot x y^* + x \dot y^*\right)\Big) + \\
+ \Big(xy^* + \tau\left(\dot x y^* + x \dot y^*\right)\Big)\Big(yy^* + \tau\left(\dot v y^* + y \dot v^*\right)\Big) + \\
- \Big(xx^* + \tau\left(\dot u x^* + x \dot u^*\right)\Big)\Big(xy^* + \tau\left(\dot x y^* + x \dot y^*\right)\Big)\Big(yy^* + \tau\left(\dot v y^* + y \dot v^*\right)\Big)\Bigg) + \mathcal{O}(\tau^2) = \\
= f_{\overline\lambda} x y^* + \tau\left(\dot f_{\overline\lambda} x y^* + f_{\overline\lambda} x \dot y^* + f_{\overline\lambda} \dot x y^*\right) + \mathcal{O}(\tau^2).
\end{aligned}
$$

Consequently, $P_{Y(\tau)}G(\tau)$ has the same first order expansion as

$$G(\tau) = f_{\overline{\lambda}}(\tau)x(\tau)y(\tau)^* = f_{\overline{\lambda}}xy^* + \tau\left(\dot{f}_{\overline{\lambda}}xy^* + f_{\overline{\lambda}}x\dot{y}^* + f_{\overline{\lambda}}\dot{x}y^*\right) + \mathcal{O}(\tau^2),$$

which yields the result. □

For the formulation of our local convergence result we need the following assumptions. Here, $\mathcal{M}_1$ is the manifold of rank-1 matrices in $\mathbb{C}^{n\times n}$, and $\mathcal{M}_1^{\mathcal{S}} = \Pi_{\mathcal{S}}\mathcal{M}_1$ consists of structure-projected rank-1 matrices. The first assumption is made on the structure space $\mathcal{S}$. It excludes, in particular, spaces $\mathcal{S}$ that are too low-dimensional: it requires $\dim(\mathcal{S}) \geq \dim(\mathcal{M}_1) = 4n-2$ (as before dim indicates the *real* dimension).

Assumption 4.4.2. *The restricted projection $\Pi_{\mathcal{S}}\big|_{\mathcal{M}_1} : \mathcal{M}_1 \to \mathcal{M}_1^{\mathcal{S}} \subset \mathcal{S}$ is a local diffeomorphism, or equivalently:*

(i) If $E = \Pi_{\mathcal{S}}Y \in \mathcal{M}_1^{\mathcal{S}}$ for some $Y \in \mathcal{M}_1$, then Y is locally unique.

(ii) The local inverse map $(\Pi_{\mathcal{S}}\big|_{\mathcal{M}_1})^{-1} : E \to Y$ is continuously differentiable.

Remark 4.4.3. *We comment on Assumption 4.4.2 to (a) make it plausible in the case of perturbation matrices E with prescribed sparsity pattern and (b) show that it is not satisfied in the case of perturbation matrices with prescribed range and co-range.*

(a) Consider the structure $\mathcal{S}$ of real $n \times n$ matrices with a prescribed sparsity pattern. Let $E \in \mathcal{M}_1^{\mathcal{S}} \subset \mathcal{S}$ be given. So $E = \Pi_{\mathcal{S}}\widehat{Y}$ for some $\widehat{Y} \in \mathcal{M}_1$. In principle, in order to determine all solutions of the equation

$$\Pi_{\mathcal{S}}Y = E$$

we should form $Y = uv^*$ with $u, v \in \mathbb{C}^n$ with $\|u\| = 1$ and $v \neq 0$, and write a system of quadratic equations in the variables $\{u_i\}_{i=1}^n$ and $\{v_j\}_{j=1}^n$ that reads

$$\mathrm{Re}\left(u_i v_j^*\right) = E_{ij} \qquad \text{for all } (i,j) \in \mathscr{S}$$

where $\mathscr{S}$ is the considered sparsity pattern, together with the norm constraint $\|u\|^2 = 1$, and moreover the first non-zero entry of u can be chosen to be real and positive to guarantee uniqueness of the representation $Y = uv^*$. This gives a system of $s+1$ quadratic equations where $s = \#\mathscr{S} = \dim(\mathcal{S})$ is the number of entries of E which are not prescribed to be zero. In terms of the real variables $\mathrm{Re}(u_i), \mathrm{Im}(u_i), \mathrm{Re}(v_i), \mathrm{Im}(v_i)$ (excluding $\mathrm{Im}(u_1) = 0$), the system has $s+1$ quadratic equations in $4n-1$ variables: $\Phi(u,v) \equiv \Pi_{\mathcal{S}}Y = E$. We have local uniqueness of $\widehat{Y} = \hat{u}\hat{v}^*$ if the derivative matrix $D\Phi(\hat{u},\hat{v}) \in \mathbb{C}^{(s+1)\times(4n-1)}$ has only the trivial kernel 0. This can be expected to hold true generically if $s \geq 4n-2$ (and the more so as s gets larger). On the other hand, if $s < 4n-2$, then integrating the gradient system (4.8), translated into a system of differential equations in terms of the s non-zero entries of E, would be favorable over integrating the rank-1 matrix differential equation (4.8). This further indicates that Assumption 4.4.2 is reasonable in the case where the structure is given by a sparsity pattern. For the structure space $\mathcal{S}$ of matrices with a prescribed sparsity pattern, Assumption 4.4.2 is reminiscent of the problem of matrix completion, where the aim is to minimize the rank r such that there exists a unique matrix M of rank r with $\Pi_{\mathcal{S}}M = E$ for a given matrix $E \in \mathcal{S}$; see e.g. [12]. Note, however, that in Assumption 4.4.2 the condition is not about existence but about local uniqueness, and the rank is fixed to 1.

(b) Assumption 4.4.2 is not satisfied in the case where the structure $\mathcal{S}$ is given by matrices with prescribed range and co-range. Since in this case the orthogonal projection

onto the structure is given by $\Pi_{\mathcal{S}}Y = BB^{\dagger}YC^{\dagger}C$ *(see Proposition B.0.5), we have that for a rank-*1 *matrix* $Y = uv^*$*, the projected matrix* $E = \Pi_{\mathcal{S}}Y$ *can also be written as* $E = \Pi_{\mathcal{S}}\widetilde{Y}$ *with* $Y = (u + \widetilde{u})(v + \widetilde{v})^*$ *for arbitrary* $\widetilde{u} \in \ker(B^\top)$ *and* $\widetilde{v} \in \operatorname{Ker} C^\top$*, and so condition (i) in Assumption 4.4.2 is violated. This could be remedied by requiring that* $Y = uv^*$ *be such that* $u \in \ker(B^\top)^\perp = \operatorname{Ran} B$ *and* $v \in \ker(C^\top)^\perp = \operatorname{Ran} C$ *and incorporating these constraints in the differential equation. We will not carry this out in detail for two reasons: On the one hand it did not seem necessary in our numerical experiments, and on the other hand we can here work instead with the projected gradient system* (4.9) *on* $\mathcal{M}_1^{\mathcal{S}}$*, as is described in Section 4.8.*

The next assumption is made on the Hessian of the functional F_ε at a stationary point of the differential equation (4.8).

Assumption 4.4.4. *Let* $E_0 \in \mathcal{S}_1$ *be a stationary point of the constrained gradient system* (4.8)*. We assume that* E_0 *is a strict minimum of the functional* F_ε *on* $\mathcal{S}_1$ *(see Definition 3.1.9), that is, the Hessian matrix* $H_\varepsilon(E_0)$ *of* F_ε *at* E_0 *yields a positive definite quadratic form when restricted to the tangent space* $\mathcal{T}_{E_0}\mathcal{S}_1$ *of the manifold* $\mathcal{S}_1$ *at* E_0*. In other words there exists* $\alpha > 0$ *such that*

$$\langle Z, H_\varepsilon(E_0)Z\rangle \geq \alpha\|Z\|_F^2, \qquad \forall\, Z \in \mathcal{T}_{E_0}\mathcal{S}_1.$$

Under these assumptions we have the following result.

Theorem 4.4.5 (Local convergence to a strict local minimum)**.** *Under Assumption* 4.4.2*, let the rank-1 matrix* $Y_0 \in \mathcal{M}_1$ *be a stationary point of the projected differential equation* (4.11) *such that* $E_0 = \Pi_{\mathcal{S}}Y_0 \in \mathcal{S}_1$ *is of unit Frobenius norm and* $P_{Y_0}G_\varepsilon(E_0) \neq 0$*. We assume that* E_0 *satisfies Assumption 4.4.4. Then, for an initial datum* $Y(0)$ *sufficiently close to* Y_0*, the solution* $Y(t)$ *of* (4.11) *converges to* Y_0 *exponentially as* $t \to \infty$*. Moreover,* $F_\varepsilon\left(\Pi_{\mathcal{S}}Y(t)\right)$ *decreases monotonically with* t *and converges exponentially to the local minimum value* $F_\varepsilon(E_0)$ *as* $t \to \infty$*.*

Proof. See Theorem 3.1.10. □

Note that $E_0 = \Pi_{\mathcal{S}}Y_0 \in \mathcal{S}_1$ is a stationary point of (4.8) by Theorem 4.3.1. So the assumption on E_0 reduces to the condition in Assumption 4.4.4 on the Hessian $H_\varepsilon(E_0)$.

4.5 Numerical integration by a splitting method

In this section we discuss how to integrate numerically the differential equations (4.13). The objective here is not to follow a particular trajectory accurately, but to arrive quickly at a stationary point, which corresponds to the sought solution of the optimization problem (4.2). The simplest method is the normalized Euler's method, or normalized gradient descent method, where the result after an Euler step (i.e. a steepest descent step) is normalized to unit norm for both the u- and v-component. This can be combined with a standard line search strategy to determine the stepsize adaptively. However, a more efficient approach is obtained with a splitting method instead of Euler's method.

4.5.1 Splitting method

The splitting method consists in dividing the right-hand sides of system (4.13) in two parts: the integration of a first step, that acts as an horizontal move, is applied to the equations

$$\begin{cases} \rho\dot{u} = \left(\alpha\overline{\beta}\gamma\right) u - \left(\overline{\beta}\gamma\right) x \\ \rho\dot{v} = \left(\overline{\alpha}\beta\overline{\gamma}\right) v - \left(\overline{\alpha\gamma}\right) y \end{cases}, \tag{4.15}$$

followed by a step for the differential equations

$$\begin{cases} \rho\dot{u} = -\dfrac{\mathrm{i}}{2}\ \mathrm{Im}(\alpha\overline{\beta}\gamma)u \\ \rho\dot{v} = +\dfrac{\mathrm{i}}{2}\ \mathrm{Im}(\alpha\overline{\beta}\gamma)v \end{cases}. \tag{4.16}$$

Since the right-hand sides of the equations in system (4.16) consists, respectively, of just an imaginary coefficient for u and v, the differential equations in the second step are a mere rotation of u and v. In the case of a real eigenvalue of a real matrix, the system (4.16) has a vanishing right-hand side and can therefore be ignored. The following result show that this approach preserves stationary points, which is unusual for splitting methods.

Lemma 4.5.1. *A pair of vectors* (u, v) *is a stationary point of system* (4.13) *if and only if* (u, v) *is a stationary point of systems* (4.15) *and* (4.16).

Proof. If (u, v) is a stationary point of (4.13), then u is proportional to x and v is proportional to y. This implies that $x = \alpha u$ and $y = \beta v$, by means of the definition of α and β and hence (u, v) is a stationary point of (4.15). Thus

$$0 = (\alpha\overline{\beta}\gamma)u - (\overline{\beta}\gamma)x - \frac{\mathrm{i}}{2}\,\mathrm{Im}(\alpha\overline{\beta}\gamma)u = -\frac{\mathrm{i}}{2}\,\mathrm{Im}(\alpha\overline{\beta}\gamma)u$$

and an analogous relation holds also for the second equation. Hence (u, v) is a stationary point also of (4.16). The converse direction is evident. □

4.5.2 Fully discrete splitting algorithm

Now we describe a numerical method for integrating systems (4.15) and (4.16). Starting from the pair of vectors (u_k, v_k) of unit norm, we define

$$\rho_k = \frac{1}{\|\Pi_{\mathcal{S}}(u_k v_k^*)\|_F},$$

we denote by x_k and y_k the left and right eigenvectors to the target eigenvalue λ_k of $A + \varepsilon\rho_k\Pi_{\mathcal{S}}(u_k v_k^*)$ and we set

$$\alpha_k = u_k^* x_k, \qquad \beta_k = v_k^* y_k, \qquad \gamma_k = 2 f_{\overline{\lambda}_k}. \tag{4.17}$$

We apply Euler's method with stepsize h to (4.15) to obtain

$$\begin{cases} \hat{u}(h) = u_k + \dfrac{h}{\rho_k}\Big(\left(\alpha_k\overline{\beta}_k\gamma_k\right) u_k - \left(\overline{\beta}_k\gamma_k\right) x_k\Big) \\ \hat{v}(h) = v_k + \dfrac{h}{\rho_k}\Big(\left(\overline{\alpha}_k\beta_k\overline{\gamma}_k\right) v_k - \left(\overline{\alpha_k\gamma_k}\right) y_k\Big) \end{cases}, \tag{4.18}$$

followed by a normalization to unit norm

$$\tilde{u}(h) = \frac{\hat{u}(h)}{\|\hat{u}(h)\|}, \qquad \tilde{v}(h) = \frac{\hat{v}(h)}{\|\hat{v}(h)\|}.$$

Then, as a second step, we integrate the rotating differential equations (4.16) by setting

$$u(h) = \mathrm{e}^{\mathrm{i}\vartheta h}\, \tilde{u}(h), \qquad v(h) = \mathrm{e}^{-\mathrm{i}\vartheta h}\, \tilde{v}(h), \qquad \rho(h) = \frac{1}{\|\Pi_{\mathcal{S}}(u(h)v(h)^*)\|_F}, \tag{4.19}$$

where

$$\vartheta = -\frac{1}{2\rho_k}\ \mathrm{Im}(\alpha_k \overline{\beta_k} \gamma_k)$$

and finally we compute the target eigenvalue $\lambda(h)$ of the perturbed matrix

$$A + \varepsilon\rho(h)\Pi_{\mathcal{S}}\big(u(h)v(h)^*\big).$$

The fully discrete method, which preserves the stationary points of the continuous system (4.13), is implemented by Algorithm 2 and in the following we comment on its main steps. One motivation for choosing this method is that near a *non real* stationary point, the motion is almost rotational since $x \approx \alpha u$ and $y \approx \beta v$. The dominant term determining the motion is then the rotational term on the right-hand sides of (4.13), which is integrated by a rotation in the above scheme (the integration would be exact if α, β and γ were constant). Algorithm 2 requires in each step one computation of target eigenvalues and associated eigenvectors of structure-projected rank-1 perturbations to the matrix A, which can be computed at moderate computational cost for large sparse matrices A by a Krylov Schur algorithm [64], as implemented in the MATLAB function `eigs`. We also tried a variant where, in the rotation step, α, β and γ are updated from $(\tilde{u}(h), \tilde{v}(h))$ and from the left and right eigenvectors associated to the target eigenvalue $\tilde{\lambda}(h)$ of $A + \varepsilon\rho(h)\Pi_{\mathcal{S}}(\tilde{u}(h)\tilde{v}(h)^*)$. In our numerical experiments we found, however, that the slight improvement in the speed of convergence to the stationary state does not justify the nearly doubled computational cost per step.

According to all our numerical experiments we have observed that if we start the integration from the initial datum (corresponding to the free gradient)

$$Y(0) = -2f_{\overline{\lambda}}xy^*, \tag{4.20}$$

where (λ, x, y) is the target eigentriplet of the matrix A, the functional F_ε turns out to be decreasing along solution of (4.11), which is consistent with the quasi-gradient local structure of the ODE, as discussed in the previous section (see Theorem 4.4.5). As a consequence we have designed a stepsize control strategy based on the assumption of monotonicity of F_ε. If this were not the case we would commute (for the non descent steps) to a standard stepsize control method for ODE solvers, based on standard error estimation of the solution. We use an Armijo-type line search strategy, adapted to the possibility that the functional $f(\lambda, \overline{\lambda})$ is not everywhere reduced along the flow of the differential equation (4.11) (even though this was never observed in our numerical experiments when we chose the initial value according to (4.20)). By Lemma 4.2.3, the change of the functional along solutions of (4.11) is (with $G = G_\varepsilon(E)$ for short)

$$\frac{\mathrm{d}}{\mathrm{d}t}F_\varepsilon(E(t)) = \varepsilon\kappa\, \mathrm{Re}\langle G, \dot{E}\rangle = -\varepsilon\kappa\Big(\|\Pi_{\mathcal{S}}G\|_F^2 - \mathrm{Re}\langle \Pi_{\mathcal{S}}P_Y G, E\rangle\ \mathrm{Re}\langle \Pi_{\mathcal{S}}G, E\rangle\Big) =: -g \tag{4.21}$$

For the choice $E = E_k = u_k v_k^*$, we write $g_k = g$, $G = G_\varepsilon(E_k) = 2f_{\overline{\lambda}}(\lambda_k, \overline{\lambda_k})x_k y_k^*$, and $\kappa = \kappa_k = 1/(x_k^* y_k)$. We set

$$f_k = f(\lambda_k, \overline{\lambda_k}), \qquad f(h) = f(\lambda(h), \overline{\lambda(h)})$$

Algorithm 2 Integration step for the rank-1 differential equations (4.13).

Input: A given matrix A, a perturbation size ε, a parameter $\theta > 1$, two starting vectors $u_k \approx u(t_k)$ and $v_k \approx v(t_k)$, a proposed stepsize h_k and a target eigenvalue λ_k of $A + \varepsilon \Pi_{\mathcal{S}}(u_k v_k^*)/\|\Pi_{\mathcal{S}}(u_k v_k^*)\|_F$.

Output: The updated variables u_{k+1}, v_{k+1}, h_{k+1} and λ_{k+1}.

```
Initialize the stepsize by the proposed one: $h = h_k$.
Compute left/right eigenvectors $x_k$ and $y_k$ of $A + \Delta_k$ to $\lambda_k$ that fulfils condition (4.5).
Compute $\alpha_k, \beta_k$ and $\gamma_k$ by (4.17) and $g_k$ by (4.21).
Initialize $f(h) = f_k$.
while $f(h) \geq \max(f_k, f_k - h\theta g_k)$ do
    Compute $(u(h), v(h))$ according to (4.18) and (4.19).
    Compute $\Delta(h) = \varepsilon\rho(h)\Pi_{\mathcal{S}}(u(h)v(h)^*)$ with $\rho(h) = 1/\|\Pi_{\mathcal{S}}(u(h)v(h)^*)\|_F$.
    Compute $\lambda(h)$ target eigenvalue of $A + \Delta(h)$.
    Compute the value $f(h) = f\big(\lambda(h), \overline{\lambda(h)}\big)$.
    if $f(h) \geq \max(f_k, f_k - h\theta g_k)$ then
        Reduce the stepsize by setting $h := h/\theta$
    end if
end while
if $\big(g_k \geq 0$ and $f(h) \geq f_k - (h/\theta)g_k\big)$ or $\big(g_k < 0$ and $f(h) \geq f_k - h\theta g_k\big)$ then
    Reduce the stepsize for the next step: $h_{\text{next}} := h/\theta$.
else
    if $h = h_k$ then
        Set $h_{\text{next}} := \theta h_k$ (augment the stepsize if no rejection has occurred).
    else
        Set $h_{\text{next}} := h_k$.
    end if
end if
Set $h_{k+1} := h_{\text{next}}$, $\lambda_{k+1} := \lambda(h)$ and define the starting values for the next step as $u_{k+1} := u(h)$ and $v_{k+1} := v(h)$.
```

and we accept the result of the step with stepsize h if, for a given parameter $\theta > 1$,

$$f(h) < \max(f_k, f_k - h\theta g_k).$$

If $g_k \geq 0$ and $f(h) \geq f_k - (h/\theta)g_k$, or if $g_k < 0$ and $f(h) \geq f_k - h\theta g_k$, then we reduce the stepsize for the next step to h/θ. If the stepsize has not been reduced in the previous step, we try for a larger stepsize. Algorithm 2 describes in detail the step from t_k to $t_{k+1} = t_k + h_k$ of the splitting method.

4.6 Application to structured matrix nearness problems

In this section we consider some matrix nearness problems that arise in a stability setting and that are closely related to the eigenvalue optimization method considered in this chapter. Given a structure space $\mathcal{S}$, let again $A \in \mathbb{C}^{n \times n}$ be a given matrix and let $\lambda(A) \in \mathbb{C}$ be a target eigenvalue of A. We again consider the smooth function $f(\lambda, \overline{\lambda})$ satisfying (4.1) that is to be minimized. For a prescribed real number $a_\star$ in the range of f we assume that

$$f(\lambda(A), \overline{\lambda}(A)) > a_\star,$$

so that, for sufficiently small $\varepsilon > 0$, we have $\varphi(\varepsilon) > a_\star$, where

$$\varphi(\varepsilon) := \min_{\Delta \in \mathcal{S},\ \|\Delta\|_F = \varepsilon} f\left(\lambda\left(A+\Delta\right), \overline{\lambda}\left(A+\Delta\right)\right).$$

The aim is to find the smallest $\varepsilon > 0$ such that $\varphi(\varepsilon) = a_\star$:

$$\varepsilon_\star = \min\bigl\{\varepsilon > 0 : \varphi(\varepsilon) \leq a_\star\bigr\}. \tag{4.22}$$

Determining $\varepsilon_\star$ is a one-dimensional root-finding problem for the function φ that is defined by the considered eigenvalue optimization problem.

4.6.1 Structured distances to singularity and to instability

Let us consider three problems which corresponds to three different target eigenvalues: the structured distance to instability, the stability radius of a Schur matrix and the structured distance to singularity.

- Let A be a Hurwitz matrix, i.e. with negative spectral abscissa $\alpha(A) < 0$, where

  $$\alpha(A) := \max\{\mathrm{Re}(\lambda) : \lambda \text{ is an eigenvalue of } A\}.$$

  With the function $f(\lambda, \overline{\lambda}) = -\frac{1}{2}(\lambda + \overline{\lambda}) = -\mathrm{Re}(\lambda)$, meaning that the target eigenvalue λ is the one with largest real part, and $a_\star = 0$, we arrive at the problem of computing the *structured distance to instability* of A, defined as

  $$\varepsilon_\star = \min\{\varepsilon > 0 : \alpha_\varepsilon^{\mathcal{S}}(A) = 0\},$$

  where

  $$\alpha_\varepsilon^{\mathcal{S}}(A) = \max_{E \in \mathcal{S},\ \|E\|_F = 1} \alpha(A + \varepsilon E)$$

  is the ε-pseudospectral abscissa with respect to the structure space $\mathcal{S}$.

- With $f(\lambda, \overline{\lambda}) = -\lambda\overline{\lambda} = -|\lambda|^2$, meaning that the target eigenvalue $\lambda(M)$ is chosen as an eigenvalue of largest modulus of a matrix M, and $a_\star = -1$ we arrive at the problem of computing the *stability radius of a Schur matrix* A, i.e. a matrix with spectral radius $\rho(A) < 1$, where

  $$\rho(A) := \max\{|\lambda| : \lambda \text{ is an eigenvalue of } A\}.$$

  The *stability radius* is defined as

  $$\varepsilon_\star = \min\{\varepsilon > 0 : \rho_\varepsilon^{\mathcal{S}}(A) = 1\},$$

  where

  $$\rho_\varepsilon^{\mathcal{S}}(A) = \max_{E \in \mathcal{S},\ \|E\|_F = 1} \rho(A + \varepsilon E)$$

  is the ε-pseudospectral radius with respect to the structure space $\mathcal{S}$.

- Let A be a nonsingular matrix. With $f(\lambda, \overline{\lambda}) = \lambda\overline{\lambda} = |\lambda|^2$, implying that the target eigenvalue λ the one of smallest modulus, we arrive at the problem of computing the *distance to singularity* of A, defined as

$$\varepsilon_\star = \min\{\varepsilon > 0 \,:\, \varrho_\varepsilon^{\mathcal{S}}(A) = 0\},$$

with

$$\varrho_\varepsilon^{\mathcal{S}}(A) = \min_{E\in\mathcal{S},\ \|E\|_F=1} \varrho(A+\varepsilon E),$$

where $\varrho(M)$ is the smallest modulus of eigenvalues of a matrix M. In this case, instead of eigenvalues of smallest modulus, we could take the smallest singular value in the objective functional: the resulting approach is similar to the one presented, but the role of the eigenvectors in the associated gradient is replaced by the singular vectors. A possible advantage is that the smallest singular value could have a better conditioning with respect to the eigenvalue with smallest modulus, for instance in the presence of a couple of conjugate eigenvalues that coalesce under real perturbations.

As seen in Chapter 2 and Chapter 3, to solve these kind of problems we use the following two-level method:

(i) *Inner iteration:* Given $\varepsilon > 0$, we aim to compute a matrix $E_\star(\varepsilon) \in \mathcal{S}$ of unit Frobenius norm, such that $F_\varepsilon(E) = f\left(\lambda\left(A+\varepsilon E\right), \overline{\lambda}\left(A+\varepsilon E\right)\right)$ is minimized:

$$E(\varepsilon) = \arg\min_{E\in\mathcal{S},\ \|E\|_F=1} F_\varepsilon(E). \tag{4.23}$$

(ii) *Outer iteration:* We compute the smallest positive value $\varepsilon_\star$ with

$$\varphi(\varepsilon_\star) = a_\star, \tag{4.24}$$

where

$$\varphi(\varepsilon) = F_\varepsilon\left(E(\varepsilon)\right) = f\left(\lambda\left(A+\varepsilon E(\varepsilon)\right), \overline{\lambda}\left(A+\varepsilon E(\varepsilon)\right)\right).$$

The eigenvalue optimization problem (4.23) is precisely of the type studied in the previous sections. To compute $E_\star(\varepsilon)$ for a given $\varepsilon > 0$, we integrate numerically either the ODE system (4.8) or (4.11); see Section 4.5. The computational cost can be significantly reduced if we are able to compute efficiently $\Pi_{\mathcal{S}}(Y)$ and the matrix vector multiplication $\Pi_{\mathcal{S}}(Y)v$ (with $v \in \mathbb{C}^n$) which is typically used by an iterative eigensolver applied to $A + \varepsilon\Pi_{\mathcal{S}}(Y)$. This is true for example when $\mathcal{S}$ is the set of matrices with a prescribed sparsity pattern. Note that often also linear system solves are required to find the desired eigenvalue and a convenient solution of the structured linear systems is desirable (see e.g. [63]).

The outer iteration determines the smallest positive solution of the one-dimensional root-finding problem (4.24). We make use of a locally quadratically convergent Newton-type method, which can be justified under appropriate regularity assumptions (see Section 2.4 and Section 3.2). It turns out that the derivative of φ is then simply (see Lemma 3.2.1)

$$\varphi'(\varepsilon) = -\|\Pi_{\mathcal{S}} G_\varepsilon(E(\varepsilon))\|_F / (x(\varepsilon)^* y(\varepsilon)), \tag{4.25}$$

where $x(\varepsilon)$ and $y(\varepsilon)$ with $x(\varepsilon)^* y(\varepsilon) > 0$ are the eigenvectors to the (simple) target eigenvalue $\lambda(\varepsilon)$ of $A + \varepsilon E(\varepsilon)$ at the extremizer $E(\varepsilon)$; cf. [24, 32] for related derivative formulas. If the assumptions justifying this formula are not met, we can always resort to bisection. The algorithm we use is indeed a combined Newton/bisection approach, similar to [24, 28, 32] and it is described in Algorithm 1.

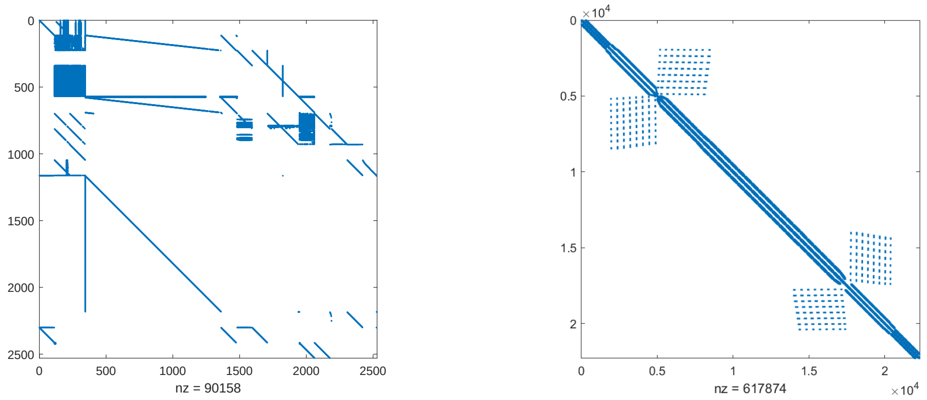


Figure 4.2: Sparsity patterns of the matrices ORANI678 (left) and FIDAPM11 (right).

k	$\mathrm{Re}(\lambda_k)$
0	-1.232670912085709
1	-1.745212357950066
2	-1.917229680782718
3	$-\mathbf{2}.076407232182272$
4	$-\mathbf{2}.249359154133923$
5	$-\mathbf{2.3}43018078428841$
6	$-\mathbf{2.3}43036033336665$
7	$-\mathbf{2.3}49611649664635$
8	$-\mathbf{2.350}556073486847$
9	$-\mathbf{2.3506}20017092603$
$\vdots$	$\vdots$
25	$-\mathbf{2.35063477526}2768$

Table 4.1: Computed values using Algorithm 2 for the ORANI678 matrix.

4.7 Numerical experiments

In this section we show the behaviour of Algorithm 2, which is based on the rank-1 differential equation (4.11), on two sparse matrices and an example with prescribed range and co-range. We start by considering two well-known sparse matrices.

4.7.1 The matrix ORANI678 from the Harwell Boeing collection

The matrix A is a sparse real nonsymmetric square matrix taken from the set ECONAUS. It has dimension $n = 2529$ and a number of non-zero entries $nz = 90158 \approx 40n$. Its sparsity pattern is plotted in Figure 4.2.

We have set $\varepsilon = 1$ and applied our algorithms to the minimization problem (4.3) with $f(\lambda, \overline{\lambda}) = -\frac{1}{2}(\lambda + \overline{\lambda}) = -\mathrm{Re}(\lambda)$ and $\mathcal{S}$ the space of real matrices with the sparsity pattern of A. The target eigenvalue is the one with largest real part. We thus aim to compute the structured ε-pseudospectral abscissa of A with an accuracy of 14 digits.

We denote by n_{eig} the total number of eigenvalue computations (that is the number of calls to the Matlab routine eigs). We integrated (4.11) by Algorithm 2 and obtained the results in Table 4.1. The main cost is the number of eigentriplets evaluations by the Matlab routine `eigs` and it is given by $n_{\mathrm{eig}} = 38$. The CPU time is around 1.5 seconds. For comparison we also integrated the full-rank ODE (4.8) by Euler's

k	ε_k	$\varphi(\varepsilon_k)$	# eigs
1	**0.0**104015	$1.1019564 \cdot 10^{-2}$	13
2	**0.0**176409	$9.5284061 \cdot 10^{-4}$	13
3	**0.02**19541	$2.5263758 \cdot 10^{-4}$	14
4	**0.02**43116	$6.5050153 \cdot 10^{-5}$	13
5	**0.02**55439	$1.6503282 \cdot 10^{-6}$	13
6	**0.026**1739	$4.1561289 \cdot 10^{-6}$	13
7	**0.026**4923	$1.0428313 \cdot 10^{-6}$	13
8	**0.0266**524	$2.6118300 \cdot 10^{-7}$	13
9	**0.0267**327	$6.5355110 \cdot 10^{-8}$	13
10	**0.0267**728	$9.6346192 \cdot 10^{-9}$	13
11	**0.0267930**	$1.7293467 \cdot 10^{-10}$	4

Table 4.2: Distance to singularity for the ORANI678 matrix: computed values ε_k, $\varphi(\varepsilon_k) = |\lambda_{\min}(A + \varepsilon_k E_k)|^2$ and number of eigenvalue computations of the inner rank-1 algorithm.

method (gradient descent) with variable stepsize and we obtained a similar behaviour. The number of eigentriplets evaluations is $n_{\text{eig}} = 35$ and the final approximation to the ε-pseudospectral abscissa is 2.350634775261177, which coincides with the value computed by the rank-1 method up to the 11-th digit. The CPU time is 1.6 seconds. Since u and v turn out to be real, the gain in terms of memory requirements for the rank-1 algorithm is $90158/5058 \approx 17.82$, which is a significant reduction in the storage of the iterates.

Setting next $f(\lambda, \overline{\lambda}) = \lambda\overline{\lambda} = |\lambda|^2$ and the target eigenvalue the one - say $\lambda_{\min}$ - with smallest modulus, we approximated the structured distance to singularity of A. Given the convergence to a local optimizer of Algorithm 2 we obtain this way an upper bound to this distance. An immediate lower bound is the unstructured distance $\sigma_{\min}(A)$, i.e. the smallest singular value, which is equal to 0.0033388. As we see in Table 4.2, the effective structured distance to singularity is one order of magnitude larger. By applying a Newton-bisection method we obtained the results shown in Table 4.2. Since the function φ and its derivative (see (4.25)) are computed inexactly (by Algorithm 2), we do not observe quadratic convergence. The average CPU time of an outer iteration is around 528.6 seconds, which is due to the augmented computational cost required by the routine eigs for linear systems solves and it is also motivated by the high accuracy requested. The average number of eigentriplets evaluation is $n_{\text{eig}} = 12$.

4.7.2 The matrix FIDAPM11 from the SPARSKIT collection

The FIDAPM11 matrix A considered now is a sparse real nonsymmetric square matrix taken from the set ECONAUS. It has dimension $n = 22294$ and a number of nonzero entries $nz = 623554 \approx 30n$. Its sparsity pattern is plotted in Figure 4.2. We have set $\varepsilon = 0.5$ and applied our algorithms to the minimization problem (4.3) with $f(\lambda, \overline{\lambda}) = -\lambda\overline{\lambda} = -|\lambda|^2$ and $\mathcal{S}$ the space of real matrices with the sparsity pattern of A, and the target eigenvalue is the one with largest absolute value. We are thus aiming to compute the structured ε-pseudospectral radius of A. Integrating both ODEs (4.8) and (4.11), we obtain the same optimizer $\lambda = 1.9716893$. The number of computed eigentriplets is $n_{\text{eig}} = 107$ and $n_{\text{eig}} = 99$, with a slight advantage of the rank-1 method. The CPU time is close to 32.95 and 31.32 seconds respectively. Also in this case u and v turn out to be real so that the gain in terms of memory requirements is significant, $623554/44588 \approx 13.98$.

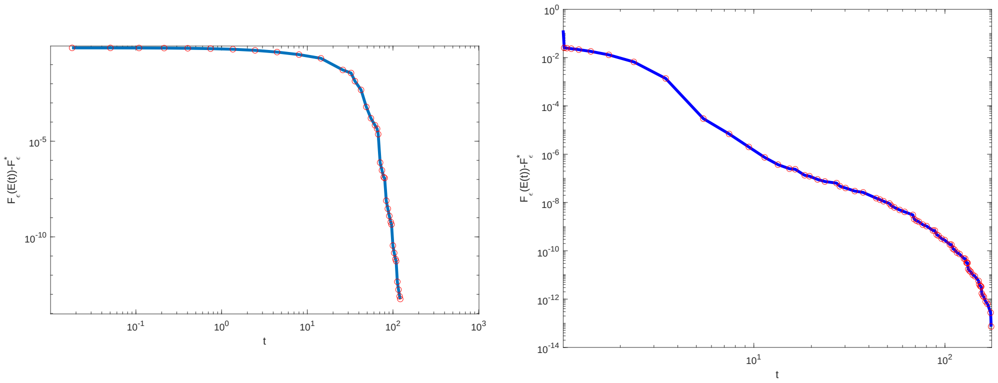


FIGURE 4.3: behaviour $F_\varepsilon\left(E(t)\right) - F_\varepsilon^*$ (where F_ε^* is the computed value of F_ε at the stationary point) in the numerical integration by Algorithm 2 for the matrix ORANI678 with $f(\lambda, \overline{\lambda}) = -\operatorname{Re}(\lambda)$ (left picture) and for the matrix FIDAPM11 with $f(\lambda, \overline{\lambda}) = -|\lambda|^2$ (right picture). In both cases $F_\varepsilon\left(E(t_k)\right)$ decays monotonically with k.

4.7.3 A comparison with Manopt

In order to compare our method, we made experiments on the sparse matrices considered above using Manopt, a well-known toolbox for optimization on manifolds and matrices [6]. By applying Manopt to the same problem considered for the ORANI678 matrix, where we provided the Riemannian gradient on the manifold of sparse matrices with unit Frobenius norm, the method yields a result very close to the one computed with our method (the difference is around 10^{-13}). The CPU time for our algorithm is approximately 1.4 seconds. Concerning the algorithm implemented in Manopt, with the conjugate-gradient method, we have found that the method converges in 20 iterations using a CPU time which is approximately 14 seconds; with the BFGS solver it converges in 15 iterations using a CPU time of approximately 16 seconds and finally with the Barzilai-Borwein method it converges in 117 iterations in about 57 seconds. The trust region method (which is the default choice) instead turns out to converge very slowly.

Then we applied Manopt, with the conjugate-gradient solver, to the considered example with the FIDAPM11 matrix and we obtained that the result coincides with the one we obtain to 5 digits. However the CPU time exceeds 5 hours when the default accuracy is used, but it drops to around 8 minutes when a tolerance of 10^{-2} is required, which still gives 3 correct digits. With the option of adaptive line search and the same tolerance, a result with the same accuracy is obtained in a slightly larger CPU time.

4.7.4 An example of control of the Stokes problem

We consider an example from [41], which arises in the discretization of the 2-dimensional Stokes problem on a uniform quadratic grid. Setting 25 grid points on both sides of the square, we get a sparse matrix A ($J - R$ in the notation of [41]) which has dimension $n = 1824$, while we choose the control matrices B and $C = B^\top$ to have size $n \times k$ and $l \times n$, respectively with $k = l = 40$, with randomly i.i.d. entries and unit Frobenius norm. The matrix A has the rightmost eigenvalue $\lambda = -6.4343098 \cdot 10^{-4}$, which suggests a non-robust Hurwitz stability. Although Assumption 4.4.2 is not fulfilled we successfully execute the rank-1 algorithm. Running it on this example, we find the structured stability radius to be 0.0384039, which is 60 times larger than $|\lambda|$.

k	ε_k	$\varphi(\varepsilon_k)$	# eigs
0	0	$-6.4343098 \cdot 10^{-4}$	1
1	0.02	$-3.1062242 \cdot 10^{-4}$	23
2	0.0385299	$2.1414201 \cdot 10^{-6}$	26
3	0.0384039	$9.7779625 \cdot 10^{-11}$	18

Table 4.3: Iterates for computing the *structured stability radius* for the Stokes problem matrix with range- and co-range-constrained perturbations.

Since the matrix is sparse we can exploit favourably the matrix vector products of the form (with $p \in \mathbb{R}^n$ the vector, $Z = uv^* \in \mathcal{M}_1$, and ρ the normalization factor)

$$\left(A + \varepsilon\rho B\, B^\dagger Z C^\dagger C\right) p,$$

whose cost is linear in n. In Table 4.3 we show the Newton iteration where the number of eigentriplets evaluation is again indicated by n_{eig}. The quadratically convergent behaviour is evident.

4.8 Perturbation matrices of prescribed range and co-range

In this final section, we consider the (complex) structure space $\mathcal{S}$ of (4.14), which only allows for perturbations of given range and co-range. We recall that the orthogonal projection onto $\mathcal{S}$ is given by $\Pi_{\mathcal{S}} Z = BB^\dagger Z C^\dagger C$ (see Proposition B.0.5). In this case, the set $\mathcal{M}_1^{\mathcal{S}} = \Pi_{\mathcal{S}}\mathcal{M}_1$ of structure-projected rank-1 matrices equals the submanifold of rank-1 matrices that have the prescribed range and co-range:

$$\mathcal{M}_1^{\mathcal{S}} = \{E \in \mathbb{C}^{n\times n} \,:\, E = \rho uv^* \ \text{ with } \ \rho > 0,\ u \in \operatorname{Ran}(B),\ v \in \operatorname{Ran}(C)\} \subseteq \mathcal{M}_1.$$

For such an $E = \rho uv^*$ with u and v of unit norm and in the range of B and C, respectively, the orthogonal projection $P_E^{\mathcal{S}}$ onto the tangent space $\mathcal{T}_E\mathcal{M}_1^{\mathcal{S}}$ turns out to be given by the same expression as in (4.12):

$$P_E^{\mathcal{S}}(Z) = Z - (I - uu^*)Z(I - vv^*).$$

This has important consequences. On the theoretical side, it allows us to use the same argument as in the proof of part (b) of Theorem 4.3.1 to show that every stationary point of the gradient system (4.9) on $\mathcal{M}_1^{\mathcal{S}}$ is also a stationary point of the gradient system (4.8) on $\mathcal{S}$; hence, there are no spurious stationary points. On the computational side, for a solution $E(t) = u(t)v(t)^* \in \mathcal{M}_1^{\mathcal{S}}$ of unit Frobenius norm of the differential equation (4.9) we therefore obtain differential equations for the factors u and v of unit norm that are formally the same as in Lemma 4.3.2: with the projected gradient $G^{\mathcal{S}} = \Pi_{\mathcal{S}} G(E)$ for short,

$$\begin{cases} \dot u = -(I - uu^*)G^{\mathcal{S}} v - \dfrac{\mathrm{i}}{2}\operatorname{Im}(u^* G^{\mathcal{S}} v)u, \\ \dot v = -(I - vv^*)(G^{\mathcal{S}})^* u + \dfrac{\mathrm{i}}{2}\operatorname{Im}(u^* G^{\mathcal{S}} v)v \end{cases}.$$

Note that here $\dot u$ and $\dot v$ are in the range of B and C, respectively, so that u and v stay in these ranges. In order to obtain a further compression we set $u = Bp$ and

k	ε_k	$\phi(\varepsilon_k)$	# eigs
1	0.02	$-3.1061082 \cdot 10^{-4}$	26
2	0.0386113	$4.3234455 \cdot 10^{-5}$	33
3	0.0384036	$8.2212343 \cdot 10^{-10}$	16

Table 4.4: Iterates for computing the structured stability radius for the Stokes problem matrix with range- and co-range-constrained perturbations with inner iteration realized integrating (4.26).

$v = C^*q$ with $p \in \mathbb{C}^k$ and $q \in \mathbb{C}^l$. In this way - with $G = G_\varepsilon(E)$ - we obtain for $E = uv^* = Bpq^*C$ the differential equations

$$\begin{cases} \dot{p} = -B^\dagger GC^*q \ + \ pp^*B^*GC^*q - \dfrac{\mathrm{i}}{2}\,\mathrm{Im}(p^*B^*GC^*q)p \\ \dot{q} = -(C^*)^\dagger G^*Bp + qq^*CG^*Bp + \dfrac{\mathrm{i}}{2}\,\mathrm{Im}(p^*B^*GC^*q)q \end{cases} . \tag{4.26}$$

With the rank-1 matrix $G = G_\varepsilon(E) = 2f_{\overline{\lambda}}xy^*$ (see Lemma 4.2.3) and with $\alpha = p^*B^*x$, $\beta = q^*Cy$ and $\gamma = 2f_{\overline{\lambda}}$, we thus obtain the differential equations (cf. (4.13))

$$\begin{cases} \dot{p} = \left(\alpha\overline{\beta}\gamma\right)p - \left(\overline{\beta}\gamma B^\dagger\right)x - \left(\dfrac{\mathrm{i}}{2}\,\mathrm{Im}(\alpha\overline{\beta}\gamma)\right)p \\ \dot{q} = \left(\overline{\alpha}\beta\overline{\gamma}\right)q - \left(\overline{\alpha}\overline{\gamma}(C^*)^\dagger\right)y - \left(\dfrac{\mathrm{i}}{2}\,\mathrm{Im}(\overline{\alpha}\beta\overline{\gamma})\right)q. \end{cases}$$

This system of differential equations is treated numerically in the same way as described in Section 4.5, using a splitting between the first two terms on the right-hand side and the third term. We present numerical results for the Stokes example of Section 4.7.4, now treated with the above implementation of the gradient system (4.9) for comparison.

Chapter 5

Spectral clustering robustness

In this chapter we aim to adapt the structured two-level method introduced in Chapter 3 in a graph theory setting, making it a three-level method. The problem is the following: given an undirected weighted graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}, W)$ with n vertices $\mathscr{V}$, edges $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$ and weights described by the non-negative symmetric matrix $W \in \mathbb{R}^{n \times n}$, we look for the best integer k for partitioning $\mathscr{G}$ into k clusters. More precisely, given $2 \leq k_{\min} < k_{\max} \leq n-1$, we seek an integer $k \in \{k_{\min}, \dots, k_{\max}\}$ for performing the spectral clustering algorithm on $\mathscr{G}$ such that the partitioning reflects the graph's main features and such that the clustering is stable under small perturbations. In order to do so, we introduce a new measure for the robustness of this method which is more appropriate with respect to the state-of-the-art spectral gaps, since it also takes into account the sparsity pattern of W. The chapter is mainly based on the article [37] and it is an extension of the results from [3] and [34].

5.1 Introduction

Clustering is the task of dividing a data set into k communities such that members in the same groups are related. It is an unsupervised method in machine learning that discovers data groupings without the need of human intervention and its aim is to gain important insights from collected data. Spectral clustering (originating with Fiedler [19]) is a type of clustering that makes use of the Laplacian matrix of an undirected weighted graph to cluster its vertices into k clusters. More precisely it performs a dimensionality reduction of the dataset and then it clusters in lower dimension.

The stability of this procedure is often associated with the spectral gap g_k, i.e. the difference between the $(k+1)$-st and k-th eigenvalues of the Laplacian. When g_k is not large, usually small perturbations may cause a coalescence of the two consecutive eigenvalues and they could significantly change the clustering. Thus, according to the spectral gaps criterion, a suitable number of clusters is the index of the largest spectral gap. This choice is also motivated by the fact that spectral gaps can be seen as an unstructured measure to ambiguity. In fact, up to a constant factor, g_k represents the minimum Frobenius norm of the difference between the Laplacian and a symmetric matrix with coalescing k-th and $(k+1)$-st eigenvalues (see Theorem 5.2.2). Since the computation of the spectral gaps is not expensive it is widely used with the aim of identifying an optimal index k.

In this chapter we introduce a structured measure to stability that takes into account the preservation of the pattern of the weight matrix of the graph. In this way it is possible to achieve a result that is more appropriate than the one provided by the spectral gaps criterion. The distance considered here is similar to the one presented in [3], but it makes use of a different metric.

The main objective of the chapter is to describe in detail how to determine the new criterion for spectral clustering stability and how to exploit the underlying low-rank structure of extremizers. We propose to compute the structured distance to ambiguity via a three-level approach, similar to the two-level approach of [28, 32], which is divided in a *structured inner iteration*, a *structured outer iteration* and then a *selection of* k, i.e. the best value for the number of clusters chosen in the given set $\{k_{\min}, \dots, k_{\max}\}$.

As already mentioned in Chapter 3, the *structured inner iteration* is the part of the algorithm that requires more effort: it consists in the solution of a non-convex structured eigenvalue optimization problem whose extremizers are seen as stationary points of a system of matrix ODEs with size depending on the structural pattern of the weight matrix of the graph. Then, by generalizing the approach of [34], we define a rank-4 symmetric ODE whose stationary points are the same as those of the full-rank system and we integrate it on the rank-4 manifold until it converges to a stationary point. When the $n \times n$ weight matrix has a number of non-zeros (significantly) higher than $4n$, then integrating the rank-4 ODE turns out to be more convenient due to the (significantly) lower memory requirements.

The chapter is organized as follows. In Section 5.2 we briefly describe the spectral clustering method and we illustrate how to measure its robustness by the introduction of a structured distance to ambiguity. In Section 5.3 we discuss how to solve the *structured inner iteration* by means of a structured matrix ODE that is a gradient system. In Section 5.4 we exploit the underlying low-rank structure of the gradient system to project it and formulate a similar low-rank ODE that is used to solve the *structured inner iteration.* In Section 5.5 we describe the *structured outer iteration.* Finally in Section 5.6 we present the numerical results of the algorithm in a few graphs with different features.

5.2 Distance to ambiguity for spectral clustering

Consider an undirected weighted graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}, W)$, with vertices $\mathscr{V} = \{1, \dots, n\}$, edges $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$ and non-negative weight matrix $W = (w_{i,j}) \in \mathbb{R}^{n \times n}$. Its Laplacian matrix is defined as

$$L = L(W) = \operatorname{diag}(W\mathbb{1}) - W, \qquad \mathbb{1} = (1, \dots, 1)^\top.$$

It is well-known that L is a symmetric and positive semi-definite matrix. Indeed, since the graph is undirected, $w_{ij} = w_{ji}$ for all $i, j \in \{1, \dots, n\}$ and, for all $v = (v_i) \in \mathbb{R}^n$, we have

$$v^\top L v = \frac{1}{2}\sum_{i=1}^n v_i^2 \sum_{j=1}^n w_{i,j} + \frac{1}{2}\sum_{j=1}^n v_j^2 \sum_{i=1}^n w_{i,j} - \sum_{i=1}^n\sum_{j=1}^n w_{i,j} v_i v_j = \frac{1}{2}\sum_{i,j=1}^n (v_i - v_j)^2 w_{i,j} \geq 0. \tag{5.1}$$

Thus the spectral theorem ensures that the eigenvalues of L

$$\lambda_n \geq \dots \geq \lambda_2 \geq \lambda_1 = 0$$

are real non-negative and that their associated unit eigenvectors $x_n, \dots, x_1$ form an orthonormal basis of $\mathbb{R}^n$. The following result (see e.g. [68]) gives the theoretical reason behind the spectral clustering algorithm, which is shown in Algorithm 3. We denote the indicator vector $\mathbb{1}_{C_i}$ associated to the set $C_i \subseteq \mathscr{V}$ as the vector whose j-th entry is 1 if $j \in C_i$ and 0 otherwise.

Theorem 5.2.1. *Let $W \in \mathbb{R}^{n \times n}$ be the weight matrix of an undirected weighted graph $\mathscr{G}$ and denote by $L(W)$ its Laplacian. Then the number of the connected components $C_1, \dots, C_k$ of the graph equals the dimension of the kernel of $L(W)$.*

Moreover the eigenspace associated with the eigenvalue 0 *is spanned by the indicator vectors* $\mathbb{1}_{C_1}, \dots, \mathbb{1}_{C_k}$.

Algorithm 3 Unnormalized spectral clustering

Input: An undirected weighted graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}, W)$ and the number of clusters k

Output: Clusters $C_1, \dots, C_k$ that form a partition of $\mathscr{V}$

```
Find the $k$ smallest eigenvalues $0 = \lambda_1 \le \cdots \le \lambda_k$ of $L(W)$ and denote the associated normalized eigenvectors by $x_1, \dots, x_k \in \mathbb{R}^n$.
Build
    $$X = \left( \begin{array}{c|c|c|c} x_1 & x_2 & \cdots & x_k \end{array} \right) := \left( \begin{array}{c} r_1 \\ \hline r_2 \\ \hline \vdots \\ \hline r_n \end{array} \right).$$
Associate the $i$-th row $r_i$ of $X$ with the $i$-th vertex of the graph.
Cluster the points $r_1, \dots, r_n \in \mathbb{R}^k$ by the $k$-means algorithm (see e.g. [40]) into $k$ clusters $C_1, \dots, C_k$.
```

Spectral gaps provide a criterion to identify a reasonable value of the number of clusters k. The k-th spectral gap is characterized as the unstructured distance between the Laplacian and the set of symmetric matrices with coalescing eigenvalues λ_k and λ_{k+1}, as stated in the following result.

Theorem 5.2.2. *The k-th spectral gap $g_k = \lambda_{k+1} - \lambda_k$ of $L(W)$ is characterized as*

$$\frac{g_k}{\sqrt{2}} = \min \left\{ \|L(W) - \widehat{L}\|_F : \widehat{L} \in \operatorname{sym}\left(\mathbb{R}^{n\times n}\right),\ \lambda_{k+1}(\widehat{L}) = \lambda_k(\widehat{L}) \right\},$$

where $\operatorname{sym}(\mathbb{R}^{n\times n})$ *denotes the set of the symmetric real matrices.*

Proof. We follow the approach used in [3, Theorem 3.1], where we first exhibit a matrix $\widehat{L}$ such that $g_k = \sqrt{2}\|L(W) - \widehat{L}\|$ and then we prove that this matrix is a minimizer. Let $L := L(W) = Q\Lambda Q^\top$ be a spectral decomposition with the eigenvalues $\lambda_i = \Lambda_{i,i}$ in descending order and define $\widehat{L} = Q\widehat{\Lambda}Q^\top$, where $\lambda_i = \hat{\lambda}_i := \widehat{\Lambda}_{i,i}$ for all $i \notin \{k, k+1\}$ and $\hat{\lambda}_k = \hat{\lambda}_{k+1} = \frac{\lambda_{k+1} - \lambda_k}{2}$. Then, since the Frobenius norm is unitarily invariant,

$$\|L - \widehat{L}\|_F^2 = \|\Lambda - \widehat{\Lambda}\|_F^2 = (\lambda_{k+1} - \hat{\lambda}_{k+1})^2 + (\lambda_k - \hat{\lambda}_k)^2 = \frac{\lambda_{k+1} - \lambda_k}{2}.$$

In order to show that $\frac{g_k}{\sqrt{2}}$ is the minimum value, we make use of the Hoffman-Wielandt theorem (see [48, Theorem 6.3.5 and Corollary 6.3.8]) that, for all symmetric matrices L and $\widehat{L}$ with eigenvalues λ_i and $\hat{\lambda}_i$ in descending order, states that

$$\sum_{i=1}^{n} (\lambda_i - \hat{\lambda}_i)^2 \le \|L - \widehat{L}\|_F^2.$$

Let $\widehat{L}$ be an arbitrary symmetric matrix with coalescing eigenvalues $\hat{\lambda}_{k+1} = \hat{\lambda}_k$. Then

$$\frac{\lambda_{k+1} - \lambda_k}{2} = \min_{\lambda \in \mathbb{R}} \left((\lambda_{k+1} - \lambda)^2 + (\lambda_k - \lambda)^2 \right) \le$$

$$\leq (\lambda_{k+1} - \hat{\lambda}_{k+1})^2 + (\lambda_k - \hat{\lambda}_k)^2 \leq \sum_{i=1}^{n} (\lambda_i - \hat{\lambda}_i)^2$$

and the Hoffman-Wielandt theorem proves the claim. □

However, in the minimization problem of Theorem 5.2.2, the optimizer is a symmetric real matrix that in general is not a graph Laplacian, making this unstructured measure associated with the spectral gaps not completely reliable. This motivates us to introduce a new stability measure that takes into account the structure of the weight matrix W, that is described by the sets

$$\mathcal{S} = \left\{ A = (a_{ij}) \in \mathbb{R}^{n\times n} : \ a_{ij} = 0 \quad \forall (i,j) \notin \mathscr{E} \right\} \qquad \text{and} \qquad \mathcal{E} = \mathcal{S} \cap \operatorname{sym}(\mathbb{R}^{n\times n}).$$

We define the optimization problem

$$\Delta_\star^{(k)} = \underset{\Delta\in\mathcal{D}}{\arg\min} \left\{ \|\Delta\|_F : \ \lambda_k(L(W+\Delta)) = \lambda_{k+1}(L(W+\Delta)) \right\}, \tag{5.2}$$

where

$$\mathcal{D} = \{\Delta \in \mathcal{E} : \ W + \Delta \geq 0 \text{ entrywise}\}$$

is the set of all admissible perturbation that added to the weight matrix W return a matrix with non-negative entries and with the same structure of W. The minimum of (5.2)

$$d_k(W) = \|\Delta_\star^{(k)}\|_F$$

defines the k-th structured distance to ambiguity between W and $W_\star^{(k)} := W + \Delta_\star^{(k)}$. This new distance considered is similar to the one defined in [3], but it concerns a different geometry: in this framework we work with the Frobenius norm of the perturbation Δ, instead of considering a unit normalization of $L(\Delta)$. The reason behind this new choice is that in this way it is possible to exploit the underlying low-rank properties of the problem by the introduction of a rank-4 symmetric ODE. Instead it is less evident how to take advantage of these low-rank features for the distance introduced in [3].

In the following we denote the unit Frobenius norm sphere by

$$\mathbb{S}_1 = \left\{ A \in \mathbb{R}^{n\times n} : \|A\|_F = 1 \right\}$$

and we introduce the sets

$$\mathcal{S}_1 = \mathcal{S} \cap \mathbb{S}_1, \qquad \mathcal{E}_1 = \mathcal{E} \cap \mathbb{S}_1, \qquad \mathcal{D}_1 = \mathcal{D} \cap \mathbb{S}_1.$$

The approach presented in this chapter consists of a three-level procedure, whose first two steps are analogous to the method presented in Chapter 2:

- *Structured inner iteration:* Given a perturbation size $\varepsilon > 0$, we consider the non-negative objective functional

  $$F_\varepsilon^{(k)}(E) = \lambda_{k+1}\left(L(W+\varepsilon E)\right) - \lambda_k\left(L(W+\varepsilon E)\right),$$

  where we have written the perturbation $\Delta = \varepsilon E$ with $\|E\|_F = 1$. We look for a minimizer, which we denote as $E_\star^{(k)}(\varepsilon)$, of the optimization problem

  $$\underset{E\in\mathcal{D}_1}{\arg\min} \ F_\varepsilon^{(k)}(E). \tag{5.3}$$

- *Structured outer iteration:* We tune the parameter ε to obtain the smallest value $\varepsilon_\star^{(k)}$ of the perturbation size such that the objective functional evaluated in the minimizer vanishes, that is
$$F_{\varepsilon_\star}^{(k)}\left(E_\star^{(k)}(\varepsilon_\star^{(k)})\right)=0.$$
Then the approximated closest weighted adjacency matrix with coalescing eigenvalues k and $k+1$ would be $W_\star^{(k)}=W+\varepsilon_\star^{(k)}E_\star^{(k)}(\varepsilon_\star^{(k)})$.

- *Choice of k:* We repeat the procedure for all the values of $k\in\{k_{\min},\dots,k_{\max}\}$ and then select
$$k_{\mathrm{opt}}(W)=\underset{k_{\min}\le k\le k_{\max}}{\arg\max}\ \varepsilon_\star^{(k)}.$$

The objective functional introduced for this problem is a particular case of the structured version of
$$\mathscr{F}(\varepsilon E)=f\left(\mathscr{H}(\varepsilon E),\overline{\mathscr{H}(\varepsilon E)}\right)$$
described in (2.4). Indeed we have selected $a_\star=0$ and

- $f(z,\overline{z})=\mathrm{Re}(z)$, which in this case acts like the identity since the eigenvalues of a symmetric matrix are real,

- $\mathscr{H}(\varepsilon E)=\lambda_{k+1}\left(L(W+\varepsilon E)\right)-\lambda_k\left(L(W+\varepsilon E)\right)$, where the sum in (2.7) is actually made up by just two addends.

5.3 Constrained gradient system for the inner iteration

In this section we describe how to apply the ODE based approach to solve the optimization problem (5.3) defined in the *structured inner iteration* for problem (5.2). We will consider as fixed parameters the perturbation size $\varepsilon>0$ and a positive integer $k\in\{k_{\min},\dots,k_{\max}\}$. We introduce a matrix path $E(t)\subseteq\mathcal{E}_1$ that represents the normalized perturbation of the weight matrix W. As done in Chapter 3, we look for a time derivative of the objective functional $F_\varepsilon^{(k)}(E(t))$. Whenever there is no ambiguity, we avoid to write the dependence of k, for instance we denote the objective functional in short as $F_\varepsilon(E(t))$. For computing an explicit formula, we make use of the orthogonal projection $\Pi_{\mathcal{E}}$ with respect to the Frobenius inner product onto the pattern $\mathcal{E}$, whose expression for all $V=(v_{i,j})\in\mathbb{R}^{n\times n}$ is
$$(\Pi_{\mathcal{E}}(V))_{i,j}=\begin{cases}\dfrac{v_{i,j}+v_{j,i}}{2} & \text{if } (i,j)\in\mathscr{E}\\ 0 & \text{otherwise}\end{cases}.$$
Indeed it is easy to show the definition of orthogonal projection (see Definition B.0.1)
$$\langle\Pi_{\mathcal{E}}(V),W\rangle=\langle V,W\rangle,\qquad\forall V\in\mathbb{R}^{n\times n},\quad\forall W\in\mathcal{E}.$$
In this way we are able to introduce the adjoint L^* of the Laplacian operator with respect to the Frobenius inner product, which satisfies the formula (see Proposition D.0.5)
$$L^*(V)=\Pi_{\mathcal{E}}(\mathrm{diagvec}(V)\mathbb{1}^\top-V),\qquad\forall V\in\mathbb{R}^{n\times n},\tag{5.4}$$
where $\mathrm{diagvec}(V)\in\mathbb{R}^n$ is the vector of the diagonal entries of V. Then Lemma 2.3.1 can be adapted to this setting as shown in the following result (see e.g. [3, 37]), where

we make use of the componentwise product denoted by the symbol $\bullet$ such that

$$(x \bullet y)_i := x_i y_i, \qquad \forall x, y \in \mathbb{R}^n, \quad \forall i \in \{1, \dots, n\}.$$

Lemma 5.3.1. *Let $E(t)$ be a differentiable path of matrices in $\mathcal{E}_1$ for $t \in [0, +\infty)$. Assume that, for a given $\varepsilon > 0$, the eigenvalues $\lambda(t) = \lambda_{k+1}(L(W + \varepsilon E(t)))$ and $\mu(t) = \lambda_k(L(W + \varepsilon E(t)))$ are simple for all t. Let $x(t)$ and $y(t)$ be the normalized eigenvectors associated with $\lambda(t)$ and $\mu(t)$. Then*

$$\frac{1}{\varepsilon}\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) = \langle G_\varepsilon(E(t)), \dot{E}(t) \rangle,$$

where

$$G = G_\varepsilon(E(t)) := L^*\left(xx^\top - yy^\top\right) = \Pi_{\mathcal{E}}\left((x \bullet x - y \bullet y)\mathbb{1}^\top - (xx^\top - yy^\top)\right)$$

is the rescaled gradient of the objective functional $F_\varepsilon(E(t))$.

Proof. We proceed similarly as shown in Lemma 2.3.1 and Lemma 3.1.1, where the result is formulated for the generic objective functional $\mathscr{F}(\Delta) = f\left(\mathscr{H}(\Delta), \overline{\mathscr{H}(\Delta)}\right)$, with $\Delta = \varepsilon E$. The framework of this chapter yields that the setting is real, meaning that $\mathscr{H}(\varepsilon E) = \lambda_{k+1}\left(L(W + \varepsilon E)\right) - \lambda_k\left(L(W + \varepsilon E)\right) := \lambda\left(L(W + \varepsilon E)\right) - \mu\left(L(W + \varepsilon E)\right)$ and $f(z, \overline{z}) = \mathrm{Re}(z)$. Thus Lemma 4.2.1 yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathscr{H}(\varepsilon E(t)) = x^\top L(\dot{E})x - y^\top L(\dot{E})y,$$

and hence the derivative formula for F_ε turns into

$$\frac{1}{\varepsilon}\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) = x^\top L(\dot{E})x - y^\top L(\dot{E})y = \langle xx^\top - yy^\top, L(\dot{E})\rangle.$$

In order to get an expression analogous to that of Lemma 2.3.1 and Lemma 3.1.1, we use the definition of L^* and we get the claim

$$\frac{1}{\varepsilon}\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) = \langle L^*(xx^\top - yy^\top), \dot{E}\rangle = \left\langle \Pi_{\mathcal{E}}\left((x \bullet x - y \bullet y)\mathbb{1}^\top - (xx^\top - yy^\top)\right), \dot{E}\right\rangle.$$

□

The negative gradient $-G = -G_\varepsilon(E(t))$ introduced in Lemma 5.3.1 gives the steepest descent direction for minimizing the objective functional, without considering the constraint on the norm of E. In this setting, it is possible to show that, when $\lambda \neq \mu$, the gradient $G_\varepsilon(E(t))$ never vanishes, as predicted in Chapter 2.

Lemma 5.3.2. *With the same notations introduced in Lemma 5.3.1, assume that $E \in \mathcal{D}_1$ and that the Laplacian $L(W + \varepsilon E)$ has two distinct eigenvalues $\lambda \neq \mu$. Then $L^*(xx^\top - yy^\top) \neq 0$, which means that $G_\varepsilon(E(t)) \neq 0$ for all t.*

Proof. Assume by contradiction that $L^*(xx^\top - yy^\top) = 0$. Then, by means of (5.4), for all $(i, j) \in \mathscr{E}$ (i.e. the entries (i, j) belonging to the pattern $\mathcal{E}$) it holds that

$$\frac{x_i^2 + x_j^2}{2} - x_i x_j = \frac{y_i^2 + y_j^2}{2} - y_i y_j,$$

which is equivalent to

$$\frac{1}{2}(x_i - x_j)^2 = \frac{1}{2}(y_i - y_j)^2.$$

Multiplying by the weights $\hat{w}_{i,j} = w_{i,j} + \varepsilon e_{i,j}$ and summing over all $(i,j) \in \mathscr{E}$ yields

$$\sum_{(i,j)\in\mathscr{E}} \frac{1}{2}\hat{w}_{i,j}(x_i - x_j)^2 = x^\top L(W + \varepsilon E)x = \lambda$$

where we have followed the same steps as in (5.1). Similarly

$$\sum_{(i,j)\in\mathscr{E}} \frac{1}{2}\hat{w}_{i,j}(y_i - y_j)^2 = y^\top L(W + \varepsilon E)y = \mu,$$

which yields $\lambda = \mu$, in contradiction with the assumption. □

The following result, see e.g. [3, 32] and [34], shows the optimal direction to take in order to fulfil the unit norm condition, which can be rewritten as $\langle E, \dot{E}\rangle = 0$.

Lemma 5.3.3. *Given $E \in \mathcal{E}_1$ and $G \in \mathcal{E}$, the solution of the optimization problem*

$$\underset{Z\in\mathcal{E}_1,\ \langle Z,E\rangle=0}{\arg\min}\quad \langle G, Z\rangle \tag{5.5}$$

is

$$\alpha Z_\star = -G + \langle G, E\rangle E,$$

where α is a normalization parameter assuring that $\|Z_\star\|_F = 1$.

Proof. By considering the real setting, the proof is identical to that of Lemma 3.1.2. □

Lemmas 5.3.1 and 5.3.3 suggest to consider the matrix ordinary differential equation

$$\dot{E}(t) = -G_\varepsilon(E(t)) + \langle G_\varepsilon(E(t)), E(t)\rangle E(t), \tag{5.6}$$

whose stationary points are zeros of the derivative of the objective functional $F_\varepsilon(E(t))$. Equation (5.6) is a gradient system for $F_\varepsilon(E(t))$, since along its trajectories

$$\frac{\mathrm{d}}{\mathrm{d}t}F_\varepsilon(E(t)) = \varepsilon\left(-\|G_\varepsilon(E(t))\|_F^2 + (\langle G_\varepsilon(E(t)), E(t)\rangle)^2\right) \le 0$$

by means of the Cauchy-Schwarz inequality, which also implies, similarly to Theorem 3.1.4, that the derivative vanishes in $E_\star$ if and only if $E_\star$ is a stationary point of (5.6). Thanks to the monotonicity property along the trajectories, an integration of this gradient system leads necessarily to a stationary point $E_\star$ that belongs, by construction, to $\mathcal{E}_1$.

However, in the formulation of (5.6), it is not guaranteed that the found stationary point $E_\star$ is an admissible perturbation of W, because $W + \varepsilon E_\star$ may have some negative entries and hence it would not provide a solution of the optimization problem (5.3). In our experience this usually does not occur, but it is a possibility we wish to avoid for safety. In order to ensure the admissibility of $E_\star$, we need to take into account the non-negative constraint $W + \varepsilon E_\star \ge 0$ componentwise, i.e. that $E_\star \in \mathcal{D}_1$.

5.3.1 Penalized gradient system

A possible way to impose that the path $E(t)$ is contained in $\mathcal{D}_1$ is by introducing the penalization term proposed in [3, 37]

$$Q_\varepsilon(E) = \frac{1}{2} \sum_{(i,j)\in\mathscr{E}} \left((w_{ij} + \varepsilon e_{ij})_-\right)^2 ,$$

where $(a)_- = \min(a, 0)$ denotes the negative part of a. The new objective functional becomes

$$F_{\varepsilon,c}(E) = F_\varepsilon(E) + cQ_\varepsilon(E),$$

where $c > 0$ is the penalization size and the new optimization problem for the *structured inner iteration* (5.3) turns into the *constrained structured inner iteration*

$$\arg\min_{E\in\mathcal{E}_1} F_{\varepsilon,c}(E). \tag{5.7}$$

In this way solutions of (5.7) are forced to stay close to the set $\mathcal{D}$ if c is big enough, in order to fulfil the non-negativity constraint of the weight matrix. As shown in [3, 37], the results for $F_\varepsilon(E)$ extend to this new functional.

Lemma 5.3.4. *With the same hypothesis of Lemma 5.3.1 we have*

$$\frac{1}{\varepsilon}\frac{\mathrm{d}}{\mathrm{d}t} F_{\varepsilon,c}(E(t)) = \langle G_{\varepsilon,c}(E(t)), \dot{E}(t)\rangle,$$

where

$$G_{\varepsilon,c}(E) = G_\varepsilon(E) + c(W + \varepsilon E)_-$$

is the penalized gradient.

Proof. Since $E, \dot{E} \in \mathcal{E}$ are symmetric, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} Q_\varepsilon(E(t)) = \sum_{(i,j)\in\mathscr{E}} \varepsilon \dot{e}_{ij}(t)(w_{ij} + \varepsilon e_{ij}(t))_- = \varepsilon\langle \dot{E}(t), (W + \varepsilon E(t))_-\rangle,$$

where $\dot{E}(t) = (\dot{e}_{ij}(t))$. The same steps of the proof of Lemma 5.3.1 lead to the claim. □

By replacing the gradient with the penalized gradient $G_{\varepsilon,c}(E)$, we obtain, as we did for equation (5.6), the ODE

$$\dot{E} = -G_{\varepsilon,c}(E) + \langle G_{\varepsilon,c}(E), E\rangle E. \tag{5.8}$$

In analogy to the previous section, we can show that equation (5.8) is a gradient system whose stationary points are the only zeros of the derivative of $F_{\varepsilon,c}(E)$. Thus the trajectory $E(t)$ of equation (5.8) is forced to stay close to $\mathcal{D}_1$, when c is big enough, and hence the reached stationary points are admissible solutions of problem (5.2) up to an error that is small if c is large. We give further details in the numerical examples in Section 5.6.

5.4 Rank-4 symmetric projection of the gradient system

In this section we will consider a modified version of (5.2), which does not take into account the non-negativity constraint of the set $\mathcal{D}$:

$$\widetilde{W}_\star^{(k)} = \arg\min_{\Delta\in\mathcal{E}} \{\|\Delta\|_F : \ \lambda_k(L(W+\Delta)) = \lambda_{k+1}(L(W+\Delta))\}\,. \tag{5.9}$$

The introduction of this new problem is motivated by the observation that in our experiments the violation of this constraint seems to be uncommon and hence generally $\widetilde{W}_\star^{(k)}$ and the solution of (5.2) coincide. In the case when the solutions of (5.2) and (5.9) are the same, we propose a new matrix ODE whose aim is to exploit the underlying low-rank property of the problem and it allows to solve more efficiently the *structured inner iteration* (5.3). In this way it is possible to generalize also in this framework the ideas discussed in Section 3.1.2.

5.4.1 Formulation of the low-rank symmetric ODE

We introduce two low-rank matrices N and R, depending on the matrix E, on the perturbation size ε and on the fixed positive integer k (which will be omitted for brevity):

$$N = N_\varepsilon(E) = z\mathbb{1}^\top - xx^\top + yy^\top, \quad R = R_\varepsilon(E) = \frac{N+N^\top}{2} = \frac{z\mathbb{1}^\top + \mathbb{1}z^\top}{2} - xx^\top + yy^\top,$$

where

$$z = x \bullet x - y \bullet y \tag{5.10}$$

is the vector of entries $z_i = x_i^2 - y_i^2$ and $\bullet$ denotes the componentwise product. We observe that the gradient introduced in Lemma 5.3.1 can be rewritten as

$$G = G_\varepsilon(E) = \Pi_{\mathcal{E}}(N_\varepsilon(E)) = \Pi_{\mathcal{S}}(R_\varepsilon(E)),$$

where $\Pi_{\mathcal{S}}$ denotes the (orthogonal) projection onto the pattern $\mathcal{S}$ associated with the low-rank symmetric matrix R. Note that

$$\Pi_{\mathcal{E}}(M) = \Pi_{\mathcal{S}}(\mathrm{sym}(M)), \qquad \forall M \in \mathbb{R}^{n\times n}.$$

Remark 5.4.1. *Since x and y have unit 2-norm, we observe that*

$$z^\top\mathbb{1} = \sum_{i=1}^n (x_i^2 - y_i^2) = 1 - 1 = 0,$$

which means that $\mathbb{1}$ and z are orthogonal. The vectors x, y and z are generally linearly independent, but it may happen that they are not; however this seems to be a non-generic case, up to some very specific counterexamples (see e.g. Example C.0.2). In the following we assume that x, y and z are linearly independent, which implies, that the matrix N has rank 3 *and hence R has rank* 4.

As done for equation (3.4), we observe that solutions of (5.6) can be rewritten as $E = \Pi_{\mathcal{S}}Z$, where Z solves the ODE

$$\dot{Z} = -R_\varepsilon(E) + \langle R_\varepsilon(E), E\rangle Z, \tag{5.11}$$

and we recall $G_\varepsilon(E) = \Pi_{\mathcal{S}} R_\varepsilon(E)$. We take inspiration from equation (5.11) and consider the ODE in $\mathcal{M}_4 = \{A \in \mathbb{R}^{n\times n} : \mathrm{rank}(A) = 4\}$

$$\dot{Y} = -P_Y R_\varepsilon(E) + \langle P_Y R_\varepsilon(E), E\rangle Y, \qquad E = \Pi_{\mathcal{S}} Y, \tag{5.12}$$

where P_Y is the orthogonal projection, with respect to the Frobenius inner product, onto the tangent space $\mathcal{T}_Y\mathcal{M}_4$ at Y (see Proposition B.0.3 or [51]). Since in this case Y is symmetric of rank 4, it can be written as $Y = USU^\top$ where $U \in \mathbb{R}^{n\times 4}$ has full-rank and orthonormal columns and $S \in \mathrm{sym}(\mathbb{R}^{4\times 4})$ is invertible and hence the expression for the projection P_Y takes the simpler form

$$P_Y A = A - (I - UU^\top)A(I - UU^\top) = UU^\top A + AUU^\top - UU^\top AUU^\top, \tag{5.13}$$

where I denotes the $n \times n$ identity matrix.

The following result states two important properties of the solution $Y(t)$ of (5.12).

Lemma 5.4.2. *Let $Y(t)$ be a solution of equation* (5.12) *for $t \in [0, +\infty)$ with starting value $Y(0) = Y_0 \in \mathrm{sym}(\mathbb{R}^{n\times n}) \cap \mathcal{M}_4$. Then $Y(t) \in \mathrm{sym}(\mathbb{R}^{n\times n}) \cap \mathcal{M}_4$ for all t. Moreover, if $\|\Pi_{\mathcal{S}} Y_0\|_F = 1$, then $\|\Pi_{\mathcal{S}} Y(t)\|_F = 1$ for all t.*

Proof. Since $P_Y(Y) = Y$, the right hand side of (5.12) is $P_Y(-R_\varepsilon(E) + Y) \in \mathcal{T}_Y\mathcal{M}_4$, which means that the whole trajectory $Y(t)$ belongs to the rank-4 manifold. Similarly, $\dot{Y}(t) \in \mathrm{sym}(\mathbb{R}^{n\times n})$ and hence $Y(t)$ is symmetric for all t. Finally the unit norm of $E = \Pi_{\mathcal{S}} Y$ is conserved by the ODE, since $\langle E, Y\rangle = \|E\|_F^2 = 1$ and

$$\langle \Pi_{\mathcal{S}} Y, \Pi_{\mathcal{S}} \dot{Y}\rangle = \langle \Pi_{\mathcal{S}} Y, \dot{Y}\rangle = -\langle E, P_Y(R_\varepsilon(E))\rangle + \langle P_Y R_\varepsilon(E), E\rangle\langle E, Y\rangle = 0.$$

□

In the next sections we investigate how the solution of (5.12) is related to that of (5.6). More precisely we are interested in their stationary points and in the monotonicity property of the low-rank system, which are crucial for the implementation of the *structured inner iteration* (5.3) by means of the low-rank ODE. If these properties are shared between the equations, then it would be possible to integrate the low-rank ODE instead of the original ODE.

5.4.2 Relationship between stationary points

As shown in Theorem 3.1.4 and Theorem 3.1.6, it is possible to characterize the stationary points of the ODEs (5.6) and (5.12) as real multiples of the associated gradient, where here the structured gradient writes as $G_\varepsilon(E) = \Pi_{\mathcal{S}} R_\varepsilon(E)$. Also Theorem 3.1.7 can be adapted to this setting and it turns out that equations (5.6) and (5.12), under non-degeneracy conditions, share the same stationary points, as stated by the following result.

Theorem 5.4.3. *Consider the two matrix ordinary differential equations*

$$\dot{E} = -G_\varepsilon(E) + \langle G_\varepsilon(E), E\rangle E, \tag{5.14}$$

$$\dot{Y} = -P_Y R_\varepsilon(E) + \langle P_Y R_\varepsilon(E), E\rangle Y, \tag{5.15}$$

1. *Let $E_\star \in \mathcal{E}_1$ of unit Frobenius norm be a stationary point of* (5.14). *Then $E_\star = \Pi_{\mathcal{S}} Y_\star$ for a certain symmetric matrix $Y_\star \in \mathcal{M}_4$ that is a stationary point of* (5.15).

2. *Conversely, let $Y_\star \in \mathcal{M}_4$ be a symmetric stationary point of* (5.15) *such that $E_\star = \Pi_{\mathcal{S}} Y_\star$ has unit Frobenius norm and $P_{Y_\star} R_\star \neq 0$, where $R_\star = R_\varepsilon(E_\star)$. Then $P_{Y_\star} R_\star = R_\star$, $Y_\star$ is a non-zero real multiple of $R_\star$ and $E_\star$ is a stationary point of* (5.14).

Proof. It is analogous to that of Theorem 3.1.7. □

5.4.3 Local convergence to the stationary points of the rank-4 ODE

Theorem 5.4.3 ensures that the original and the low-rank ODEs share the same stationary points. Now we are interested in understanding whether the integration of (5.12) leads to at least one of the local minima (i.e. the stationary points of the low-rank ODE) or not. This convergence is always guaranteed for equation (5.6), since it is a gradient system, but unfortunately equation (5.12) is not a gradient system and the monotonicity property of the functional may not hold.

However, provided a suitable starting value for the integration of the ODE sufficiently close to a local minimum, the low-rank ODE turns out to be close to a gradient system. The following key lemma adapts the result of Lemma 3.1.8 and it shows the main reason behind this fact.

Lemma 5.4.4. *Let $Y_\star \in \mathcal{M}_4 \cap \mathrm{sym}(\mathbb{R}^{n\times n})$ be a stationary point of the rank-4 ODE* (5.12) *such that $E_\star = \Pi_{\mathcal{S}} Y_\star \in \mathcal{E}_1$ and $P_{Y_\star} R(E_\star) \neq 0$. Then there exists $\delta_\star > 0$ such that for all $\hat{Y} \in \mathcal{M}_4$ that satisfy*

$$\|\hat{Y} - Y_\star\|_F = \delta < \delta_\star, \qquad \hat{E} = \Pi_{\mathcal{S}} \hat{Y} \in \mathcal{E}_1,$$

we have

$$\|P_{\hat{Y}} R_\varepsilon(\hat{E}) - R_\varepsilon(\hat{E})\|_F \leq C\delta^2,$$

where C is a positive constant independent of δ.

Proof. The proof is similar to that of Lemma 3.1.8. We just show that in this framework it is possible to bound the derivatives of the eigenvectors x and y through suitable formulas as in [58] that involve the group inverse (see [11]), that here coincides with the more familiar Moore-Penrose pseudo-inverse and then apply the same ideas of [34, Lemma 4.5]. The matrix $R = R_\varepsilon(E(t))$ can be rewritten in the form

$$R = \left(\frac{z+\mathbb{1}}{2}\right)\left(\frac{z+\mathbb{1}}{2}\right)^\top - \left(\frac{z-\mathbb{1}}{2}\right)\left(\frac{z-\mathbb{1}}{2}\right)^\top - xx^\top + yy^\top = R_U R_S R_U^\top,$$

where

$$R_U = \begin{pmatrix} z+\mathbb{1} & z-\mathbb{1} & x & y \end{pmatrix}, \qquad R_S = \begin{pmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & -\frac{1}{4} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Thus, by means of a QR decomposition $R_U = Q\Lambda$ where $Q \in \mathbb{R}^{n\times 4}$ has orthonormal columns and Λ is a 4-by-4 invertible symmetric matrix, it follows that Q and Λ depend smoothly on $\mathbb{1}, x, y$ and z, where we assume that the order and sign of the columns of Q do not change. Hence R depends smoothly on the eigenvectors of the Laplacian of the perturbed weight matrix and x and y have bounded derivatives. □

Now we are ready to adapt the local convergence result of Theorem 3.1.10 to the framework of this chapter.

Theorem 5.4.5. *Let $Y_\star \in \mathcal{M}_4 \cap \operatorname{sym}(\mathbb{R}^{n\times n})$ be a stationary point of the projected differential equation* (5.12) *such that $E_\star = \Pi_{\mathcal{S}} Y_\star \in \mathcal{S}_1$ and $P_{Y_\star} R_\varepsilon(E_\star) \neq 0$. Suppose that $E_\star$ is a strict local minimum of the functional F_ε on $\mathcal{S}_1$ and assume that*

$$\Pi_{\mathcal{S}}\Big|_{\mathcal{M}_4} : \mathcal{M}_4 \to \Pi_{\mathcal{S}}(\mathcal{M}_4) \subseteq \mathcal{S}$$

is a diffeomorphism. Then, for an initial datum $Y(0)$ sufficiently close to $Y_\star$, the solution $Y(t)$ of (5.12) *converges to $Y_\star$ exponentially as $t \to +\infty$. Moreover $F_\varepsilon(\Pi_{\mathcal{S}} Y(t))$ decreases monotonically with t and converges exponentially to the local minimum value $F(E_\star)$ as $t \to +\infty$.*

Proof. It follows the same idea of the proof of Theorem 3.1.10. □

The assumption that $\Pi_{\mathcal{S}}\big|_{\mathcal{M}_4}$ is a diffeomorphism is sufficient in Theorem 5.4.5, but it seems that it is not necessary. In fact in our experience we observed that the integration of (5.12) converges even in many examples where the dimension of $\mathcal{S}$ is small. Thus we believe that this assumption is not restrictive (see Section 4.4 for more details).

Theorem 5.4.5 proves that, at least locally, the integration of the low-rank ODE approaches a stationary point, as it happens for the gradient system (5.6). Hence equation (5.12) can replace the original ODE. This fact can lead to computational benefit from the low-rank underlying structure of the problem.

5.4.4 Implementation of the low-rank inner iteration

In this section we illustrate some details of the implementation of the *structured inner iteration* for solving equation (5.9), through a numerical integration of a system of ODEs. We show how to highlight the low-rank properties of equation (5.12) by means of an equivalent system of ODEs, and we discuss the numerical integration of the system.

Given a decomposition $Y = USU^\top$, where $U \in \mathbb{R}^{n\times 4}$ has orthonormal columns and $S \in \mathbb{R}^{4\times 4}$ is invertible, we can rewrite equation (5.12) as

$$\dot{U}SU^\top + U\dot{S}U^\top + US\dot{U}^\top = -R + (I - UU^\top)R(I - UU^\top) + \eta USU^\top, \tag{5.16}$$

where $\eta = \langle P_Y(R_\varepsilon(E)), E\rangle$. In the real setting, the property $U^\top U = I$ implies $U^\top \dot{U} + \dot{U}^\top U = 0$. Since the decomposition of Y is not unique, because for any orthogonal square matrix $Q \in \mathbb{R}^{4\times 4}$ we have another equivalent decomposition $Y = UQ(Q^\top SQ)Q^\top U^\top$, we uniquely select the following one

$$\begin{cases} \dot{U} = -(I - UU^\top)RUS^{-1} \\ \dot{S} = -U^\top RU + \eta S \end{cases}, \tag{5.17}$$

which satisfies the gauge condition $U^\top \dot{U} = 0$ and satisfies (5.16) (equivalently (5.12)). The matrix S may lose the diagonal structure along the trajectory, but the decomposition $Y = USU^\top$ still holds, where U has orthonormal columns and S is symmetric. System (5.17) consists of two matrix ODEs of dimension n-by-4 and 4-by-4 respectively.

Integration of system (5.17) can be done in many ways. The simplest choice is the normalized explicit Euler's method, which generally performs well. However in some cases the matrix S may be close to singularity and Euler's method may suffer the presence of the inverse of S in its formulation. This problem can be overcome by means of a different integrator. Since we are not interested in the whole trajectory

$Y(t)$, but only in the approximation of its stationary points, we can use a splitting method similar to that proposed in [14]. Algorithm 4 shows the outline of a single step integration of this approach. The right-hand side of (5.12)

$$-P_Y(R_\varepsilon(E)) + \eta Y = P_Y(-R_\varepsilon + \eta Y), \qquad \eta := \langle P_Y(R_\varepsilon(E)), E\rangle,$$

can be rewritten explicitly, according to equation (5.13), as the sum of three alternating projections that are integrated consecutively by the splitting integrator. The first two steps of Algorithm 4 correspond to the "K-step" and to the "L-step" of the algorithm proposed in [14], which in this case coincide since the ODE is symmetric, while the remaining steps perform the "S-step". Then the *structured inner iteration* (5.3) is given by a combinations of the single integration steps of Algorithm 4, as shown in Algorithm 5.

The choice of the stepsize is performed by means of an Armijo-type line search strategy as in [34], since the time derivative of the objective function is available. Provided a starting point sufficiently close to a stationary point, Theorem 5.4.5 ensures the convergence towards that stationary point. A possible choice for U_0 and S_0 comes from a decomposition of the gradient as provided in the proof of Lemma 5.4.4: this choice generally leads to a suitable approximation of a minimizer. We compute $x_0 = x(L(W))$, $y_0 = y(L(W))$ and $z_0 = x_0 \bullet x_0 - y_0 \bullet y_0$ and we choose $Y_0 := U_0 S_0 U_0^\top$, where

$$[U_0, D_0] = \mathtt{qr}\left(\begin{pmatrix} z_0 + \mathbb{1} & z_0 - \mathbb{1} & x_0 & y_0 \end{pmatrix}\right), \qquad S_0 = -D_0 \begin{pmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & -\frac{1}{4} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} D_0^\top, \tag{5.18}$$

and $[Q, R] = \mathtt{qr}(A)$ denotes the thin QR factorization of $A = QR$. However, during the *structured outer iteration* (5.9), it could be more convenient to choose as starting value for the iteration of $\varepsilon_{\ell+1}$ the stationary points found in the ℓ-th outer iteration, as shown in Algorithm 6.

Algorithm 4 Splitting method for the numerical integration step from t_0 to $t_1 = t_0 + h$

```
Input: $U_0 \in \mathbb{R}^{n\times 4}$ orthogonal and $S_0 \in \mathrm{sym}(\mathbb{R}^{4\times 4})$ invertible such that $Y_0 = U_0 S_0 U_0^\top = Y(t_0)$

Output: $U_1 \in \mathbb{R}^{n\times 4}$ orthogonal and $S_1 \in \mathrm{sym}(\mathbb{R}^{4\times 4})$ invertible such that $Y_1 = U_1 S_1 U_1^\top \approx Y(t_1)$

    $K$-step/$L$-step
    Compute $K_1 = U_0 S_0 + h\left(-R(Y_0)U_0 + \langle P_{Y_0} R(Y_0), \Pi_{\mathcal{S}} Y_0\rangle U_0 S_0\right)$.
    Perform a thin QR factorization $K_1 = U_1 T_1$ and compute $M_1 = U_1^\top U_0$.
    $S$-step
    Define $\hat{S}_0 = M S_0 M^\top$ and $\hat{Y}_0 = U_1 \hat{S}_0 U_1^\top$.
    Normalize $\hat{Y}_0$ and get $\tilde{Y}_0 = U_1 \tilde{S}_0 U_1$ such that $\|\Pi_{\mathcal{S}} \tilde{Y}_0\|_F = 1$.
    Compute $\tilde{R}_0 = R(\tilde{Y}_0)$ and $\eta = \langle P_{\tilde{Y}_0} \tilde{R}_0, \Pi_{\mathcal{S}} \tilde{Y}_0\rangle$.
    Compute $\tilde{S}_1 = \tilde{S}_0 + h U_1^\top \left(-\tilde{R}_0 + \eta \tilde{Y}_0\right) U_1$.
    Normalize $\tilde{S}_1$ and get $S_1$ such that $\|\Pi_{\mathcal{S}}(U_1 S_1 U_1^\top)\|_F = 1$.
    Return $U_1$ and $S_1$.
```

Algorithm 5 Inner iteration

Input: A weight matrix W, a perturbation size $\varepsilon > 0$, the starting values U_0 and S_0, an initial stepsize h_0, a tolerance τ_{inn} and a maximum number of iterations maxit

Output: The matrices $U_\star(\varepsilon)$ and $S_\star(\varepsilon)$ that form the solution of the optimization problem (5.3) $E_\star(\varepsilon) = \Pi_{\mathcal{S}}(U_\star(\varepsilon)S_\star(\varepsilon)U_\star(\varepsilon)^\top)$

```
Initialize $U_0$ and $S_0$ (e.g. by means of (5.18)).
Compute $f_0 = F_\varepsilon(\Pi_{\mathcal{S}}(U_0 S_0 U_0^\top))$ and set $f_1 = +\infty$
Set $j = 0$.
while $|f_1 - f_0| > \tau_{\text{inn}}$ and $j <$ maxit do
    With an Armijo stepsize choice, perform Algorithm 4 and compute $U_1$ and $S_1$.
    Update $f_0 = f_1$, $f_1 := F_\varepsilon(\Pi_{\mathcal{S}}(U_1 S_1 U_1^\top))$ and set $j := j + 1$.
end while
```

5.5 The structured outer iteration

Once that a computation of the optimizers is available for a given $\varepsilon > 0$ and a fixed k, we need to determine an optimal value for the perturbation size. Let $E_\star(\varepsilon)$ be a solution of the optimization problem (5.3) and consider the function

$$\varphi(\varepsilon) := F_\varepsilon(E_\star(\varepsilon)).$$

This function is non-negative and we define $\varepsilon_\star$ as the smallest zero of φ. By assuming that the k-th and $(k+1)$-st eigenvalues of $L(W + \varepsilon E_\star(\varepsilon))$ are simple for $0 \le \varepsilon < \varepsilon_\star$, we conclude that φ is a smooth function in the interval $[0, \varepsilon_\star)$. The aim of the *structured outer iteration* is to approximate $\varepsilon_\star$, which is the solution of the optimization problem (5.9).

In order to find $\varepsilon_\star$ we use a combination of the well-known Newton and bisection methods, which provides an approach similar to [24, 28] or [34]. If the current approximation ε lies in $(0, \varepsilon_\star)$, it is possible to exploit Newton's method, since φ is differentiable there (see Lemma 5.5.1); otherwise, if $\varepsilon > \varepsilon_\star$, we use the bisection method (see Algorithm 6). The following result provides a simple formula for the first derivative of φ required by Newton's method.

Lemma 5.5.1. *It holds that*

$$\varphi'(\varepsilon) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} F_\varepsilon(E_\star(\varepsilon)) = \langle G_\varepsilon(E_\star(\varepsilon)), E_\star(\varepsilon)\rangle = -\|G_\varepsilon(E_\star)\|_F.$$

Proof. It follows the same pattern of Lemma 2.4.1. □

The tolerance τ_{inn} and τ_{out} may coincide, but sometimes it could be useful to choose slightly different values to improve the accuracy of the result. Finally we perform Algorithm 6 for some values of $k \in \{k_{\min}, \dots, k_{\max}\}$ and we select the index of the largest structured distance computed.

5.5.1 The penalized version

A similar approach can be followed for the penalized problem (5.7). For $\varepsilon, c > 0$, let $E_\star(\varepsilon, c)$ be a solution of the penalized *structured inner iteration* (5.7) and consider the

Algorithm 6 Outer iteration

Input: A weight matrix W, an interval and an initial guess $\varepsilon_0 \in [\varepsilon_{\rm lb}, \varepsilon_{\rm ub}]$ for $\varepsilon_\star$, a tolerance $\tau_{\rm out}$ and a maximum number of iterations niter

Output: The value $\varepsilon_\star$ and the associated minimizer $E_\star(\varepsilon_\star)$

```
Perform Algorithm 5 and compute $E_\star(\varepsilon_0)$.
Set $\ell = 0$.
while $\ell <$ niter and $\varepsilon_{\rm ub} - \varepsilon_{\rm lb} > \tau_{\rm out}$ do
    if $\varphi(\varepsilon_\ell) <$ toler then
        Set $\varepsilon_{\rm ub} := \min(\varepsilon_{\rm ub}, \varepsilon_\ell)$.
        Set $\varepsilon_{\ell+1} := \frac{\varepsilon_{\rm lb}+\varepsilon_{\rm ub}}{2}$ (bisection step).
    else
        Set $\varepsilon_{\rm lb} := \max(\varepsilon_{\rm lb}, \varepsilon_\ell)$.
        Compute $\varphi(\varepsilon_\ell)$ and $\varphi'(\varepsilon_\ell)$.
        Update $\varepsilon_{\ell+1} := \varepsilon_\ell - \frac{\varphi(\varepsilon_\ell)}{\varphi'(\varepsilon_\ell)}$ (Newton step).
    end if
    if $\varepsilon_{\ell+1} \notin [\varepsilon_{\rm lb}, \varepsilon_{\rm ub}]$ then
        Set $\varepsilon_{\ell+1} := \frac{\varepsilon_{\rm lb}+\varepsilon_{\rm ub}}{2}$.
    end if
    Set $\ell := \ell + 1$.
    Compute $E_\star(\varepsilon_\ell)$ by applying Algorithm 5 with starting value $E_\star(\varepsilon_{\ell-1})$.
end while
Return $\varepsilon_\star := \varepsilon_\ell$ and $E_\star(\varepsilon_\star)$.
```

function

$$\varphi_c(\varepsilon) = F_{\varepsilon,c}(E_\star(\varepsilon.c))$$

Let $\varepsilon_\star$ be the smallest zero of φ_c. Again assuming that the k-th and $(k+1)$-st eigenvalues of $L(W + \varepsilon E(\varepsilon, c))$ are simple, for $0 \le \varepsilon < \varepsilon_\star$, yields that φ_c is a smooth function in the interval $[0, \varepsilon_\star)$.

Lemma 5.5.2. *It holds that*

$$\varphi_c'(\varepsilon) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} F_{\varepsilon,c}(E_\star(\varepsilon)) = \langle G_{\varepsilon,c}(E_\star(\varepsilon,c)), E_\star(\varepsilon,c)\rangle = -\|G_{\varepsilon,c}(E_\star(\varepsilon,c))\|_F.$$

Proof. It is a direct consequence of Lemma 5.5.1. □

As done for the low-rank method, it is possible to implement an algorithm to solve problem (5.2) by introducing the penalization term. The only difference is in the *structured inner iteration* where we integrate equation (5.8), for instance with the normalized Euler's method.

5.6 Numerical experiments

In this section we compare the behaviour of the spectral gaps

$$g_k(W) = \lambda_{k+1}(W) - \lambda_k(W)$$

and the structured distance to ambiguity as stability indicators. For the computation of $d_k(W)$ we use both Algorithm 6 and the integration of the full-rank system. To distinguish between these two results, we will denote, respectively, the unstructured distances as

$$d_k^{\mathrm{low}}(W), \qquad d_k^{\mathrm{full}}(W).$$

If the optimizer found is not admissible, we integrate only the penalized equation (5.8), since the low-rank system is not suitable. However, if the solution (ε, E) found is such that the norm of $\min(W + \varepsilon E, 0)$ is not large, say less than 0.05, we make it admissible by replacing it with the admissible solution $(\tilde{\varepsilon}, \widetilde{E})$ defined as

$$W + \varepsilon E - \min(W + \varepsilon E, 0) = W + \tilde{\varepsilon}\widetilde{E}, \tag{5.19}$$

where the minimum between two matrices is meant entry-wise, and hence we consider also the low-rank solution in these cases. In the experiments we also report the absolute value of the difference between the two distances

$$e_k(W) = |d_k^{\mathrm{low}}(W) - d_k^{\mathrm{full}}(W)|,$$

the number of outer iterations, denoted by M_k^{low} and M_k^{full} and the number of calls to the MATLAB's `eigs` function, denoted by N_k^{low} and N_k^{full}. In order to have a more detailed analysis, for the first example we compare our results with those presented in [3], even though the latter method uses a different metric with respect to the one we use here for the computation of the distance to ambiguity (which will be denoted by $\delta_k(W)$).

We present four different examples with different features: in the first three the penalization term is not required and hence we can use Algorithm 6, while the last one shows a non-common case where the non-negativity constraint must be taken into account. Finally we compare the algorithm with a graph partitioning method proposed in [43]. In all experiments we set the tolerance of $\tau_{\mathrm{out}} = 10^{-2}$ for the *structured outer iteration.*

5.6.1 A Machine Learning example: the ECOLI matrix

The ECOLI matrix is a Machine Learning dataset from the SuiteSparse Matrix Collection and the UCI Machine Learning Repository (see [61]) that describes the protein localization sites of the bacteria *E. coli.* For n data points, the connectivity matrix $C \in \mathbb{R}^{n\times n}$ is created from a k-nearest neighbours routine, with k set such that the resulting graph is connected. The similarity matrix $S \in \mathbb{R}^{n\times n} = (s_{ij})$ between the data points is defined as

$$s_{ij} = \max\{s_i(j), s_j(i)\} \;\text{ with } s_i(j) = \exp\left(-4\frac{\|x_i - x_j\|^2}{\sigma_i^2}\right)$$

with σ_i standing for the Euclidean distance between the i-th data point and its closest k-nearest neighbour, namely the nearest connected vertex in the graph. The adjacency matrix W is then created as $W = C \bullet S$ (here $\bullet$ denotes the componentwise product).

This matrix has $n = 336$ vertices and $m = 4560 \approx 13.6n$ edges and its pattern is shown in Figure 5.1. In this case the structure of W contains three possible clusters and the second and third appear to be split in two sub-communities. This facts suggest that a suitable number of clusters should be $k \in \{3, 4, 5\}$. We test the capability of our methods to identify this feature in Table 5.1, which shows the performances of the algorithms with $\tau_{\mathrm{inn}} = 10^{-9}$, where for $k = 3, 4$ the solutions have been made admissible by means of (5.19). It is evident that for all the methods the best choices are $k = 3, 4$ or 8, but the criteria disagree on the optimal choice. More precisely the

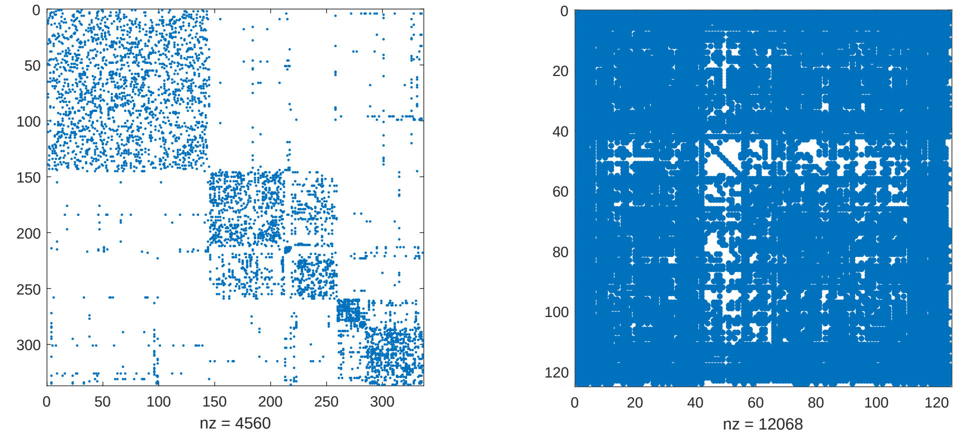


FIGURE 5.1: Patterns of the ECOLI (left) and JOURNALS (right) matrices.

k	$g_k(W)$	$\delta_k(W)$	$d_k^{\text{low}}(W)$	$d_k^{\text{full}}(W)$	$e_k(W)$	M_k^{low}	M_k^{full}	N_k^{low}	N_k^{full}
2	0.0079	0.0056	$< 10^{-7}$	$< 10^{-7}$	$< 10^{-7}$	2	2	36	138
3	**0.0328**	**0.2739**	0.1096	0.1139	0.0043	8	8	840	1522
4	0.0223	0.1164	**0.1544**	**0.1535**	0.0019	9	9	122	1206
5	0.0048	0.0034	$< 10^{-7}$	$< 10^{-7}$	$< 10^{-7}$	2	2	12	158
6	0.0093	0.0066	$< 10^{-7}$	$< 10^{-7}$	$< 10^{-7}$	2	2	18	127
7	0.0074	0.0053	$< 10^{-7}$	$< 10^{-7}$	$< 10^{-7}$	2	2	22	158
8	0.0316	0.0888	0.0731	0.0731	$< 10^{-7}$	7	7	462	635
9	0.0133	0.0174	0.0110	0.0110	$< 10^{-7}$	6	6	36	545
10	0.0113	0.0109	0.0035	0.0035	$< 10^{-7}$	6	6	60	700

TABLE 5.1: Comparison between the distances for ECOLI matrix. The marked bold results indicate the best value for k according to each method. According to all the criteria, $k = 3, 4$ are the best choices.

largest spectral gap is $g_3(W) = 0.328$ slightly greater than $g_8(W) = 0.316$, the largest structured distance for the method of [3] is $\delta_3 = 0.2739$, while for the structured distances d_k^{low} and d_k^{full} the best choice is $k = 4$, that is preferable than $k = 3$. In this example, the size and the pattern of the matrix implies that the low-rank ODE is more convenient than the full-rank gradient system. In particular the gain in memory requirement is given by the ratio between m (for E of ODE (5.6)) and $4n + 16$ (for U and S of system (5.17)), that is

$$\frac{m}{4n+16} = \frac{4560}{4 \cdot 336 + 16} \approx 3.35,$$

while the CPU times are 7 seconds for the low-rank, 20 seconds for the full-rank gradient system and 16 seconds for the method of [3]. Thus in this case the low-rank method is the fastest.

5.6.2 A slightly sparse example: the JOURNALS matrix

The JOURNALS matrix comes from a Pajek network converted to a sparse adjacency matrix for inclusion in the University of Florida SuiteSparse matrix collection (see [17]

k	$g_k(W)$	$d_k^{\text{low}}(W)$	$d_k^{\text{full}}(W)$	$e_k(W)$	M_k^{low}	M_k^{full}	N_k^{low}	N_k^{full}
3	8.9963	5.3133	5.3076	0.0056	9	10	2068	1954
4	**39.1923**	6.3186	6.3190	0.0004	6	6	1792	1108
5	27.9853	5.3639	5.3702	0.0063	12	12	3888	2657
6	10.4987	4.3464	4.3561	0.0097	12	13	3966	2742
7	22.0178	7.4559	7.4324	0.0235	12	12	4210	2384
8	33.9873	**8.1698**	**7.9233**	0.2464	7	12	842	2236

TABLE 5.2: Comparison between the distances for Journals matrix. The marked bold results indicate the best value for k according to each method.

for more details). It represents an undirected weighted graph with $n = 124$ vertices and $m = 12068 \approx 97n$ edges, whose structural pattern is shown in Figure 5.1.

The pattern of W does not suggest a suitable number of clusters to partition the graph. We select $k \in \{3, \dots, 8\}$ and we compare in Table 5.2 the results of the unstructured distances with the spectral gaps, where we have set the inner tolerance $\tau_{\text{inn}} = 10^{-4}$. For $k = 8$ the low-rank result has been made admissible by means of (5.19). Also in this case the two criteria for the choice of the best number of clusters disagree: while the structured distances select $k_{\text{opt}} = 8$, the unstructured distance, i.e. the largest spectral gap, prefers $k = 4$. Even in this setting there is a gain in memory saving, even larger than the previous example,

$$\frac{m}{4n+16} = \frac{12068}{4 \cdot 124 + 16} \approx 23.57,$$

which shows the ability to perform the low-rank approach for even larger matrices with moderate memory requirements. However the CPU time of the low-rank system method is 78 seconds against the 55 seconds of the full-rank system. A possible reason behind this behaviour is that the gradient system requires less effort to reach convergence and hence less eigenvalues computations, which are the most expensive procedures in all the methods. In this case the low-rank system required in total 16766 calls to `eigs`, while the full-rank system required 13081.

5.6.3 A social network community: the EGO-FACEBOOK matrix

The EGO-FACEBOOK matrix represents a dataset that consists of "circles" (or "friends lists") of the social network Facebook from the SNAP dataset (see [54] for more details). The data were collected from survey participants using this Facebook app. The whole matrix W_1 has $n_1 = 4039$ vertices with $m_1 = 176468 \approx 43.1n_1$ edges (see Figure 5.2). In order to perform further tests of the algorithms, we also consider two reduced versions, W_2 and W_3, of the whole matrix W_1.

The matrix W_2 is obtained by means of a compression that maintains the pattern and the density of the original matrix, but it halves the dimension: more precisely, for $i, j = 1, \dots, \frac{n_1-1}{2}$, we define

$$(W_2)_{i,j} = \frac{(W_1)_{2i-1,2j-1} + (W_1)_{2i-1,2j} + (W_1)_{2i,2j-1} + (W_1)_{2i,2j}}{4},$$

and then we set to zero the entries with the smallest value such that the compressed matrix has the same density of W_1 (the largest entry cancelled is 0.25). We obtain the matrix W_2 with $n_2 = 2019$ vertices and $m_2 = 43967 \approx 21.8n_2$ edges. Finally we considered the main minor W_3 of W_1 formed by the first $n_3 = 896$ vertices and with

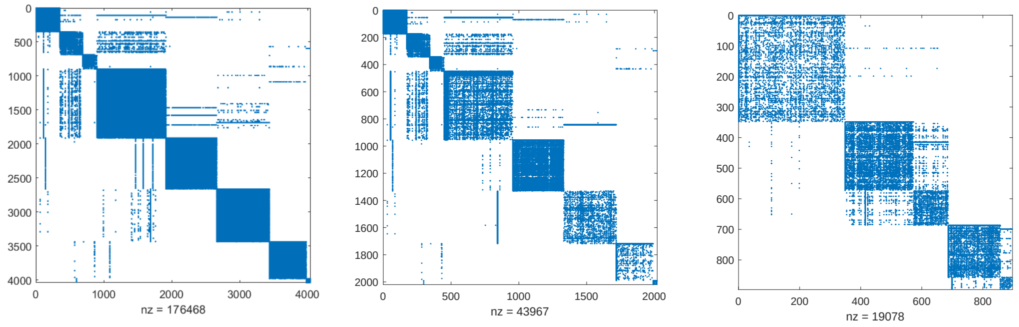


FIGURE 5.2: EGO-FACEBOOK matrices: on the left the whole structural pattern of the full matrix W_1, in the middle the compressed matrix W_2 and on the right the sub-matrix W_3.

k	$g_k(W_1)$	$d_k^{\text{low}}(W_1)$	$d_k^{\text{full}}(W_1)$	$e_k(W_1)$	M_k^{low}	M_k^{full}	N_k^{low}	N_k^{full}
3	0.0182	1.0977	1.0977	$< 10^{-7}$	10	10	436	1118
4	0.0211	3.2305	3.2305	$< 10^{-7}$	11	11	1464	1427
5	0.0423	5.7510	5.7510	$< 10^{-7}$	12	12	3086	1943
6	0.0526	**6.6364**	**6.6364**	$< 10^{-7}$	13	13	1846	1834
7	**0.5153**	1.5223	1.1744	0.3479	10	9	1704	1353
8	0.0546	0.4725	0.4725	$2 \cdot 10^{-5}$	11	11	3506	2346

TABLE 5.3: Comparison between the distances for the full EGO-FACEBOOK matrix W_1. The marked bold results indicate the best value for k according to each method.

$m_3 = 19078 \approx 21.3 n_3$ edges, which contains the first three main blocks of the whole matrix.

Whole matrix

First we analyse the whole matrix W_1. In Table 5.3 we report the results where we set the inner tolerance 10^{-9}, which will be also the tolerance of the experiments for W_2 and W_3. The criterion disagree also in this case: the unstructured distance prefers $k = 6$, while the largest spectral gap is for $k = 7$. The factor for the memory saving gain with respect to the full-rank system is

$$\frac{m_1}{4n_1 + 16} = \frac{176468}{4 \cdot 4039 + 16} \approx 10.91,$$

while the CPU time performances are 1204 seconds for the low-rank and 966 seconds for the gradient system, since also in this case the global number of calls to `eigs` of the low-rank method (12042) is larger than that of the full-rank method (10021).

Compressed matrix

In order to test the robustness of the algorithms, we compare the results between the full matrix W_1 and its compressed version W_2. Table 5.4 shows that the algorithms gives the same optimal values of the whole matrix computation, even though there are some differences in magnitude. The factor for the memory saving gain with respect to

k	$g_k(W_2)$	$d_k^{\text{low}}(W_2)$	$d_k^{\text{full}}(W_2)$	$e_k(W_2)$	M_k^{low}	M_k^{full}	N_k^{low}	N_k^{full}
3	0.0037	$< 10^{-7}$	$< 10^{-7}$	$< 10^{-7}$	2	2	98	96
4	0.0155	0.2996	0.2996	$< 10^{-7}$	10	10	1556	1146
5	0.0030	$< 10^{-7}$	$< 10^{-7}$	$< 10^{-7}$	2	2	34	103
6	0.0638	**2.0849**	**1.7597**	0.3253	13	13	1132	1870
7	**0.3865**	1.8066	1.5887	0.2179	11	10	190	1633
8	0.0036	$< 10^{-7}$	$< 10^{-7}$	$2 \cdot 10^{-5}$	2	2	4	4

Table 5.4: Comparison between the distances for the compressed EGO-FACEBOOK matrix W_2. The marked bold results indicate the best value for k according to each method.

k	$g_k(W_3)$	$d_k^{\text{low}}(W_3)$	$d_k^{\text{full}}(W_3)$	$e_k(W_3)$	M_k^{low}	M_k^{full}	N_k^{low}	N_k^{full}
3	**0.6040**	**1.5039**	**1.3450**	0.1589	10	12	1334	1025
4	0.1724	0.4883	0.4883	$< 10^{-7}$	10	10	500	922
5	0.1870	0.2650	0.2650	$< 10^{-7}$	9	9	234	706
6	$< 10^{-7}$	$< 10^{-7}$	$< 10^{-7}$	$< 10^{-7}$	2	2	4	4

Table 5.5: Comparison between the distances for EGO-FACEBOOK matrix reduced W_3. The highlighted results indicate the best value for k according to each method.

the full-rank system is

$$\frac{m_2}{4n_2+16} = \frac{43967}{4 \cdot 2019 + 16} \approx 5.43,$$

while the CPU time performances are 73 seconds for the low-rank and 112 seconds for the gradient system. This means that the computational time has scaled between W_1 and W_2 approximately by a factor 16.5 for the low-rank system and by a factor 8.6 for the full-rank system.

Reduced matrix

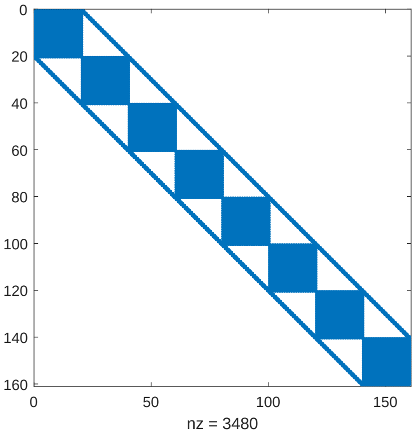
Now we focus on W_3. From its pattern it is clear that the most reasonable choices for the number of clusters k should be between $3, 4$ and 5. We also include $k = 6$ and we investigate the performances of the methods for these values. In this case all the methods agree and the best number of clusters is $k = 3$. The factor for the memory saving gain with respect to the full-rank system is

$$\frac{m_3}{4n_3+16} = \frac{19078}{4 \cdot 896 + 16} \approx 5.30,$$

while the CPU time performances are 23 seconds for the low-rank and 37 seconds for the gradient system.

5.6.4 An example with penalization: the Stochastic Block Model

The Stochastic Block Model (SBM) is a model of generating random graphs that tend to have communities. It is an important model in a wide range of fields, from sociology to physics. In this example we consider $n = 160$ vertices partitioned in $p = 8$ clusters $C_1, \ldots, C_p$ of $q = 20$ elements each. We consider a random full symmetric matrix



k	$g_k(W)$	$d_k^{\text{full}}(W)$	M_k^{full}	N_k^{full}
3	0.6488	6.2489	14	2560
4	0.7654	8.0190	12	2991
5	0.7654	6.1975	10	2542
6	0.6488	8.5668	11	1225
7	0.4335	5.6359	14	3437
8	**12.1650**	**11.5727**	26	2808
9	0.1522	0.7656	15	1859

FIGURE 5.3: SBM matrix: structural pattern (left) and comparison of the distances (right). The marked bold results indicate the best value for k according to each method.

$J \in \mathbb{R}^{q\times q}$ and we build the matrix

$$W = I_p \otimes J + B_p \otimes I_q, \qquad B_p = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & \ddots & 1 \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix} \in \mathbb{R}^{p\times p},$$

where $\otimes$ denotes the Kronecker product. The weight matrix generated has the pattern in Figure 5.3, with $m = 3480 \approx 21.75n$ non-zero entries and it has p blocks by construction. If we apply Algorithm 6, some values of k provide a non-admissible solution, which means that in this case a penalization is needed and the low-rank system (5.17) cannot be exploited. In particular for $k = 8$, which is one of the candidate optimal values, the non-negativity constraint violation cannot be ignored. In Figure 5.3 we show the results of the integration of the full-rank gradient system (5.8), where we introduce in the j-th *structured inner iteration* a penalization c_j that starts from $c_0 = 0$ and then increases by adding 0.5 during each iteration, that is $c_j = 0.5j$. The results found are admissible or slightly not, with the norm of $\min(W + \varepsilon_\star E_\star, 0)$ of order 10^{-5}: in the last case we ensure that the optimizer is admissible by removing this error. The time required by the computation is 33 seconds.

5.6.5 Comparison with a different graph partitioning algorithm

In this section we compare the clustering of the Graph Partitioning algorithm proposed in [43] with the result provided by the spectral clustering. More precisely we consider the examples of the previous sections and we compare the cost of the cut associated with the partitioning, for the most interesting values of k.

For a fixed k, the Graph Partitioning algorithm presented in [43] computes a partition $\mathcal{P} = \{V_1, \dots, V_k\}$ of the vertices set and it minimizes the cost function

$$\mathcal{C}(\mathcal{P}) = \sum_{h\neq l} \sum_{i\in V_h, j\in V_l} \hat{w}_{i,j},$$

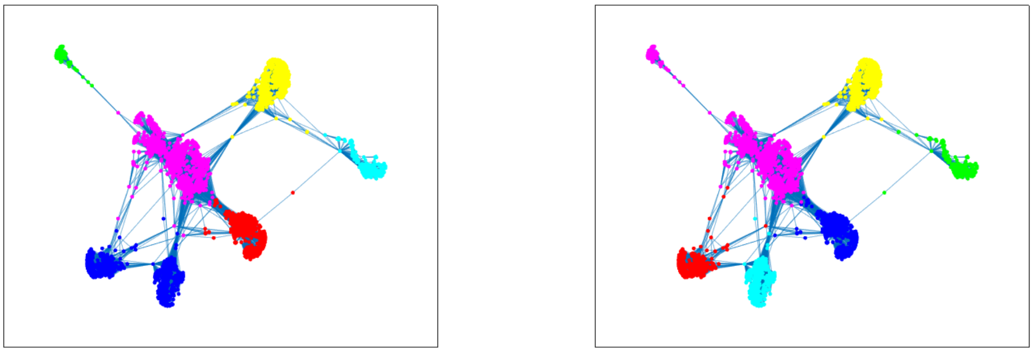


Figure 5.4: Clustering of the EGO-FACEBOOK graph for $k = 6$, on the left Spectral Clustering, while on the right Graph Partitioning.

	k	Graph Partitioning	Spectral Clustering
ECOLI	3	2.5699	2.4968
	4	6.7949	5.2607
	8	15.0172	11.5777
EGO-FACEBOOK	6	0.6228	0.5608
	7	0.8172	0.5933
SBM	8	10.8664	10.8664

Table 5.6: Cost function $\mathcal{C}(\mathcal{P})$ for the examples shown in the previous section.

where $\widehat{W} = (\hat{w}_{i,j})$ is the normalized weight matrix, i.e. the rows sum to 1. It can be shown (see [43, Lemma 2]) that a simple formula for the computation of the cost is given by

$$\mathcal{C}(\mathcal{P}) = \mathbb{1}^\top \widehat{W} \mathbb{1} - \mathrm{tr}(\Pi^\top \widehat{W} \Pi),$$

where $\Pi \in \mathbb{R}^{n,k}$ is the permutation matrix associated with the partitioning, that is

$$\Pi_{i,j} = \begin{cases} 1 & \text{if } i \in V_j \\ 0 & \text{if } i \notin V_j \end{cases} \qquad i = 1, \ldots, n, \quad j = 1, \ldots, k.$$

The minimization of $\mathcal{C}(\mathcal{P})$ is then performed by means of the spectral decomposition of $\hat{W}$, where the discrete partitions are obtained after applying the k-means algorithm to the retrieved eigenvectors. The algorithms provide similar partitionings (see for instance Figure 5.4), but the cost associated with the Spectral Clustering result is less or equal than that of the Graph Partitioning, as shown in Table 5.6 (the JOURNALS matrix has not been considered here, since the Graph Partitioning algorithm computes a large cluster that includes almost all the vertices and so the clustering is not meaningful).

Code and Data Availability

The codes implementing the algorithms discussed in this chapter are publicly available at:

https://github.com/StefanoSicilia/Spectral-Clustering-stability

Chapter 6

Low-rank-adaptive stabilization of a matrix

In this chapter we adapt the structured two-level method introduced in Chapter 2 and in Chapter 3 in order to stabilize a matrix, in the Hurwitz sense, by moving all its eigenvalues in the left open complex half-plane.

Let A be a square matrix with a given structure (e.g. real matrix, sparsity pattern, Toeplitz structure, etc.) and assume that it is unstable, i.e. at least one of its eigenvalues lies in the complex right half-plane. The problem of stabilizing A consists in the computation of a matrix B, whose eigenvalues have all negative real part and such that the perturbation $\Delta = B - A$ has infimal norm. The structured stabilization further requires that the perturbation preserves the structural pattern of A. This non-convex problem is solved by a two-level procedure which involves the computation of the stationary points of a matrix ODE. It is possible to exploit the underlying low-rank features of the problem by using a rank-adaptive integrator that follows rigidly the rank of the solution. Some benefits derived from the low-rank setting are shown in several numerical examples. These computational advantages also allow to deal with high dimensional problems.

The chapter is based on [38], which is a paper that extends the results of [31] and it generalizes them to the structured case by means of the method introduced in Chapter 4 (see [34]) and by using the rank-adaptive integrator proposed in [13] for the solution of the ODE.

6.1 Introduction

Given a matrix $A \in \mathbb{C}^{n\times n}$ with spectrum $\sigma(A) = \{\lambda_1, \dots, \lambda_n\}$ ordered such that

$$\mathrm{Re}(\lambda_n) \leq \cdots \leq \mathrm{Re}(\lambda_1),$$

we consider the problem of its stabilization, that is we look for a matrix B close to A such that B is a stable matrix, i.e. all its eigenvalues lie in the open left complex half-plane. This problem is well-known in the literature and it has been addressed with different methods (see e.g. [9, 20, 22, 59, 60]). In this chapter we follow the approach of [32] and we improve it by means of an implementation that highlights the low-rank features of the problem. We combine the approach presented in [34] with the rank-adaptive integrator of [13, 14] and we derive a new efficient method. We also introduce an alternative functional that exploits the cubic Hermite interpolating polynomial and we compare it with the one presented in [32]. The results of our new method are similar to the other approaches, but the novelty introduced also allows to deal with higher dimensional problems.

In order to obtain a strict stability, we fix a parameter $\delta > 0$, usually called stability margin (see e.g. [10, 56]), and we further require that the stabilized matrix B is δ-stable, that is all its eigenvalues lie in the set

$$\mathbb{C}_\delta^- = \{\lambda \in \mathbb{C} : \mathrm{Re}(\lambda) \leq -\delta\}.$$

Formally, we wish to find a perturbation Δ of minimum Frobenius norm such that $A + \Delta$ is δ-stable, that is we consider the optimization problem

$$\arg\min_{\sigma(A+\Delta)\subseteq\mathbb{C}_\delta^-} \|\Delta\|_F \tag{6.1}$$

and we look for its minimum and its minimizer(s). We denote by $m_\delta(A)$ the number of δ-unstable eigenvalues of A (i.e. the eigenvalues with real part larger than $-\delta$), which is zero if and only if A is δ-stable.

We focus on the most interesting case where $m_\delta(A)$ is moderate, since in many applications the original matrix is close to be stable and so generally just a few of its eigenvalues lie in the right closed complex half-plane. In this way it will be possible to exploit the low-rank features of the problem, by taking inspiration from what has been discussed in Section 3.1.2.

For solving problem (6.1), we use a two-level approach similar to the one presented in [32]. Given a fixed perturbation size $\varepsilon > 0$ and a predetermined parameter $\delta > 0$ (that ensures strict stability), we rewrite the matrix perturbation $\Delta = \varepsilon E$ with $\|E\|_F = 1$ and we minimize in the *inner iteration* the objective functional

$$F_\varepsilon(E) = \frac{1}{2}\sum_{i=1}^{n} \left((\mathrm{Re}\,(\lambda_i(A+\varepsilon E)) + \delta)_+\right)^2 = \frac{1}{2}\sum_{i=1}^{m_\delta(A+\varepsilon E)} (\mathrm{Re}\,(\lambda_i(A+\varepsilon E)) + \delta)^2, \tag{6.2}$$

where $a_+ = \max(a, 0)$ is the positive part of a real number a. Then the *outer iteration* tunes the perturbation size $\varepsilon > 0$ and finds the minimum value $\varepsilon_\star$ such that $F_{\varepsilon_\star}(E_\star(\varepsilon_\star)) = 0$, where

$$E_\star(\varepsilon) = \arg\min_{\|E\|_F=1} F_\varepsilon(E) \tag{6.3}$$

is the solution of the optimization problem that arises in the *inner iteration.*

While in [32] the number of addends of the summation in the objective functional is fixed and it is based on an initial guess of the amount of unstable eigenvalues, in our new approach the number of summands depends on E and relies on its current number of δ-unstable eigenvalues. This feature makes it possible to exploit the properties of the perturbation E with an adaptive choice of its rank. The optimizers of problem (6.1) are seen as stationary points of a gradient system, which is integrated through a rank-adaptive strategy based on the one presented in [13, 14].

The procedure described can be adapted also to the structured stabilization problem, that takes into account the structure of the original matrix and hence gives a more meaningful result; as far as we know, this problem has received much less attention in the literature than the unstructured. Given $A \in \mathcal{S}$, where $\mathcal{S} \subseteq \mathbb{C}^{n\times n}$ is a subspace of the complex matrices, we consider

$$\arg\min_{\sigma(A+\Delta)\subseteq\mathbb{C}_\delta^-,\ \Delta\in\mathcal{S}} \|\Delta\|_F, \tag{6.4}$$

that is the structured version of (6.1). Also in this case we proceed with the two-level approach and we show how the method can be reused in order to exploit all the low-rank properties that arise in the unconstrained problem.

The chapter is organized as follows. In Section 6.2 we introduce the gradient system for the *inner iteration* of the unstructured problem. In Section 6.3 we show the rank-adaptive integrator used for the solution of the gradient system. In Section 6.4 we investigate the behaviour of the rank during the integration of the gradient system and we show it in several numerical examples. The *outer iteration* is presented in Section 6.5, where an alternative functional for the *inner iteration* is introduced. In Section 6.6 we adapt the procedure to the structured case and in Section 6.7 we show some numerical examples of large dimension.

6.2 Gradient system

In this section we fix the perturbation size $\varepsilon > 0$ and we describe an ordinary differential equation that is used to solve problem (6.3). We follow the same approach shown in Chapter 2 and we adapt the results of Lemma 2.3.2, Theorem 2.3.4 and Lemma 2.3.1.

In the following, given a matrix whose eigenvalues are simple, we denote by x_i and y_i, respectively, the unit left and right eigenvectors associated to λ_i, such that $x_i^* y_i$ is real and non-negative. We will often exploit the following standard perturbation result for eigenvalues (see [50]), which we have stated also in Chapter 4 as Lemma 4.2.1.

Lemma 6.2.1. *Let $\lambda(t)$ be a simple eigenvalue of a differentiable matrix path $A(t)$ in a neighbourhood of t_0 and let $x(t)$ and $y(t)$ be, respectively, the left and right unit eigenvectors associated. Then $x(t_0)^* y(t_0) \neq 0$ and*

$$\dot{\lambda}(t_0) = \frac{x(t_0)^* \dot{A}(t_0) y(t_0)}{x(t_0)^* y(t_0)}.$$

We will always suppose that the hypothesis of Lemma 6.2.1 holds true in our setting. This is a generic assumption in the unstructured perturbation case, since the matrices with at least a 2-by-2 Jordan block form a set of zero measure in $\mathbb{C}^{n\times n}$, see e.g. [4] for more details.

In order to find an optimal value of E that minimizes the objective functional $F_\varepsilon(E)$, we introduce a matrix differentiable path $E(t)$ of unit Frobenius norm matrices that depends on a real non-negative time variable $t \geq 0$. By denoting $\mathbb{S}_1$ the unit norm sphere in $\mathbb{C}^{n\times n}$

$$\mathbb{S}_1 = \left\{ A \in \mathbb{C}^{n\times n} : \|A\|_F = 1 \right\},$$

it holds that $E(t) \subseteq \mathbb{S}_1$. In this way it is possible to consider the continuous version $F_\varepsilon(E(t))$ of the objective functional, whose derivative is characterized by the following result.

Lemma 6.2.2. *Let $E(t) \subseteq \mathbb{S}_1$ be a differentiable path of matrices for $t \in [0, +\infty)$. Let ε and δ be fixed. Then $F_\varepsilon(E)$ is differentiable in $[0, +\infty)$ with*

$$\frac{\mathrm{d}}{\mathrm{d}t} F_\varepsilon(E(t)) = \varepsilon \operatorname{Re}\langle G_\varepsilon(E(t)), \dot{E}(t)\rangle,$$

where $G_\varepsilon(E(t))$ is the gradient

$$G_\varepsilon(E(t)) = \sum_{i=1}^{n} \gamma_i(t) x_i(t) y_i(t)^*, \qquad \gamma_i(t) = \frac{\left(\operatorname{Re}\left(\lambda_i(A + \varepsilon E(t))\right) + \delta\right)_+}{x_i(t)^* y_i(t)} \in \mathbb{R}.$$

Proof. It is a direct adaptation of Lemma 2.3.1. Since

$$u^*Bv = \langle uv^*, B\rangle, \qquad \forall B \in \mathbb{C}^{n\times n}, \quad \forall u, v \in \mathbb{C}^n$$

and $((x_+)^2)' = 2x_+$, Lemma 6.2.1 implies

$$\frac{\mathrm{d}}{\mathrm{d}t}F_\varepsilon(E) = \sum_{i=0}^{n} \mathrm{Re}\left(\frac{\varepsilon x_i^*\dot{E}y_i}{x_i^*y_i}\right)(\mathrm{Re}\,(\lambda_i(A+\varepsilon E)+\delta))_+ =$$

$$= \varepsilon\,\mathrm{Re}\left\langle \sum_{i=1}^{n} \gamma_i x_i y_i^*, \dot{E}\right\rangle = \varepsilon\,\mathrm{Re}\langle G_\varepsilon(E), \dot{E}\rangle.$$

□

The matrix $G = G_\varepsilon(E(t))$ introduced in Lemma 6.2.2 gives the steepest descent direction for minimizing the objective functional, without considering the constraint on the norm of E. Differently from the analogous formulas for the objective functional derivative in [32, 34, 37], where the gradient G has less or none dynamical changes in the rank, the gradient introduced here is a matrix whose rank strongly depends on the δ-unstable eigenvalues of $A + \varepsilon E(t)$ and hence it may change in time. This feature is relevant, since it implies low-rank properties in the problem we focus on. Indeed, if the number of unstable eigenvalues of A is moderate, that is the case of our interest, the gradient G has low-rank by construction. The next result shows the best direction to follow in order to fulfil the unit norm condition, which is equivalent to $\mathrm{Re}\langle E, \dot{E}\rangle = 0$.

Lemma 6.2.3. *Given $E \in \mathbb{S}_1$ and $G \in \mathbb{C}^{n\times n} \setminus \{0\}$, the solution of the optimization problem*

$$\underset{Z\in\mathbb{S}_1,\ \mathrm{Re}\langle Z,E\rangle=0}{\arg\min} \quad \mathrm{Re}\langle G, Z\rangle$$

is

$$\alpha Z_\star = -G + \mathrm{Re}\langle G, E\rangle E,$$

where $\alpha > 0$ is the normalization parameter.

Proof. The proof is a direct consequence of Lemma 2.3.2. □

Lemmas 6.2.2 and 6.2.3 suggest to consider the matrix ordinary differential equation

$$\dot{E}(t) = -G_\varepsilon(E(t)) + \mathrm{Re}\langle G_\varepsilon(E(t)), E(t)\rangle E(t), \tag{6.5}$$

whose stationary points are zeros of the derivative of the objective functional $F_\varepsilon(E(t))$. Lemma 6.2.2 implies that equation (6.5) is a gradient system for $F_\varepsilon(E(t))$, since along its trajectories

$$\frac{\mathrm{d}}{\mathrm{d}t}F_\varepsilon(E(t)) = \varepsilon\langle G_\varepsilon(E(t)), \dot{E}(t)\rangle = \varepsilon\left(-\|G_\varepsilon(E(t))\|_F^2 + (\langle G_\varepsilon(E(t)), E(t)\rangle)^2\right) \leq 0$$

by means of the Cauchy-Schwarz inequality, which also implies that the derivative vanishes in $E_\star$ if and only if $E_\star$ is a real multiple of $G_\varepsilon(E_\star)$. Thanks to the monotonicity property along the trajectories, an integration of this gradient system must lead to a stationary point $E_\star$. The next result provides the important property of the minimizers that reveals the underlying low-rank structure of the problem.

Theorem 6.2.4. *Let $E_\star$ be a stationary point of (6.5), such that $F_\varepsilon(E_\star) > 0$. Then $G_\varepsilon(E_\star) \neq 0$ and $E_\star$ is a real multiple of $G_\varepsilon(E_\star)$, that is there exists $\nu \neq 0$ and an* *integer* $1 \leq m \leq n$ *such that*

$$E_\star = \nu G(E_\star) = \nu \sum_{i=1}^{m} \gamma_i x_i y_i^*.$$

Proof. For all $E \in \mathbb{S}_1$, and with the same notations of Lemma 6.2.2, it holds that

$$\mathrm{Re}\langle G_\varepsilon(E), A + \varepsilon E \rangle = \mathrm{Re}\left\langle \sum_{i=1}^{m} \gamma_i x_i y_i^*, A + \varepsilon E \right\rangle = \mathrm{Re}\left(\sum_{i=1}^{m} \gamma_i x_i^* (A + \varepsilon E) y_i \right) =$$

$$= \sum_{i=1}^{m} \mathrm{Re}(\gamma_i \lambda_i x_i^* y_i) = \sum_{i=1}^{m} \mathrm{Re}(\lambda_i) \gamma_i (x_i^* y_i) \geq 0,$$

which means that the gradient $G_\varepsilon(E)$ vanishes if and only if $A + \varepsilon E$ is δ-stable, which is not the case since $F_\varepsilon(E_\star) > 0$. Thus the claim follows from the fact that the right hand side of (6.5) vanishes for $E = E_\star$, implying that $E_\star$ must be a real multiple of the non-zero matrix $G_\varepsilon(E_\star)$. □

Theorem 6.2.4 together with the monotonicity property, show that the integration of equation (6.5) always lead to a low-rank stationary point.

6.3 A rank-adaptive integrator for the gradient system

In this section we discuss how to integrate system (6.5). Let us assume that for all t the rank $r(t)$ of the matrix $E(t) \in \mathbb{C}^{n \times n}$ is piecewise constant, that is in agreement with the changing rank of the gradient. This means that in an interval where $r(t) \equiv r$ is constant, $E(t)$ can be decomposed as an analytic SVD-like (see e.g. [8])

$$E(t) = U(t)S(t)V(t)^*, \quad U(t), V(t) \in \mathbb{C}^{n \times r}, \quad S(t) \in \mathbb{C}^{r \times r} \text{ invertible.}$$

This decomposition generalizes the SVD since it is not required that the matrix S is diagonal and, up to the points of discontinuity of $r(t)$, it can be extended to the whole trajectory of E. In particular, in a neighbourhood of a stationary point, we expect that the rank $r(t)$ is constant. We can rewrite equation (6.5) as

$$\dot{U}SV^* + U\dot{S}V^* + US\dot{V}^* = -G + \mu USV^*,$$

where $G = G_\varepsilon(E)$ and $\mu = \mathrm{Re}\langle G, E \rangle$. Imposing the gauge conditions $U^*\dot{U} = V^*\dot{V} = 0$, similarly as done in system (5.17) or in [37, Equation 14], yields

$$\begin{cases} \dot{U} = -(I - UU^*)GVS^{-1} \\ \dot{S} = -U^*GV + \mu S \\ \dot{V} = (I - VV^*)G^*US^{-*} \end{cases}, \tag{6.6}$$

which is equivalent to (6.5). The structure of the matrix S is not diagonal in general, but the decomposition assumed for E holds along all the trajectory. System (6.6) consists of two matrix ODEs of dimension n-by-r and one of dimension r-by-r, where r is the rank of $E(t)$.

A classical integration, e.g. by means of Euler's method, of system (6.6) is not suitable. Indeed the presence of S^{-1} may cause numerical issues and moreover this integration does not capture the change of the rank of G along the trajectory. To overcome this problem, we exploit a rank-adaptive strategy similar to the one exposed in [13]. We define

$$g(E) := -G_\varepsilon(E) + \mathrm{Re}\langle G_\varepsilon(E), E\rangle E$$

as the left-hand side of (6.5) such that $\dot{E} = g(E)$ and we fix a tolerance τ. To update the perturbation path $E(t) = U(t)S(t)V(t)^*$ from t_0 to t_1, we start from $E_0 = E(t_0) = U(t_0)S(t_0)V(t_0)^* = U_0S_0V_0^*$ of rank r_0 and we get $E_1 = E(t_1) = U(t_1)S(t_1)V(t_1)^* = U_1S_1V_1^*$ of rank r_1 by performing the following steps:

1. Set $\rho = \min(2r_0, n)$. This choice differs a bit from the one in [13], where $\rho = 2r_0$, and it is made in order to avoid that the dimension of the augmented basis exceeds the dimension of the space. This fact is unlikely, but it cannot be excluded a priori.

2. Compute augmented basis $\widehat{U} \in \mathbb{C}^{n\times\rho}$ and $\widehat{V} \in \mathbb{C}^{n\times\rho}$

   K-step: Integrate from t_0 to t_1 the n-by-r_0 differential equation

   $$K(t_0) = U_0S_0, \qquad \dot{K}(t) = f\left(K(t)V_0^*\right)V_0.$$

   Perform a QR factorization of $(K(t_1), U_0)$, save the first ρ columns in $\widehat{U} \in \mathbb{C}^{n\times\rho}$ and compute $\widehat{M} = \widehat{U}^*U_0 \in \mathbb{C}^{\rho\times r_0}$.

   L-step: Integrate from t_0 to t_1 the n-by-r_0 differential equation

   $$L(t_0) = V_0S_0^*, \qquad \dot{L}(t) = f\left(U_0L(t)^*\right)^*U_0.$$

   Perform a QR factorization of $(L(t_1), V_0)$, save the first ρ columns in $\widehat{V} \in \mathbb{C}^{n\times\rho}$ and compute $\widehat{N} = \widehat{V}^*V_0 \in \mathbb{C}^{\rho\times r_0}$.

3. Augment and update S

   S-step: Integrate from t_0 to t_1 the ρ-by-ρ differential equation

   $$\widehat{S}(t_0) = \widehat{M}S_0\widehat{N}^*, \qquad \dot{\widehat{S}}(t) = \widehat{U}^*f\left(\widehat{U}\widehat{S}(t)\widehat{V}^*\right)\widehat{V}.$$

4. Adapt the rank for the updated matrices

   Truncation: Compute the SVD $\widehat{S}(t_1) = \widehat{P}\widehat{\Sigma}\widehat{Q}^*$, where $\widehat{\Sigma} = \mathrm{diag}(\sigma_1, \ldots \sigma_\rho)$, and choose the new rank $r_1 \le \rho$ such that

   $$\left(\sum_{i=r_1+1}^{\rho} \sigma_i^2\right)^{\frac{1}{2}} \le \tau.$$

   Define S_1 as the r_1-by-r_1 diagonal main sub-matrix of $\widehat{\Sigma}$ and denote by $P_1, Q_1 \in \mathbb{C}^{\rho\times r_1}$ the first r_1 columns of $\widehat{P}$ and $\widehat{Q}$ respectively.

5. Return $U_1 = \widehat{U}P_1 \in \mathbb{C}^{n\times r_1}$, $V_1 = \widehat{V}Q_1 \in \mathbb{C}^{n\times r_1}$ and $S_1 \in \mathbb{C}^{r_1\times r_1}$.

In [13], it is shown that this algorithm computes an approximation of $U(t), S(t)$ and $V(t)$ with an error proportional to the tolerance τ and the time step $t_1 - t_0$. This integration method is ideal for its structure-preserving properties in order to maintain the main features of the gradient system (6.5), that is the monotonicity of the objective functional and the constraint on the unit Frobenius norm (see [13, Section 4]). Moreover this algorithm allows the truncation of the rank, according to the tolerance

τ	$\varepsilon = 2$		$\varepsilon = 2.38$	
	$F_\varepsilon(E_\star(\varepsilon))$	max rank	$F_\varepsilon(E_\star(\varepsilon))$	max rank
10^{-1}	0.6981	6	0.2501	6
10^{-2}	0.4695	6	0.1600	6
10^{-3}	0.2408	6	0.0283	7
10^{-4}	0.2062	8	0.0039	8
10^{-5}	0.2450	9	0.0001	9
10^{-6}	0.2458	9	$8.4212 \cdot 10^{-6}$	9

TABLE 6.1: Illustrative matrix (6.7): numerical results of the *inner iteration*

τ in order to adapt the size of the invertible matrix S along the trajectory. These properties make the rank-adaptive integrator a suitable choice for the computation of the solution of the gradient system (6.5) and strongly motivate the usage of this approach for the integration of this ODE.

6.4 Rank adaptivity for fixed perturbation size

In this section we show with three different examples how the integrator introduced in Section 6.3 chooses the adaptive rank of the perturbation. Unless otherwise stated, we set the parameter $\delta = 10^{-3}$. We consider here illustrative examples with small dimension n, while larger examples will be presented in Section 6.7.

6.4.1 An illustrative example

Consider the matrix

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & -1 & 0 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & -1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 1 & -1 & 0 & 0 & 1 & 1 & 0 \\ -1 & 1 & -1 & 1 & 1 & 0 & -1 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \in \mathbb{C}^{10\times 10}, \qquad (6.7)$$

which has 6 unstable eigenvalues. Table 6.1 contains the results of the functional $F_\varepsilon(E_\star)$ and the maximum rank of $E(t)$ achieved during the *inner iteration* with different choices of ε and of the tolerance τ; in particular $\varepsilon = 2.38$ is close to be $\varepsilon_\star$. In both cases it is possible to observe how much the value of the objective function decreases when the tolerance is lowered, which is explained by the higher adaptive rank of the perturbation. This means that lower values of the tolerance lead to more accurate results, but they also increase the rank of the perturbation and hence the computational cost. Thus, a suitable choice of τ is crucial to balance these two factors.

τ	$\varepsilon = 4$		$\varepsilon = 5.5$	
	$F_\varepsilon(E_\star(\varepsilon))$	max rank	$F_\varepsilon(E_\star(\varepsilon))$	max rank
10^{-1}	3.7646	20	0.2321	20
10^{-2}	3.7646	20	0.2469	20
10^{-3}	3.7646	20	0.0306	20
10^{-4}	3.7646	20	0.0487	20
10^{-5}	3.7646	20	0.0330	20
10^{-6}	3.7646	20	0.0292	20

Table 6.2: Grcar matrix (6.8): numerical results of the *inner iteration*

6.4.2 Grcar matrix

The rank-adaptive procedure is not effective in all cases. For instance let us consider the Grcar n-by-n matrix, with $n \geq 5$, that is a Hessenberg and Toeplitz matrix of the form

$$G_n = \begin{pmatrix} 1 & 1 & 1 & 1 & & & \\ -1 & \ddots & \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \ddots & 1 & \\ & & \ddots & \ddots & \ddots & 1 & \\ & & & \ddots & \ddots & 1 & \\ & & & & -1 & 1 & \end{pmatrix} \in \mathbb{R}^{n\times n}. \tag{6.8}$$

The eigenvalues of G_n are all unstable and also quite sensitive. We consider $n = 20$ and we show in Table 6.2 the results of the *inner iteration* with different choices of ε and of the tolerance τ.

In this case the fact that all the eigenvalues are unstable provides a perturbation of full (or almost full) rank and the rank-adaptive integrator does not seem to be effective. During the integration, some eigenvalues may become stable before the other ones and after that they begin to cross back and forth the imaginary axis. Consequently they activate or disable their respective rank-1 component in the gradient, producing a minimal change of the rank (e.g. 18 or 19) that is not worth to exploit with the adaptive integrator.

6.4.3 Smoke matrix

In this example we show in more detail the changes of the rank perturbation during the integration. Let S_n be the $n \times n$ Smoke matrix from the `gallery` function of Matlab

$$S_n = \begin{pmatrix} \zeta_1 & 1 & & & & \\ & \zeta_2 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & 1 & \\ & & & & \zeta_{n-1} & 1 \\ 1 & & & & & \zeta_n \end{pmatrix} \in \mathbb{R}^{n\times n}, \tag{6.9}$$

whose diagonal contains the n distinct roots of the unit

$$\zeta_j = \mathrm{e}^{\frac{2\pi \mathrm{i} j}{n}}, \qquad j = 1, \dots, n$$

τ	$\varepsilon = 2.5$	
	$F_\varepsilon(E_\star(\varepsilon))$	max rank
10^{-1}	0.1916	11
10^{-2}	0.0498	12
10^{-3}	0.0061	13
10^{-4}	0.0011	15
10^{-5}	0.0002	16
10^{-6}	$5.0540 \cdot 10^{-6}$	17

Table 6.3: Smoke matrix (6.9): numerical results of the *inner iteration*

such that $\zeta_j^n = 1$ for all j. The characteristic polynomial $p(\lambda) = \det(S_n - \lambda I_n)$ associated to S_n is

$$p(\lambda) = (-1)^{n+1} + \prod_{j=1}^{n}(\zeta_j - \lambda) = (-1)^n \left(-1 + \prod_{j=1}^{n}(\lambda - \zeta_j) \right) = (-1)^n(\lambda^n - 2)$$

and hence the eigenvalues of S are equally distributed along the circle of radius $\sqrt[n]{2}$. For n even the matrix S_n has half eigenvalues stable and half unstable. We consider $n = 20$ and we collect the results of the *inner iteration* in Table 6.3.

Figure 6.1 shows the trend of the rank and the objective functional. We can notice how the rank is increased or decreased by the integrator, but definitely it stabilizes when it approaches the stationary point.

6.5 Outer iteration: tuning the perturbation size

Once that a computation of the optimizers is available for a given $\varepsilon > 0$, we need to determine an optimal value for the perturbation size $\varepsilon_\star$. Let $E_\star(\varepsilon)$ be a solution of the optimization problem (6.3) and consider the function

$$\varphi(\varepsilon) := F_\varepsilon(E_\star(\varepsilon)).$$

This function is non-negative and we define $\varepsilon_\star$ as the smallest zero of φ. Assuming that the unstable eigenvalues of $A + \varepsilon E_\star(\varepsilon)$ are simple (see also Remark 2.4.2), for $0 \leq \varepsilon < \varepsilon_\star$, yields that φ is a differentiable function in the interval $[0, \varepsilon_\star)$. The aim of the *outer iteration* is to approximate $\varepsilon_\star$, which is the solution of the optimization problem (6.1). In order to solve this problem, we use a combination of the well-known Newton and bisection methods, which provides an approach similar to [24, 28, 33] or [34]. If the current approximation ε is smaller than $\varepsilon_\star$, it is possible to exploit Newton's method, since φ is smooth there; otherwise, if $\varepsilon > \varepsilon_\star$, we use the bisection method. The following result provides a simple formula for the first derivative of φ which is cheap to compute, making the Newton's method easy to apply.

Lemma 6.5.1. *For $0 \leq \varepsilon < \varepsilon_\star$ it holds that*

$$\varphi'(\varepsilon) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} F_\varepsilon(E_\star(\varepsilon)) = \langle G_\varepsilon(E_\star(\varepsilon)), E_\star(\varepsilon) \rangle = -\|G_\varepsilon(E_\star)\|_F \leq 0.$$

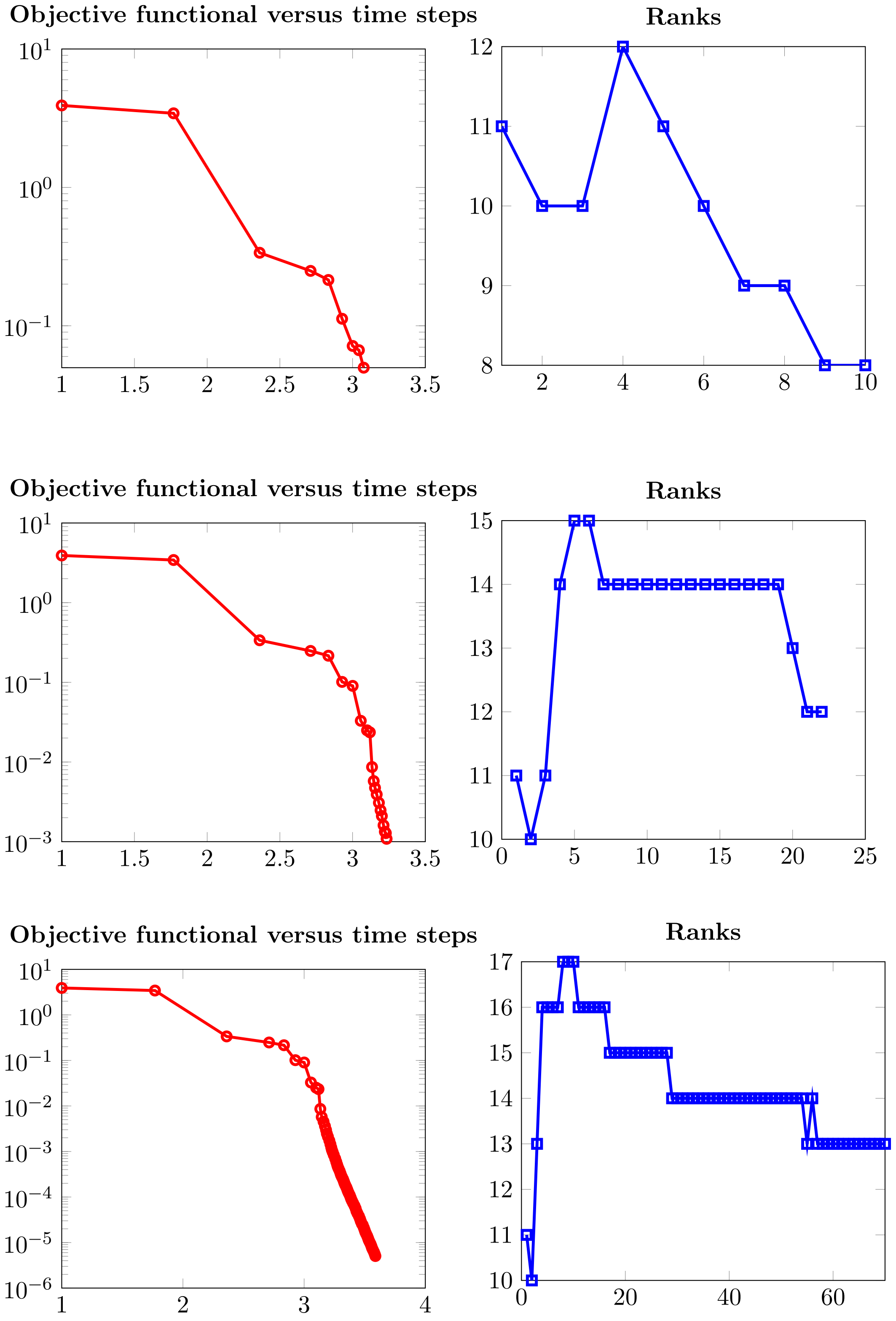


FIGURE 6.1: Smoke matrix: functional and ranks in the *inner iteration* for $\tau = 10^{-2}$ (up), $\tau = 10^{-4}$ (middle) and $\tau = 10^{-6}$ (down).

Proof. It is similar to the proof of Lemma 2.4.1. As shown in Lemma 6.2.2, we get

$$\varphi'(\varepsilon) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\left(\frac{1}{2}\sum_{i=1}^{n}\left((\mathrm{Re}\,(\lambda_i(A+\varepsilon E_\star(\varepsilon))) + \delta)_+\right)^2\right) =$$

$$= \sum_{i=1}^{n}(\mathrm{Re}\,(\lambda_i(A+\varepsilon E_\star(\varepsilon))) + \delta)_+ \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\left(\mathrm{Re}\,(\lambda_i(A+\varepsilon E_\star(\varepsilon)))_+\right) =$$

$$= \mathrm{Re}\langle G_\varepsilon(E_\star(\varepsilon)), E_\star(\varepsilon) + \varepsilon E_\star'(\varepsilon)\rangle,$$

where $E_\star'(\varepsilon)$ is the derivative with respect to ε of $E_\star(\varepsilon)$. Since $E_\star$ is a unit norm stationary point of (6.5) and a zero of the derivative of the objective functional F_ε, then $G_\varepsilon(E_\star(\varepsilon))$ is a negative multiple of $E_\star$. Thus $G_\varepsilon(E_\star) = -\|G_\varepsilon(E_\star)\|_F\, E_\star(\varepsilon)$ and, since $\|E_\star(\varepsilon)\|_F = 1$ for all ε (omitted in the following formula), it holds that

$$\mathrm{Re}\langle G_\varepsilon(E_\star), E_\star'\rangle = -\|G_\varepsilon(E_\star)\|_F \cdot \mathrm{Re}\langle E_\star, E_\star'\rangle = -\frac{\|G_\varepsilon(E_\star))\|_F}{2}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\|E_\star\|_F^2 = 0.$$

□

6.5.1 Another functional

For a deeper analysis of the method developed, we introduce also the alternative objective functional Φ_ε. Given $0 < \delta_1 < \delta_2 \ll 1$ we define it as

$$\Phi_\varepsilon(E) = \frac{1}{2}\sum_{i=1}^{n}\psi(\rho_i)\,(\rho_i + \delta_2)_+^2\,, \qquad \rho_i = \mathrm{Re}(\lambda_i(A+\varepsilon E)),$$

where $\psi \in C^1(\mathbb{R})$ is the cubic Hermite interpolating polynomial defined as

$$\psi(x) = \begin{cases} 0 & x < -\delta_2 \\ \frac{(x+\delta_2)^2(2x+3\delta_1-\delta_2)}{(\delta_1-\delta_2)^3} & -\delta_2 \le x \le -\delta_1 \\ 1 & x > -\delta_1 \end{cases},$$

with derivative

$$\psi'(x) = \begin{cases} \frac{6(x+\delta_2)(x+\delta_1)}{(\delta_1-\delta_2)^3} & -\delta_2 \le x \le -\delta_1 \\ 0 & \text{otherwise} \end{cases},$$

such that

$$\psi(-\delta_1) = 1, \qquad \psi(-\delta_2) = \psi'(-\delta_1) = \psi'(-\delta_2) = 0.$$

The aim of this functional Φ_ε is to soften the passage of an eigenvalue from unstable to stable and vice-versa, by smoothening the function $(x+\delta_2)_+$ into $\psi(x)(x+\delta_2)$. In this way the interval $[-\delta_2, -\delta_1]$ defined by the new parameters δ_1 and δ_2 acts like a transition region that covers the non-differentiable point of the original functional F_ε. Lemma 6.5.2 shows that the previous theory can be applied also in this case, by means of a slight change of the gradient associated to the objective functional. In particular the new gradient is made up by the same rank-one perturbations of the previous version, but the coefficients associated are different.

Lemma 6.5.2. *Let $E(t) \subseteq \mathbb{S}_1$ be a differentiable path of matrices for $t \in [0, +\infty)$. Let ε and δ be fixed. Then $\Phi_\varepsilon(E)$ is differentiable in $[0, +\infty)$ with*

$$\frac{\mathrm{d}}{\mathrm{d}t}\Phi_\varepsilon(E(t)) = \varepsilon \operatorname{Re}\langle \Gamma_\varepsilon(E(t)), \dot{E}(t)\rangle,$$

where $\Gamma_\varepsilon(E(t))$ is the gradient of Φ_ε

$$\Gamma_\varepsilon(E(t)) = \sum_{i=1}^{n} \kappa_i(t) x_i(t) y_i(t)^*, \quad \kappa_i(t) = \frac{(\rho_i + \delta_2)_+ \left(\psi'(\rho_i)(\rho_i + \delta_2)_+ + 2\psi(\rho_i)\right)}{2x_i(t)^* y_i(t)} \geq 0,$$

with $\rho_i(t) = \operatorname{Re}\left(\lambda_i(A + \varepsilon E(t))\right)$.

Proof. The proof is the same as that of Lemma 6.2.2, since for any $i = 1, \ldots, n$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\psi(\rho_i)(\rho_i + \delta_2)_+^2\right) = \dot{\rho}_i \psi'(\rho_i)(\rho_i + \delta_2)_+^2 + 2\psi(\rho_i)\dot{\rho}_i(\rho_i + \delta_2)_+ =$$

$$= \dot{\rho}_i(\rho_i + \delta_2)_+ \left(\psi'(\rho_i)(\rho_i + \delta_2)_+ + 2\psi(\rho_i)\right) = \langle \kappa_i x_i y_i^*, \dot{E}\rangle.$$

□

Also the *outer iteration* can be adapted to the functional Φ_ε. In practice we have always considered $\delta_2 = 2\delta_1$ and $\delta = \delta_1 = 10^{-3}$, but, depending on the case, these parameters may be tuned differently. We recall that, for both the functionals F_ε and Φ_ε, the *outer iteration* aims to achieve $\operatorname{Re}(\lambda_1(A + \varepsilon_\star E_\star(\varepsilon_\star))) < -\delta$, instead of $\operatorname{Re}(\lambda_1(A + \varepsilon_\star E_\star(\varepsilon_\star))) < 0$. In all tables of Section 6.5 and Section 6.7 we write $\operatorname{Re}(\lambda_1) = \operatorname{Re}(\lambda_1(A + \varepsilon_\star E_\star(\varepsilon_\star)))$ for short.

6.5.2 Smoke matrix

Let us consider again the Smoke matrix $S \in \mathbb{R}^{n \times n}$. For $n = 30$ we generate a random orthogonal matrix by selecting the first factor of the QR decomposition of a matrix whose entries follows the standard normal distribution (in MATLAB notation we set `rng(1)` and $[Q, \sim] = \texttt{qr}\,(\texttt{randn}(n) + \mathrm{i} \cdot \texttt{randn}(n)))$ and we apply the algorithm on the Smoke-like matrix

$$A = QSQ^*. \tag{6.10}$$

In Table 6.5 we consider both the functionals F_ε and Φ_ε and we report the results for different integration strategies: we study the difference between the rank-adaptive approach and the fixed-rank method proposed in [32] with several values of the rank r. In all the experiments we set

$$\tau_{\text{inn}} = 10^{-9}, \quad \tau_{\text{out}} = 10^{-9}, \quad \tau_{\text{rk}} = 10^{-8}, \qquad \text{maxit}_{\text{inn}} = 150, \quad \text{maxit}_{\text{out}} = 200,$$

where τ stands for tolerance, maxit for the maximum number of iteration allowed and inn and out refers to the *inner* and *outer iterations* respectively.

For the standard functional F_ε, the rank-adaptive integrator provides the smallest distance $\varepsilon_\star = 3.2613$, even though it is slightly slower than the fixed-rank approaches. For the Hermite functional Φ_ε the best results is given by the fixed-rank 23 approach (that is $\varepsilon_\star = 3.3304$), but it is very close to the result provided by the adaptive integrator (that is $\varepsilon_\star = 3.3307$). The rank-adaptive approach provides one of the best results in similar computational time and it detects quite well the rank of the best optimizer found by the fixed-rank procedures. In this way it is possible to avoid to

	Rank feature	$\varepsilon_\star$	$\mathrm{rk}(E_\star)$	Functional	$\mathrm{Re}(\lambda_1)$	Time (s)
F_ε	Adaptive	**3.2613**	21	$4.4194 \cdot 10^{-10}$	$-9.7821 \cdot 10^{-4}$	9.3625
	Fixed (15)	3.7332	**15**	$1.6790 \cdot 10^{-14}$	-0.0010	6.1151
	Fixed (16)	3.5529	16	$1.6452 \cdot 10^{-12}$	-0.0010	7.0870
	Fixed (17)	3.3478	17	$2.8299 \cdot 10^{-11}$	$-9.9248 \cdot 10^{-4}$	5.5593
	Fixed (18)	3.4473	18	$2.3842 \cdot 10^{-11}$	$-9.9324 \cdot 10^{-4}$	8.8625
	Fixed (19)	3.3726	19	$1.6593 \cdot 10^{-11}$	$-9.9560 \cdot 10^{-4}$	5.5715
	Fixed (20)	3.3364	20	$1.5095 \cdot 10^{-11}$	-0.0010	**4.9986**
	Fixed (21)	3.2845	21	$3.4372 \cdot 10^{-11}$	$-9.9464 \cdot 10^{-4}$	5.8921
	Fixed (22)	3.2867	22	$1.5229 \cdot 10^{-11}$	$-9.9494 \cdot 10^{-4}$	6.7260
	Fixed (23)	3.2960	23	$4.2070 \cdot 10^{-10}$	$-9.8680 \cdot 10^{-4}$	7.4120
Φ_ε	Adaptive	3.3307	21	$1.0000 \cdot 10^{-9}$	-0.0020	7.3604
	Fixed (15)	4.0971	**15**	**0**	$\mathbf{-0.0066}$	15.6724
	Fixed (16)	3.5358	16	$1.0000 \cdot 10^{-9}$	-0.0020	11.2574
	Fixed (17)	3.3671	17	$1.0000 \cdot 10^{-9}$	-0.0020	8.2343
	Fixed (18)	3.4353	18	$1.2461 \cdot 10^{-7}$	$-8.6848 \cdot 10^{-4}$	19.6561
	Fixed (19)	3.3453	19	$1.0000 \cdot 10^{-9}$	-0.0020	8.3218
	Fixed (20)	3.3603	20	$8.4282 \cdot 10^{-8}$	-0.0013	16.1668
	Fixed (21)	3.3530	21	$2.7206 \cdot 10^{-8}$	-0.0012	16.3811
	Fixed (22)	3.3742	22	$1.0000 \cdot 10^{-9}$	-0.0020	8.7161
	Fixed (23)	3.3304	23	$1.0000 \cdot 10^{-9}$	-0.0020	14.4259

Table 6.4: Comparison between the rank-adaptive and fixed-rank integrators of the Smoke-like matrix (6.10) for both the functionals F_ε and Φ_ε. Best results highlighted in bold.

perform several times the fixed-rank integration in order to improve the result, which is more expensive especially when a GPU is not available.

In all the experiments we observed an alignment along the imaginary axis of the unstable eigenvalues, as shown in Figure 6.2. This means that all the algorithms move the unstable eigenvalues towards the imaginary axis, but they also change some of the stable ones making them closer to have zero real part. This is the reason behind the fact that the optimal fixed-rank (between 21 and 23 in this example) is not the original number of unstable eigenvalues (15).

In Table 6.5 we have compared the results of our algorithm with other competitors: the methods by Gillis and Sharma (G-S) developed in [20] and the approach by Noferini and Poloni (N-P) proposed in [59]. In this case, in order to have a fair comparison,

	Rk feature	$\varepsilon_\star$	$\mathrm{rk}(E_\star)$	Functional	$\mathrm{Re}(\lambda_1)$	Time (s)
F_ε	Adaptive	3.2744	**21**	$5.5126 \cdot 10^{-11}$	$6.8931 \cdot 10^{-6}$	8.2525
G-S	BCD	3.5757	30	**0**	$-9.5148 \cdot 10^{-15}$	1.4973
	Grad	3.1732	30	$3.5818 \cdot 10^{-30}$	$2.6765 \cdot 10^{-15}$	1.4723
	FGM	2.8263	30	$4.9565 \cdot 10^{-26}$	$3.1485 \cdot 10^{-13}$	**1.4644**
N-P	Hurw-Cpx	**0.7803**	29	**0**	**0**	4.2401

Table 6.5: Comparison between the rank-adaptive approach and the algorithms by Gillis and Sharma (G-S) and that by Noferini and Poloni for the Smoke-like matrix (6.10). Best results highlighted in bold.

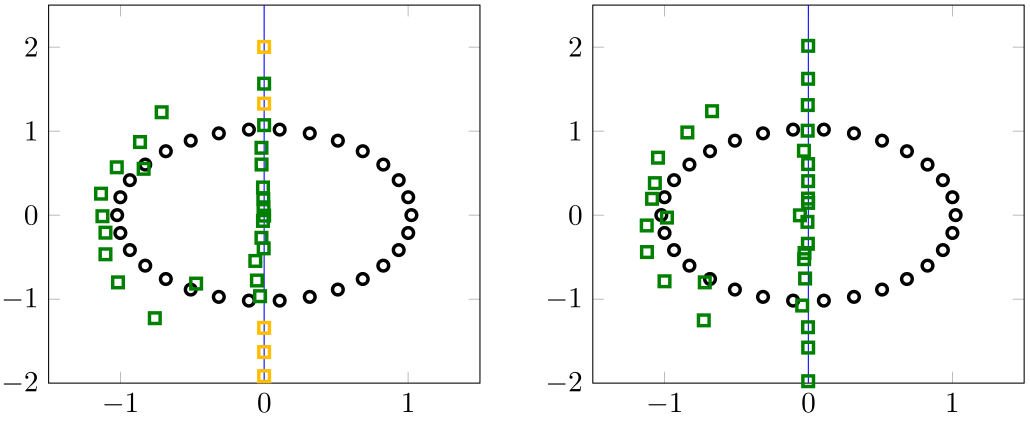


FIGURE 6.2: Smoke-like matrix (6.10): original eigenvalues (black circles), stabilized ones (if $\mathrm{Re}(\lambda) < -\delta$ in green) and unclear (if $-\delta \leq \mathrm{Re}(\lambda) \leq 0$ in orange). On the left the functional considered is F_ε, while on the right Φ_ε.

we have set $\delta = 0$ and thus we do not consider the functional Φ_ε, but just F_ε. The rank-adaptive integrator provides similar results to those shown in Table 6.4, even though the distance is a bit larger due to a different integration trajectory followed. Two of the algorithms (Grad and FGM) proposed by Gillis and Sharma reach smaller distances in less computational time and they are the fastest among all the methods shown here. But the overall best result is provided by the method proposed in [59] by Noferini and Poloni for Hurwitz and complex stabilization, Hurw-Cpx in the tables, where the value of the distance is significantly smaller than the other competitors considered in Table 6.5 and also in terms of accuracy it is the best.

6.5.3 Gcdmat matrix

For another comparison of the computation of the unstructured distance with the competitors, we consider the Gcdmat matrix $M \in \mathbb{R}^{n\times n}$ from the `gallery` function of MATLAB, whose entries are defined as

$$M_{i,j} = \mathrm{GCD}(i,j), \qquad i,j = 1,\dots,n$$

The matrix M is symmetric positive definite and in our example we fix $n = 40$ and consider

$$A = -M + \frac{1}{2}I, \tag{6.11}$$

which has 2 unstable eigenvalues. We show the results of the comparison in Table 6.6.

Also in this case the best distance $\varepsilon_\star = 0.3326$ is provided by the method of Noferini-Poloni, while the other methods compute $\varepsilon_\star = 0.3898$. But, in terms of CPU time, the approach of [59] is very slow compared to the other ones.

A possible explanation of this behaviour, due to Professor Vanni Noferini, is the following. This issue may be caused by the fact that this method computes a smaller distance that is attained by a minimizer with very large Jordan blocks, which seems to be ignored by the two-level approach. This feature could slow down the convergence if a too high accuracy is asked by the default choice of parameters of Manopt, which is the Riemannian optimization software the method in [59] relies on. In particular the algorithm may waste time in improving digits that just cannot numerically be

	Rk feature	$\varepsilon_\star$	rk($E_\star$)	Functional	Re(λ_1)	Time (s)
F_ε	Adaptive	0.3898	**2**	$1.0000 \cdot 10^{-9}$	$4.2433 \cdot 10^{-5}$	0.6165
G-S	BCD	0.3898	40	$9.1067 \cdot 10^{-31}$	$9.5429 \cdot 10^{-16}$	2.6775
	Grad	0.3898	37	$1.2439 \cdot 10^{-30}$	$1.1153 \cdot 10^{-15}$	0.0445
	FGM	0.3898	39	$9.1493 \cdot 10^{-31}$	$1.3527 \cdot 10^{-15}$	**0.0373**
N-P	Hurw-Cpx	**0.3326**	38	**0**	**0**	39.5402

Table 6.6: Comparison between the rank-adaptive approach and the algorithms by Gillis and Sharma (G-S) and that by Noferini and Poloni for the Gcdmat matrix (6.11). Best results highlighted in bold.

improved and it is plausible that a small edit in this sense can improve this issue. I agree with this interpretation.

Remark 6.5.3. *The two examples considered here suggest that for low dimensional matrices the methods proposed by Gillis and Sharma are preferable in terms of time, while the one by Noferini and Poloni computes the smallest distance; however it could be too expensive to perform them for high dimensional matrices. The algorithms in [20] and [59] do not take into account the rank properties of the problem and hence the perturbations found have higher rank, almost full. Moreover, it seems that they are not easy to extend to the structured version of the problem, while this can be done for the rank-adaptive approach.*

6.6 Structured distance via a low-rank adaptive ODE

In this section we briefly describe the generalization of the unstructured problem (6.1) to its structured version, as done in Chapter 3. Let now $\mathcal{S} \in \mathbb{C}^{n\times n}$ be a linear subspace, such as a prescribed sparsity pattern, Toeplitz matrices, real matrices, etc.. Given an unstable matrix $A \in \mathcal{S}$, we look for a matrix $\Delta \in \mathcal{S}$ with smallest norm that stabilizes A. Formally we want to solve the optimization problem

$$\arg\min_{\Delta\in\mathcal{S}} \left\{ \|\Delta\|_F : \sigma(A+\Delta) \subseteq \mathbb{C}^-_\delta \right\},$$

which is a generalization of (6.1). For the solution of this structured optimization problem we follow the same approach used in the previous sections, with the *structured inner* and *outer iterations*. Let $\Pi_\mathcal{S}$ be the orthogonal projection, with respect to the inner Frobenius product, to the subspace $\mathcal{S}$. Generally it is easy to compute an explicit formula for $\Pi_\mathcal{S}$ and some examples can be found in Appendix B. By proceeding as in the unstructured case, we project onto $\mathcal{S}$ the gradient G_ε (see for instance [32]) to get the new system

$$\dot{E} = -\Pi_\mathcal{S}(G_\varepsilon(E)) + \mathrm{Re}\langle \Pi_\mathcal{S}(G_\varepsilon(E)), E\rangle E. \qquad (6.12)$$

Equation (6.12) represents a gradient system where the gradient G_ε has been replaced by $\Pi_\mathcal{S} G_\varepsilon$. All the results for the unconstrained case extend to the structured system, with the replacement of the structured gradient. However the gradient is generally not low-rank, since the property of G_ε is now subject to the presence of the projection $\Pi_\mathcal{S}$ and cannot be exploited as before. But it turns out that also in this case we can formulate a low-rank adaptive ODE that takes into account also the structure constraint. Let us assume that $E = \Pi_\mathcal{S} Y$, for a certain Y that has the same rank r of

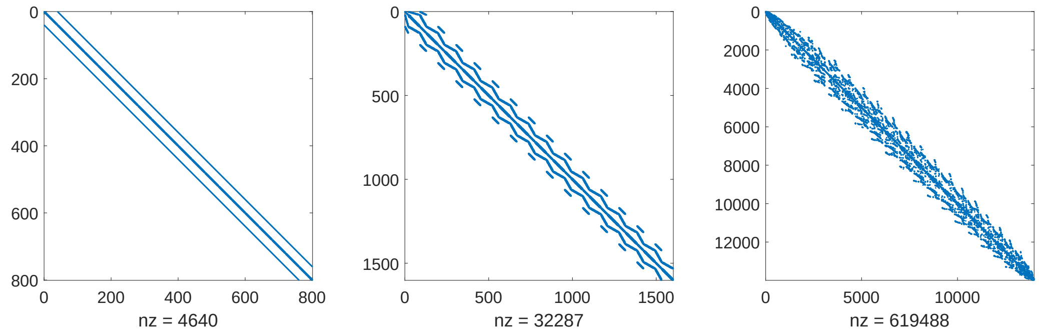


FIGURE 6.3: Structural patterns of the Brusselator (left), Fidap (centre) and BCS (right) matrices.

$G_\varepsilon(E)$. As in [34], let us consider the ODE

$$\dot{Y} = -P_Y G_\varepsilon(\Pi_{\mathcal{S}} Y) + \mathrm{Re}\langle P_Y G_\varepsilon(\Pi_{\mathcal{S}} Y), \Pi_{\mathcal{S}} Y\rangle Y, \tag{6.13}$$

where P_Y is the projection onto the tangent space $\mathcal{T}_Y\mathcal{M}_r$ at the rank r manifold $\mathcal{M}_r$. An explicit expression for P_Y is given by (see Proposition B.0.3 or [32])

$$P_Y(A) = A - (I - UU^*)A(I - VV^*)$$

where $Y = USV^*$ is an SVD decomposition of Y (here S is required to be invertible, but it may not be diagonal). If the rank of Y is fixed, this is a low-rank ODE whose stationary points are explicitly related to the ones of the gradient system and in particular this correspondence is bijective. Equation (6.13) is not a gradient system, but it can be proved, similarly as done in Section 3.1.3, that its integration locally converges to a stationary point $Y_\star$ that represents a stationary point $E_\star = \Pi_{\mathcal{S}} Y_\star$ of the original gradient system (6.12). Indeed when the trajectory is close to a stationary point, the rank of the gradient stabilizes and thus it is possible to proceed as in Theorem 3.1.10. Thus, it is possible to solve (6.13) by means of the rank-adaptive integrator, so that we can exploit the low-rank features of this problem even in the structured case.

6.7 Numerical examples for the structured case

In this section we investigate the behaviour of the algorithm for the structured optimization problem in some numerical examples, including matrices with large dimension. In all the cases we consider as structure the sparsity pattern of the matrix with the further constraint that the perturbation is real. In Figure 6.3 we show the patterns of the large examples we consider in the numerical experiments.

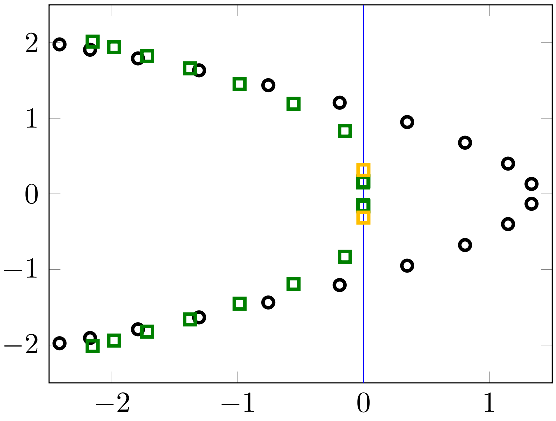


FIGURE 6.4: Pentadiagonal Toeplitz matrix: original eigenvalues (black circles), stabilized ones (if $\text{Re}(\lambda) < -\delta$ in green) and unclear (if $-\delta \leq \text{Re}(\lambda) \leq 0$ in orange).

Rk feature	$\varepsilon_\star$	$\text{rk}(E_\star)$	$F(E_\star(\varepsilon_\star))$	$\text{Re}(\lambda_1)$	Time (s)
Adaptive	**2.8573**	11	$7.0889 \cdot 10^{-10}$	$-9.7337 \cdot 10^{-4}$	4.5366
Fixed (8)	2.8888	**8**	$9.0637 \cdot 10^{-10}$	$-9.7890 \cdot 10^{-4}$	4.5569
Fixed (9)	2.8935	9	$2.6935 \cdot 10^{-11}$	$-9.9613 \cdot 10^{-4}$	4.6275
Fixed (10)	2.8848	10	$9.9999 \cdot 10^{-10}$	$-9.6838 \cdot 10^{-4}$	**3.6226**
Fixed (11)	2.8698	11	$\mathbf{7.5315 \cdot 10^{-12}}$	$\mathbf{-9.9775 \cdot 10^{-4}}$	7.7734

TABLE 6.7: Pentadiagonal Toeplitz matrix (6.14) results with functional F_ε. Best results highlighted in bold.

6.7.1 Pentadiagonal Toeplitz matrix

Let us consider the pentadiagonal Toeplitz matrix

$$P = \begin{pmatrix} -\frac{1}{2} & 1 & 1 & & & \\ 1 & -\frac{1}{2} & 1 & 1 & & \\ 1 & 1 & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 & 1 \\ & & 1 & 1 & -\frac{1}{2} & 1 \\ & & & 1 & 1 & -\frac{1}{2} \end{pmatrix} \in \mathbb{R}^{20\times 20}. \tag{6.14}$$

We apply the method where the solution of the *structured inner iteration* is obtained by the integration of (6.13) and we collect the results in Table 6.7 and in Figure 6.4.

The rank-adaptive integrator computes the smallest distance, similar to those provided by the fixed-rank methods. The results are identical also when the algorithms are applied for the functional Φ_ε.

6.7.2 Brusselator matrix

Now we consider the Brusselator matrix [1] from the NEP collection [5]. This matrix has size $n = 800$ with non-zeros $nnz = 4640 \approx 5.8n$ and it arises from a two-dimensional

[1] See https://math.nist.gov/MatrixMarket/data/NEP/brussel/rdb800l.html

	Rk feature	$\varepsilon_\star$	$\mathrm{rk}(E_\star)$	Functional	$\mathrm{Re}(\lambda_1)$	Time (s)
F_ε	Adaptive	**0.9912**	**2**	$1.0000 \cdot 10^{-9}$	$\mathbf{-9.6838 \cdot 10^{-4}}$	**8.2220**
	Fixed (2)	**0.9912**	**2**	$1.0000 \cdot 10^{-9}$	$\mathbf{-9.6838 \cdot 10^{-4}}$	16.9868
Φ_ε	Adaptive	**0.9912**	**2**	$\mathbf{9.9984 \cdot 10^{-10}}$	$\mathbf{-9.6838 \cdot 10^{-4}}$	75.9460
	Fixed (2)	**0.9912**	**2**	$1.0000 \cdot 10^{-9}$	$\mathbf{-9.6838 \cdot 10^{-4}}$	195.4750

TABLE 6.8: Brusselator matrix: features of the solution of the *structured outer iteration.* Best results highlighted in bold.

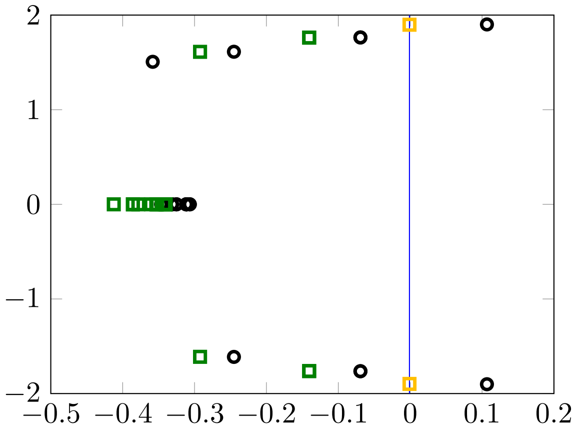


FIGURE 6.5: Zoom of Brusselator matrix eigenvalues: original eigenvalues (black circles), stabilized ones (if $\mathrm{Re}(\lambda) < -\delta$ in green) and unclear (if $-\delta \le \mathrm{Re}(\lambda) \le 0$ in orange).

reaction-diffusion model in chemical engineering. It has two conjugate unstable eigenvalues that are close to the imaginary axis and some eigenvalues with negative real part close to zero. We applied the algorithm with $\delta = 10^{-3}$ and parameters

$$\tau_{\text{inn}} = 10^{-9}, \quad \tau_{\text{out}} = 10^{-9}, \quad \tau_{\text{rk}} = 10^{-9}, \qquad \text{maxit}_{\text{inn}} = 150, \quad \text{maxit}_{\text{out}} = 200.$$

For the fixed-rank method, we were only able to select $r = 2$, since in the other cases the computations in the MATLAB function `eigs` did not converge.

The results in Table 6.8 show that for the functional F_ε the algorithm is quicker and it provides the same distances as those associated to Φ_ε. As shown in Figure 6.5, the rank-adaptive integrator captures the fact that two stable eigenvalues are close to the imaginary axis and it considers them in the gradient by moving them leftwards.

6.7.3 Fidap matrix

Now we consider the Fidap matrix [2] from the SPARSKIT collection [5], which arises in fluid dynamics modelling. This matrix has size 1601 × 1601 and it is symmetric. In our example we consider the shifted matrix $A - \frac{3}{2}I$ so that the number of unstable eigenvalues reduces to 4. In this case the parameters chosen are

$$\tau_{\text{inn}} = 10^{-9}, \quad \tau_{\text{out}} = 10^{-9}, \quad \tau_{\text{rk}} = 10^{-9}, \qquad \text{maxit}_{\text{inn}} = 150, \quad \text{maxit}_{\text{inn}} = 200.$$

[2] See https://math.nist.gov/MatrixMarket/data/SPARSKIT/fidap/fidap004.html

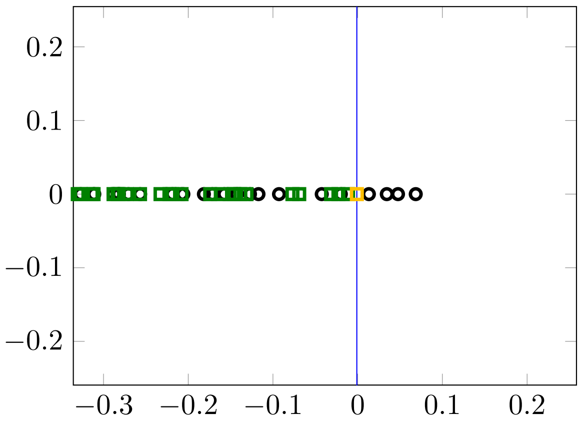


FIGURE 6.6: Zoom of Fidap matrix eigenvalues: original eigenvalues (black circles), stabilized ones (if $\mathrm{Re}(\lambda) < -\delta$ in green) and unclear (if $-\delta \le \mathrm{Re}(\lambda) \le 0$ in orange).

Table 6.9 and Figure 6.6 show the results of the algorithm applied on the shifted matrix.

	Rk feature	$\varepsilon_\star$	$\mathrm{rk}(E_\star)$	Functional	$\mathrm{Re}(\lambda_1)$	Time (s)
F_ε	Adaptive	0.1886	5	$1.0000 \cdot 10^{-9}$	$-9.5528 \cdot 10^{-4}$	**13.4614**
	Fixed (4)	0.1883	**4**	$1.0000 \cdot 10^{-9}$	$-9.5528 \cdot 10^{-4}$	23.5807
	Fixed (5)	0.1883	5	$1.0000 \cdot 10^{-9}$	$-9.5528 \cdot 10^{-4}$	25.8531
Φ_ε	Adaptive	0.1888	5	$1.0000 \cdot 10^{-9}$	$-9.5528 \cdot 10^{-4}$	85.0830
	Fixed (4)	0.1882	**4**	$\mathbf{5.9640 \cdot 10^{-10}}$	$\mathbf{-9.6546 \cdot 10^{-4}}$	433.4435
	Fixed (5)	**0.1881**	5	$6.3283 \cdot 10^{-10}$	$-9.6442 \cdot 10^{-4}$	500.0907

TABLE 6.9: Fidap matrix: results with functional F_ε. Best results highlighted in bold.

Also in this case we observe that the distance provided by the functional Φ_ε is slightly lower than the one associated to the functional F_ε, but this improvement is not significant enough to justify the higher computational time needed by Φ_ε with respect to the functional F_ε.

6.7.4 BCS matrix

Finally, in order to show that the low-rank-adaptive-integrator can be applied successfully to high dimensional examples, we consider the BCS matrix[3] from the SPARSKIT collection [5], which represents a stiffness matrix. This matrix has size $n = 13992$ with $nnz = 619488 \approx 44n$ non-zero entries.

We applied our algorithm on the shifted matrix $A - \frac{103}{2}I$ so that the number of unstable eigenvalues reduces to 2 and it is possible to use effectively the adaptive-integrator. With the same choice of the parameters, we show the results of the rank-adaptive method in Table 6.10 and Figure 6.7.

The algorithm manages to achieve the sought accuracy in 555 seconds, for a computed distance given by $\varepsilon_\star = 2.3640$.

[3] https://math.nist.gov/MatrixMarket/data/Harwell-Boeing/bcsstruc5/bcsstk29.html

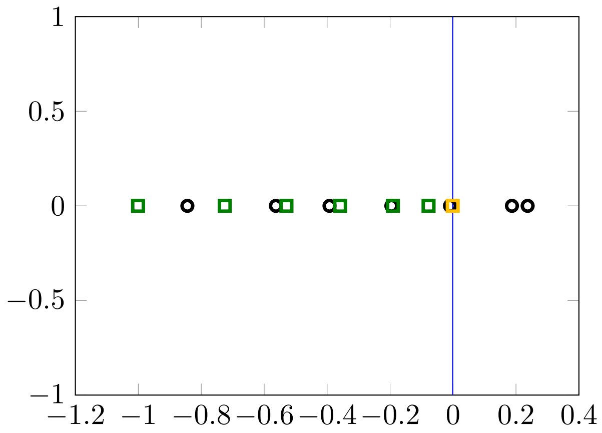


FIGURE 6.7: Zoom of BCS matrix eigenvalues: original eigenvalues (black circles), stabilized ones (if $\mathrm{Re}(\lambda) < -\delta$ in green) and unclear (if $-\delta \leq \mathrm{Re}(\lambda) \leq 0$ in orange).

	$\varepsilon_\star$	$\mathrm{rk}(E_\star)$	Functional	$\mathrm{Re}(\lambda_1)$	Time (s)
Adaptive	2.3640	3	$1.0000 \cdot 10^{-9}$	$-9.5528 \cdot 10^{-4}$	555.0496

TABLE 6.10: BCS matrix: features of the solution of the *structured outer iteration* for the functional F_ε.

Code and Data Availability

The codes implementing the algorithms discussed in this chapter are publicly available at:

https://github.com/StefanoSicilia/MatrixStabilization

The figures have been created with the usage of the software `matlab2tikz`, that can be found at

https://github.com/matlab2tikz/matlab2tikz

Chapter 7

Conclusion and perspectives

In this thesis we have presented a versatile two-level approach that can be used in a wide class of unstructured and structured matrix nearness problems and we have applied it to different settings of both *violating* and *recovering* problems. We have shown that, in the cases considered, the optimization problems possess low-rank properties and we have shown how to exploit them. In the following we briefly recap the main results of each chapter.

- Chapters 1 and 2 have introduced the general problem and the two-level approach used for its solution. In particular in Chapter 2 we have discussed in detail the method for the unstructured matrix nearness problem and this has represented the starting point of the research for the extensions of the results to the structured version.

- Chapter 3 has presented the main theoretical results of this PhD thesis that are common for all the applications studied. The two-level method has been adapted to the structured case and then the low-rank underlying property has been revealed in order to take advantage of its associated computational benefits.

- In Chapter 4 we have focused on three *violating* problems arising in matrix theory: the computation of the distance to Hurwitz-instability, the approximation of the distance to Schur-instability and the computation of the distance to singularity. We have shown how to solve these problems, both the unstructured and structured versions, by means of the two-level approach and we have exploited their intrinsic rank-1 nature by integrating a rank-1 matrix ODE whose stationary points corresponds, up to an eventual projection onto the structure, to the sought optimizers. We have implemented and tested the resulting algorithm by means of a splitting method that integrates the rank-1 ODE and we have shown the results on several numerical examples. The contents of this chapter have been published as a scientific article (see [34]).

- In Chapter 5 we have considered an application of a *violating* matrix nearness problem in a graph setting. Given an undirected weighted graph to be partitioned, we have presented a method for computing the best number k that needs to be given as an input to the spectral clustering algorithm, so that the partitioning provided is the most robust as possible. We have done this by approximating a structured distance between the weight matrix of the graph and another weight matrix whose associated Laplacian has vanishing k-th spectral gap. This approach provides a more reliable measurement of the robustness of the clustering with respect to the classical criterium of the spectral gaps, which instead relies on an unstructured distance associated to a less appropriate measure. We have applied the two-level approach to this setting and we have shown how to compute the sought optimizers of the problem by means of the integration of a symmetric rank-4 ODE. The resulting algorithm is a generalization to the low-rank case of that developed in Chapter 4. This chapter has been published as a scientific article (see [37]) and the codes associated are available on Github at the webpage https://github.com/StefanoSicilia/Spectral-Clustering-stability .

- Chapter 6 has focused on the *recovering* problem of the unstructured and structured stabilization of a matrix. The two-level approach has been adapted also in this case and the associated (*structured*) *inner iteration* deals with an objective functional with a variable number of addends, which corresponds to the unstable eigenvalues, and consequently a variable rank of the associated gradient. The low-rank properties of the problem have been exploited also in this setting by means of the rank-adaptive integrator presented in [13] which turns out to be very suitable, since it preserves many of the theoretical properties of the ODE, while also allowing to follow precisely the current rank of the solution. We have shown the results of the corresponding algorithm on several numerical examples, including some of large dimension. The contents of this chapter have been published as a scientific paper (see [38]) and the associated codes are available on Github at the webpage https://github.com/StefanoSicilia/MatrixStabilization .

While there exist other approaches to face unstructured matrix nearness problems, for the structured version, at the best of our knowledge, there are fewer methods. Hence the contribution of this PhD thesis aims to fill this gap in the literature by providing a technique that solves the structured problem and that also favourably exploits its low-rank underlying properties. Moreover the two-level approach is very versatile and, as shown, it can be adapted to many topics of mathematics. In particular the applications studied here concern robustness problems, stabilization problems and robustness for a clustering method for an undirected weighted graph. However there are many potential future directions where to extend and apply the method developed and we illustrate some possible perspectives.

First of all, the parameters in the algorithms have been generally tuned heuristically and hence it could be interesting to investigate more how to select them. In particular, the choice of the upper and lower bounds for the *outer iteration* could be improved by means of some theoretical estimates on the distance to be found and also the tolerances for both the *inner* and *outer iterations* can be further studied. This is also an important aspect for practical implementations in the applications considered.

A research direction concerns the stability of neural ODEs (see e.g. [15]). A neural ODE is an ordinary differential equation whose vector field is a neural network, that is

$$\dot{x}(t) = \sigma(Ax(t) + b), \quad t \in [0, T],$$

where $x(t) \in \mathbb{R}^n$ is the feature vector evolution function, $\sigma : \mathbb{R} \to \mathbb{R}$ is a smooth activation function applied entry-wise, $T > 0$ is the time horizon, $A \in \mathbb{R}^{n\times n}$ is the weight matrix and $b \in \mathbb{R}^n$ is the bias vector. Usually neural ODEs are naturally prone to adversarial attacks, i.e. perturbations in input designed to make a neural network return a wrong output. The two-level approach can be applied also in this case to control the stability of the weight matrix so that a new training of the parameters provides more robust outputs from the neural network. A work on this topic is currently in progress and it concerns an extension of [25].

Recently the interest in structured matrix nearness problems has increased and some new methods have been developed. For instance the method proposed in [21] is apparently quite different from the two-level approach discussed in this thesis, but somehow it resembles a sort of dual of it. It divides the main task into sub-problems and it takes advantage of the fact that some of them have an explicit solution that do not require much effort to be computed. This sort of dual connection between the methods may be exploited for a future research direction that aims to combine the two approaches in order to get the best properties from each of them. This perspective is currently under discussion.

The versatility of the two-level approach and the chance of exploiting the low-rank properties makes the application of this method very appealing also for large dimensional problems. Indeed, thanks to the low memory requirements of the developed algorithm, it is possible to tackle more practical problems which are usually associated to large matrices. This feature enables to explore many future perspectives also from this point of view.

Appendix A

Fixed rank manifold and its tangent space

In this chapter of the appendix we include some classical results in differential geometry, see e.g. [39] for further details. We show that the set of all matrices with fixed-rank is a manifold and we characterize its tangent space. We consider a field $\mathbb{K}$ of characteristic 0, such as $\mathbb{K} = \mathbb{C}$ or $\mathbb{K} = \mathbb{R}$, and we characterize the set of rank-r matrices $\mathcal{M}_r \subseteq \mathbb{K}^{m\times n}$.

Proposition A.0.1. *Given a field $\mathbb{K}$ of characteristic 0, the subset*

$$\mathcal{M}_r = \{M \in \mathbb{K}^{m\times n} \ : \ \operatorname{rank}(M) = r\}$$

is a submanifold of $\mathbb{K}^{m\times n}$ of $\mathbb{K}$-dimension $mn - (m-r)(n-r)$.

Proof. Let us define the subset of $\mathbb{K}^{m\times n}$

$$\mathcal{Z} = \left\{ \begin{pmatrix} A & B \\ C & D \end{pmatrix} : A \in \mathbb{K}^{r\times r} \text{ invertible} \right\},$$

where $B \in \mathbb{K}^{r\times(n-r)}, C \in \mathbb{K}^{(m-r)\times r}, D \in \mathbb{K}^{(m-r)\times(n-r)}$ are arbitrary matrices and consider $\mathcal{N} = \mathcal{Z} \cap \mathcal{M}_r$. Up to a permutation of rows and columns, that is a linear isomorphism mapping φ, a matrix $M \in \mathcal{M}_r$ can always be written as a matrix in $\mathcal{N}$. If we show that $\mathcal{N}$ is a manifold, then also $\mathcal{M}_r$ is, since it is possible to define charts as a composition of φ and the inherited charts of $\mathcal{N}$.

In order to show that $\mathcal{N}$ is a manifold, we make use of the preimage theorem. Post-multiplying by an invertible matrix does not change the rank of a matrix and hence

$$\operatorname{rank}\left(\begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} I_r & -A^{-1}B \\ 0 & I_{(n-r)} \end{pmatrix} \right) = \operatorname{rank}\left(\begin{pmatrix} A & 0 \\ C & -CA^{-1}B + D \end{pmatrix} \right) = r,$$

which implies that

$$M := \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \mathcal{N} \iff f(M) := D - CA^{-1}B = 0.$$

To conclude we just need to show that $0 \in \mathbb{K}^{(m-r)\times(n-r)}$ is a regular value of the smooth function $f : \mathcal{N} \to \mathbb{K}^{(m-r)\times(n-r)}$. For any $M \in \mathcal{N}$ and any $X \in \mathbb{K}^{(m-r)\times(n-r)}$, let us consider the curve parametrized in $t \geq 0$

$$\gamma(t) = \begin{pmatrix} A & B \\ C & D + tX \end{pmatrix} \subseteq \mathcal{Z}.$$

We have

$$\left.\frac{\mathrm{d}f(\gamma(t))}{\mathrm{d}t}\right|_{t=0} = \left.\frac{\mathrm{d}}{\mathrm{d}t}\big(D + tX - CA^{-1}B\big)\right|_{t=0} = X = \mathrm{d}f_M\left(\begin{pmatrix} 0 & 0 \\ 0 & X \end{pmatrix}\right),$$

which shows that $\mathrm{d}f_M$ is surjective for any M and hence 0 is a regular value for f. □

The following result provides two equivalent characterizations of the tangent space at point in $\mathcal{M}_r$.

Proposition A.0.2. *Given $M \in \mathcal{M}_r \subseteq \mathbb{K}^{m\times n}$, consider an SVD-like decomposition*

$$M = USV^*,$$

where $U \in \mathbb{K}^{n\times r}$, $V \in \mathbb{K}^{m\times r}$ and $S \in \mathbb{K}^{r\times r}$ is invertible and $U^*U = V^*V = I_r$. Then the tangent space at M is

$$\mathcal{T}_M\mathcal{M}_r = \big\{WSV^* + UXV^* + USZ^* : X \in \mathbb{K}^{r\times r},\ W^*U + U^*W = Z^*V + V^*Z = 0\big\}.$$

which coincides with

$$\mathcal{T}_M\mathcal{M}_r = \big\{WSV^* + UXV^* + USZ^* \ :\ X \in \mathbb{K}^{r\times r},\ W^*U = Z^*V = 0\big\}.$$

Proof. Let us define the curve $\gamma : \mathbb{R} \to \mathcal{M}_r$

$$\gamma(t) = (U + tW + \mathcal{O}(t^2))(S + tX + \mathcal{O}(t^2))(V + tZ + \mathcal{O}(t^2))^*.$$

By construction $\gamma(t) \in \mathcal{M}_r$ if we impose that $(U + tW + \mathcal{O}(t^2))$ and $(V + tZ + \mathcal{O}(t^2))$ have orthonormal columns. The conditions of the first constraint becomes

$$I_r = (U + tW + \mathcal{O}(t^2))^*(U + tW + \mathcal{O}(t^2)) = I_r + t(U^*W + W^*Z) + \mathcal{O}(t^2)$$

and evaluating the derivative at $t = 0$ yields $U^*W + W^*Z = 0$. Similarly $Z^*V + V^*Z = 0$ for the second equation. Thus

$$\left.\frac{\mathrm{d}\gamma}{\mathrm{d}t}\right|_{t=0} = WSV^* + UXV^* + USZ^*$$

and thus the set of tangent vectors is

$$\mathcal{T}_M\mathcal{M}_r = \big\{WSV^* + UXV^* + USZ^* : X \in \mathbb{K}^{r\times r},\ W^*U + U^*W = Z^*V + V^*Z = 0\big\}.$$

Now we show the equivalence between the two characterizations of $\mathcal{T}_M\mathcal{M}_r$. One implication is immediate, while for the other one let us consider

$$\delta M = WSV^* + UXV^* + USZ^*$$

such that $W^*U + U^*W = Z^*V + V^*Z = 0$; then

$$\delta M = (W + UW^*U)SV^* + U(S - W^*US - SV^*Z)V^* + US(Z + VZ^*V)^*$$

and it is also verified that $(W + UW^*U)^*U = (Z + VZ^*V)^*V = 0$. □

Appendix B

Projections

In this chapter of the appendix we consider some orthogonal projections with respect to the Frobenius inner product. We show the equivalence of the definitions of the projection and we provide the explicit formulas in some examples.

Definition B.0.1. *Let $\mathcal{V}$ be an Hilbert space endowed with an inner product $\langle \cdot, \cdot \rangle$ and consider a subspace $\mathcal{S} \subset \mathcal{V}$. The orthogonal projection onto $\mathcal{S}$ is the unique linear function $\Pi_{\mathcal{S}} : \mathcal{V} \to \mathcal{S}$ such that*

1. $\Pi_{\mathcal{S}}(v) = \Pi_{\mathcal{S}}(\Pi_{\mathcal{S}}(v))$ *for all $v \in \mathcal{V}$.*
2. $\langle \Pi_{\mathcal{S}}(v), w \rangle = \langle v, w \rangle$*, for all $v \in \mathcal{V}$ and for all $w \in \mathcal{S}$*

The following result gives an equivalent characterization of the orthogonal projection.

Proposition B.0.2. *Let $\mathcal{V}$ be an Hilbert space endowed with an inner product $\langle \cdot, \cdot \rangle$ and consider a subspace $\mathcal{S} \subset \mathcal{V}$ and a linear function $f : \mathcal{V} \to \mathcal{S}$. The following facts are equivalent:*

1. *for all $x \in \mathcal{V}$, we have*
$$f(x) = \arg\min_{y \in \mathcal{S}} \|x - y\|,$$
2. *the function f is the orthogonal projection onto $\mathcal{S}$.*

Proof. 1. ⇒ 2.: Since the fact that $f(x) \in \mathcal{S}$ for all $x \in \mathcal{V}$ is straightforward, the aim is to prove that
$$\operatorname{Re}\langle x - f(x), y \rangle = 0, \qquad \forall y \in \mathcal{S}.$$
For all $x \in \mathcal{V}$, $\forall y \in \mathcal{S}$ and $\forall t \in \mathbb{R}$, $f(x) - ty \in \mathcal{S}$ and the hypothesis implies
$$\|x - f(x)\|^2 \leq \|x - f(x) + ty\|^2 = \|x - f(x)\|^2 - 2t \operatorname{Re}\langle x - f(x), y \rangle + t^2 \|y\|^2,$$
that is
$$p(t) := t^2 \|y\|^2 - 2t \operatorname{Re}\langle x - f(x), y \rangle \geq 0, \qquad \forall t \in \mathbb{R}.$$
Thus the quadratic polynomial $p(t)$ must have a non-positive discriminant (otherwise we would have $p(t) < 0$), that is
$$\operatorname{Re}\langle x - f(x), y \rangle^2 \leq 0, \qquad \forall y \in \mathcal{S}$$
which implies the claim.

2. ⇒ 1.: For all $x \in \mathcal{V}$ and for all $y \in \mathcal{S}$, it holds that
$$\|x - f(x)\|^2 = \|x - y + y - f(x)\|^2 = \|x - y\|^2 + \|y - f(x)\|^2 - 2 \operatorname{Re}\langle x - y, f(x) - y \rangle$$

and since $f(x) - y \in \mathcal{S}$, the hypothesis yields

$$\mathrm{Re}\langle x-y, f(x)-y\rangle = \mathrm{Re}\langle f(x-y), f(x)-y\rangle = \mathrm{Re}\langle f(x)-f(y), f(x)-y\rangle = \|f(x)-y\|^2.$$

Thus

$$\|x - f(x)\|^2 = \|x - y\|^2 - \|y - f(x)\|^2 \leq \|x - y\|^2, \qquad \forall y \in \mathcal{S},$$

which is implies the claim. □

The following results concerns some examples of projections related to the topics of the thesis.

Proposition B.0.3. *Given a field* $\mathbb{K}$ *of characteristic* 0, *let us consider* $M \in \mathcal{M}_r \subseteq \mathbb{K}^{m\times n}$ *with decomposition*

$$M = USV^*,$$

where $U \in \mathbb{K}^{n\times r}$, $V \in \mathbb{K}^{m\times r}$ *and* $S \in \mathbb{K}^{r\times r}$ *is invertible and* $U^*U = V^*V = I_r$. *The orthogonal projection* $P_M : \mathbb{K}^{m\times n} \to \mathcal{T}_M\mathcal{M}_r$ *is given by*

$$P_M A = A - (I - UU^*)A(I - VV^*) = UU^*A + AVV^* - UU^*AVV^*.$$

Proof. As a first step we notice that P_M is linear and $P_M(A) \in \mathcal{T}_M\mathcal{M}_r$ since

$$P_M A = \left((I - UU^*)AVS^{-1}\right) SV^* + U(U^*AV)V^* + US\left((I - VV^*)A^*US^{-*}\right)^*,$$

which fulfils the characterization of Proposition A.0.2. Finally, for all $\delta M = WSV^* + UXV^* + USZ^* \in \mathcal{T}_M\mathcal{M}_r$, P_M fulfils the definition of orthogonal projection since we have

$$\mathrm{Re}\langle P_M A, \delta M\rangle = \mathrm{Re}\langle A-(I-UU^*)A(I-VV^*), WSV^*+UXV^*+USZ^*\rangle = \mathrm{Re}\langle A, \delta M\rangle,$$

where we have repeatedly used relations of the type

$$\mathrm{Re}\langle (I - UU^*)A(I - VV^*), WSV^*\rangle = \mathrm{tr}\left(WSV^*(I - VV^*)A^*(I - UU^*)\right) = 0.$$

□

Proposition B.0.4. *Given a matrix* $A = (a_{i,j}) \in \mathbb{C}^{n\times n}$, *we consider the sets*

$$\mathcal{S} = \{M = (m_{i,j}) \in \mathbb{C}^{n\times n} : m_{i,j} = 0 \quad \forall (i,j) \text{ such that } a_{i,j} = 0\},$$

$$\mathcal{R} = \{M = (m_{i,j}) \in \mathbb{R}^{n\times n} : m_{i,j} = 0 \quad \forall (i,j) \text{ such that } a_{i,j} = 0\},$$

which consist of all the matrices with the same pattern as A *and of its real version. Then*

$$(\Pi_{\mathcal{S}}(M))_{i,j} = \begin{cases} m_{i,j} & \text{if } a_{i,j} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

and

$$(\Pi_{\mathcal{R}}(M))_{i,j} = \begin{cases} \mathrm{Re}(m_{i,j}) & \text{if } a_{i,j} \neq 0 \\ 0 & \text{otherwise} \end{cases}.$$

Proof. Thanks to the result of Proposition B.0.2, the claims follow recalling that

$$\Pi_{\mathcal{S}}(M) = \arg\min_{N\in\mathcal{S}} \|M - N\|_F^2, \qquad \Pi_{\mathcal{R}}(M) = \arg\min_{N\in\mathcal{R}} \|M - N\|_F^2.$$

□

Proposition B.0.5. *Let $B \in \mathbb{C}^{n\times k}$ and $C \in \mathbb{C}^{l\times n}$ be two given full-rank matrices, with $k, l \le n$. Define the set of prescribed range and co-range as*

$$\mathcal{S} = \{B\Delta C : \Delta \in \mathbb{C}^{k\times l}\}.$$

Then $\Pi_{\mathcal{S}}(Z) = BB^\dagger Z C^\dagger C$ for all $Z \in \mathbb{C}^{n\times n}$, where $B^\dagger = (B^*B)^{-1}B^*$ and $C^\dagger = C^*(CC^*)^{-1}$ denote the Moore-Penrose pseudoinverse.

Proof. By definition $\Pi_{\mathcal{S}}(Z) \in \mathcal{S}$ for all $Z \in \mathbb{C}^{n\times n}$. Then the properties of the trace operator show that, for all $\Delta \in \mathbb{C}^{k\times l}$,

$$\mathrm{Re}\langle B\Delta C, \Pi_{\mathcal{S}}(Z)\rangle = \mathrm{Re}\left(\mathrm{tr}(C^*\Delta^* B^* B(B^*B)^{-1}B^* Z C^*(CC^*)^{-1}C)\right) = \mathrm{Re}\,\mathrm{tr}(C^*\Delta^* B^* Z),$$

which implies the claim. □

Proposition B.0.6. *Let us consider the set of Toeplitz matrices*

$$\mathcal{T} = \left\{A \in \mathbb{C}^{n\times n} : a_{i,j} = a_{i+1,j+1} \quad \forall i,j = 1,\dots,n-1\right\}.$$

For any $Z \in \mathbb{C}^{n\times n}$, we define the arithmetic means along the diagonal

$$\hat{z}_k = \frac{1}{n-|k|}\sum_{|i-j|=k} z_{i,j}, \qquad k = -n+1,\dots,n-1.$$

Then

$$\Pi_{\mathcal{T}}(Z) = \begin{pmatrix} \hat{z}_0 & \hat{z}_1 & \dots & \hat{z}_{n-1} \\ \hat{z}_{-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \hat{z}_1 \\ \hat{z}_{-n+1} & \dots & \hat{z}_{-1} & \hat{z}_0 \end{pmatrix}. \tag{B.1}$$

Proof. We rewrite the function we wish to minimize

$$\varphi(W) = \|Z - W\|_F^2 = \sum_{i,j=1}^{n} |z_{ij} - w_{ij}|^2, \qquad \forall W \in \mathcal{T},$$

as the sum of the contribution of each diagonal:

$$\varphi(W) = \sum_{k=1-n}^{n-1}\sum_{|i-j|=k} |z_{i,j} - w_k|^2 = \psi(w_{1-n},\dots,w_{n-1}),$$

where w_j are the elements of the Toeplitz matrix W located as in the shape of (B.1), that is

$$W = \begin{pmatrix} w_0 & w_1 & \dots & w_{n-1} \\ w_{-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & w_1 \\ w_{-n+1} & \dots & w_{-1} & w_0 \end{pmatrix}.$$

Now we see ψ as a real function of real variables: for all the possible indices, let $z_{i,j} = a_{i,j} + \mathrm{i}b_{i,j}$ and let $w_k = x_k + \mathrm{i}y_k$. Then

$$\theta(x_{1-n},\dots,x_{n-1},y_{1-n},\dots,y_{n-1}) = \sum_{k=1-n}^{n-1}\sum_{|i-j|=k} |a_{i,j} + \mathrm{i}b_{i,j} - x_k - \mathrm{i}y_k|^2 =$$

$$= \sum_{k=1-n}^{n-1} \sum_{|i-j|=k} (a_{i,j} - x_k)^2 + (b_{i,j} - y_k)^2 = \psi(w_{1-n}, \dots, w_{n-1}).$$

Since θ is a convex function (sum of quadratic functions), the points where the gradient vanishes are its minimizers. Thus, for all $h \in \{1-n, \dots, n-1\}$, we annihilate the partial derivatives

$$0 = \frac{\partial \theta}{\partial x_h} = -2 \sum_{|i-j|=h} (a_{i,j} - x_h),$$

that is

$$\sum_{|i-j|=h} a_{i,j} = (n - |h|) x_h,$$

since in the h^{th} diagonal there are $n - |h|$ elements. Similarly

$$y_h = \frac{1}{n-|h|} \sum_{|i-j|=h} b_{i,j}, \qquad h = -n+1, \dots, n-1,$$

which implies the claim

$$w_k = \frac{1}{n-|k|} \sum_{|i-j|=k} z_{i,j} = \hat{z}_k, \qquad k = -n+1, \dots, n-1.$$

□

Proposition B.0.7. *Consider $n = 2d$ and define the set of Hamiltonian matrices*

$$\mathcal{H} = \left\{ A \in \mathbb{R}^{2d \times 2d} : \mathrm{sym}(JA) = JA \right\}, \qquad J = \begin{pmatrix} 0 & I_d \\ -I_d & 0 \end{pmatrix},$$

where $\mathrm{sym}(A) = (A + A^T)/2$ *denotes the symmetric part of a matrix. Then, for all $Z \in \mathbb{R}^{2d \times 2d}$,*

$$\Pi_{\mathcal{H}}(Z) = J^{-1} \mathrm{sym}(J \operatorname{Re} Z).$$

Proof. In order to prove the fact we note that $J^T = -J = J^{-1}$ and we exploit theorem B.0.2. It is straightforward that $J^{-1}\mathrm{sym}(J \operatorname{Re} Z) \in \mathcal{H}$ and for all $W \in \mathcal{H}$ it holds that

$$\operatorname{Re}\langle \Pi_{\mathcal{H}}(Z), W \rangle = \langle \Pi_{\mathcal{H}}(Z), W \rangle = \ \mathrm{tr}\left(\mathrm{sym}(J \operatorname{Re} Z) J^{-T} W \right) = \langle \mathrm{sym}(J \operatorname{Re} Z), JW \rangle$$

and

$$\operatorname{Re}\langle Z, W \rangle = \langle \operatorname{Re}(Z), W \rangle = \langle J \operatorname{Re}(Z), JW \rangle.$$

By means of the decomposition $A = \mathrm{sym}(A) + \mathrm{skew}(A)$, the symmetry of JW, it follows

$$\langle J \operatorname{Re}(Z), JW \rangle = \langle \mathrm{sym}(J \operatorname{Re} Z), JW \rangle + \langle \mathrm{skew}(J \operatorname{Re} Z), JW \rangle =$$

$$= \langle \mathrm{sym}(J \operatorname{Re} Z), JW \rangle + \frac{1}{2} \mathrm{tr}\left(\operatorname{Re} Z^T J^T JW \right) - \frac{1}{2} \mathrm{tr}\left(J \operatorname{Re} Z JW \right) =$$

$$= \langle \mathrm{sym}(J \operatorname{Re} Z), JW \rangle + \frac{1}{2} \mathrm{tr}\left(\operatorname{Re} Z^T W \right) - \frac{1}{2} \mathrm{tr}\left((J \operatorname{Re} Z)^T JW \right) = \langle \mathrm{sym}(J \operatorname{Re} Z), JW \rangle,$$

where $\mathrm{skew}(A) = (A - A^T)/2$ denotes the skew-symmetric part of a matrix. □

Proposition B.0.8. *For $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$, let us define the perturbation space*

$$\mathcal{P}(B, C) = \left\{ B\Delta C : \Delta \in \mathbb{K}^{k\times l} \right\} \subseteq \mathbb{K}^{n\times n},$$

where $B \in \mathbb{K}^{n\times k}$ and $C \in \mathbb{K}^{l\times n}$ are given matrices of full rank that prescribe the range and co-range of the matrices in $\mathcal{P}(B, C)$. Then, for all $Z \in \mathbb{K}^{n\times n}$,

$$\Pi_{\mathcal{P}(B,C)}(Z) = BB^{\dagger}ZC^{\dagger}C,$$

where $B^{\dagger}$ and $C^{\dagger}$ are the Moore-Penrose inverses of B and C respectively.

Proof. Since B and C have full rank, it is well known that

$$B^{\dagger} = (B^*B)^{-1}B^*, \qquad C^{\dagger} = C^*(CC^*)^{-1}.$$

We exploit again theorem B.0.2. It is straightforward that

$$\Pi_{\mathcal{P}(B,C)}(Z) \in \mathcal{P}(B, C)$$

and, for all $\Delta \in \mathbb{K}^{k\times l}$ and for all $Z \in \mathbb{K}^{n\times n}$, we have

$$\mathrm{Re}\langle B\Delta C, BB^{\dagger}ZC^{\dagger}C\rangle = \mathrm{Re}\left(\mathrm{tr}\left(C^*\Delta^*B^*B(B^*B)^{-1}B^*ZC^*(CC^*)^{-1}C\right)\right) =$$

$$= \mathrm{Re}\left(\mathrm{tr}\left(\Delta^*B^*ZC^*(CC^*)^{-1}CC^*\right)\right) = \mathrm{Re}\left(\mathrm{tr}\left(C^*\Delta^*B^*Z\right)\right) = \mathrm{Re}\langle B\Delta C, Z\rangle,$$

where we have used the properties of the trace operator. □

Proposition B.0.9. *Let $\mathcal{S}$ be a subspace of $\mathbb{C}^{n\times n}$ (whose orthogonal projection is $\Pi_{\mathcal{S}}$) and define*

$$\mathcal{S}_1 = \{A \in \mathcal{S} : \|A\|_F = 1\}.$$

For any $E \in \mathcal{S}_1$, the orthogonal projection $\widehat{\Pi}^{\mathcal{S}}_E$ onto the tangent space $\mathcal{T}_E\mathcal{S}_1$ is

$$\widehat{\Pi}^{\mathcal{S}}_E Z = \Pi_{\mathcal{S}}Z - \mathrm{Re}\langle \Pi_{\mathcal{S}}Z, E\rangle E, \qquad \forall Z \in \mathbb{C}^{n\times n}.$$

Proof. We need to show that, for all $Z \in \mathbb{C}^{n\times n}$, $\widehat{\Pi}^{\mathcal{S}}_E Z \in \mathcal{T}_E\mathcal{S}_1$ and

$$\mathrm{Re}\langle \widehat{\Pi}^{\mathcal{S}}_E Z, W\rangle = \mathrm{Re}\langle Z, W\rangle, \qquad \forall W \in \mathcal{T}_E\mathcal{S}_1.$$

For all $0 < \delta << 1$, the matrix $E + \delta\widehat{\Pi}^{\mathcal{S}}_E Z$ is in the subspace $\mathcal{S}$ and

$$\|E + \delta\widehat{\Pi}^{\mathcal{S}}_E Z\|_F^2 = 1 + 2\delta\,\mathrm{Re}\langle E, \Pi_{\mathcal{S}}Z - \mathrm{Re}\langle \Pi_{\mathcal{S}}Z, E\rangle E\rangle + \mathcal{O}(\delta^2) = 1 + \mathcal{O}(\delta^2),$$

that is $E + \delta\widehat{\Pi}^{\mathcal{S}}_E Z \in \mathcal{S}_1$ up to quadratic terms in δ and thus $\widehat{\Pi}^{\mathcal{S}}_E Z \in \mathcal{T}_E\mathcal{S}_1$. Moreover

$$\mathrm{Re}\langle \widehat{\Pi}^{\mathcal{S}}_E Z, W\rangle = \mathrm{Re}\langle \Pi_{\mathcal{S}}Z, W\rangle - \mathrm{Re}\langle \Pi_{\mathcal{S}}Z, E\rangle \cdot \mathrm{Re}\langle E, W\rangle =$$

$$= \mathrm{Re}\langle Z, \Pi_{\mathcal{S}}W\rangle - \mathrm{Re}\langle Z, E\rangle \cdot \mathrm{Re}\langle E, \Pi_{\mathcal{S}}W\rangle =$$

$$= \mathrm{Re}\langle Z, \Pi_{\mathcal{S}}W - \mathrm{Re}\langle E, \Pi_{\mathcal{S}}W\rangle E\rangle = \mathrm{Re}\langle Z, \widehat{\Pi}^{\mathcal{S}}_E W\rangle$$

and the claim is straightforward since $W = \widehat{\Pi}^{\mathcal{S}}_E W$ by hypothesis. □

Proposition B.0.10. *With the same notation of Proposition B.0.9, let $A, B \in \mathcal{S}_1$ and assume that*

$$\delta = \|A - B\|_F << 1.$$

Then

$$\widehat{\Pi}_B(A-B) = A - B + R,$$

where $\|R\|_F = \mathcal{O}(\delta^2)$.

Proof. It holds that

$$A = B + (A - B)$$

and taking the norms yields

$$1 = 1 + 2\,\mathrm{Re}\langle B, A-B\rangle + \|A-B\|_F^2$$

and hence $\mathrm{Re}\langle B, A-B\rangle = \mathcal{O}(\delta^2)$. Thus

$$\widehat{\Pi}_B(A-B) = \Pi_{\mathcal{S}}(A-B) + \mathrm{Re}\langle \Pi_{\mathcal{S}}(A-B), B\rangle B = A - B + R,$$

where

$$\|R\|_F = |2\,\mathrm{Re}\langle B, A-B\rangle| \cdot \|B\|_F = \mathcal{O}(\delta^2).$$

□

Appendix C

Some non-generic examples

In this chapter of the appendix we collect two examples that provide uncommon events in the framework of the matrix nearness problems described in the thesis.

Example C.0.1. Let $n = 2$ and consider the matrix

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \in \mathcal{S} = \left\{ \begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix} \ : \ a \in \mathbb{C} \right\} \subseteq \mathbb{C}^{2\times 2}$$

which satisfies the property $\mathscr{P} = \{$ the matrix is singular $\}$. It is clear that for any matrix $\Delta \in \mathcal{S}$, then also $A + \Delta$ fulfils $\mathscr{P}$ and hence it is not possible to solve problems (1.4) and (3.1).

Example C.0.2. This example provides a vector z defined as in (5.10) that is a linear combination of x and y, leading to a case where the matrix R_ε defined in Section 5.4.1 has not rank 4. The counterexample was generated starting from the vectors $\mathbb{1}, u, v \in \mathbb{R}^7$

$$\mathbb{1} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad u = \begin{pmatrix} 6 \\ 7 \\ 10 \\ 7 \\ 7 \\ 10 \\ 10 \end{pmatrix}, \qquad v = \begin{pmatrix} 9 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

which are linearly independent. Performing Gram-Schmidt algorithm yields the orthogonal vectors with unit norm

$$\tilde{\mathbb{1}} = \frac{1}{\sqrt{7}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad x = \frac{1}{\sqrt{924}} \begin{pmatrix} -15 \\ -8 \\ 13 \\ -8 \\ -8 \\ 13 \\ 13 \end{pmatrix}, \qquad y = \frac{1}{\sqrt{132}} \begin{pmatrix} 9 \\ -4 \\ 1 \\ -4 \\ -4 \\ 1 \\ 1 \end{pmatrix}.$$

Now we show that the vectors $\tilde{\mathbb{1}}, x$ and y can be seen as the eigenvectors of the Laplacian of a graph. Let us complete the three vectors to an orthonormal basis

$\mathcal{B} = \{\tilde{\mathbb{1}}, x, y, d, e, f, g\}$ of $\mathbb{R}^7$, where

$$d = \frac{1}{\sqrt{6}}\begin{pmatrix}0\\2\\0\\-1\\-1\\0\\0\end{pmatrix}, \qquad e = \frac{1}{\sqrt{6}}\begin{pmatrix}0\\0\\2\\0\\0\\-1\\-1\end{pmatrix}, \quad f = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\0\\0\\1\\-1\\0\\0\end{pmatrix}, \quad g = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\0\\0\\0\\0\\1\\-1\end{pmatrix}.$$

The matrix

$$L = 0.9xx^T + 0.8yy^T + dd^T + ee^T + ff^T + gg^T,$$

is approximated as

$$L \approx \begin{pmatrix} 0.7101 & -0.1013 & -0.1354 & -0.1013 & -0.1013 & -0.1354 & -0.1354\\ -0.1013 & 0.8260 & -0.1255 & -0.1740 & -0.1740 & -0.1255 & -0.1255\\ -0.1354 & -0.1255 & 0.8373 & -0.1255 & -0.1255 & -0.1627 & -0.1627\\ -0.1013 & -0.1740 & -0.1255 & 0.8260 & -0.1740 & -0.1255 & -0.1255\\ -0.1013 & -0.1740 & -0.1255 & -0.1740 & 0.8260 & -0.1255 & -0.1255\\ -0.1354 & -0.1255 & -0.1627 & -0.1255 & -0.1255 & 0.8373 & -0.1627\\ -0.1354 & -0.1255 & -0.1627 & -0.1255 & -0.1255 & -0.1627 & 0.8373 \end{pmatrix}$$

and it is the Laplacian of the graph with weight matrix W

$$W \approx \begin{pmatrix} 0 & 0.1013 & 0.1354 & 0.1013 & 0.1013 & 0.1354 & 0.1354\\ 0.1013 & 0 & 0.1255 & 0.1740 & 0.1740 & 0.1255 & 0.1255\\ 0.1354 & 0.1255 & 0 & 0.1255 & 0.1255 & 0.1627 & 0.1627\\ 0.1013 & 0.1740 & 0.1255 & 0 & 0.1740 & 0.1255 & 0.1255\\ 0.1013 & 0.1740 & 0.1255 & 0.1740 & 0 & 0.1255 & 0.1255\\ 0.1354 & 0.1255 & 0.1627 & 0.1255 & 0.1255 & 0 & 0.1627\\ 0.1354 & 0.1255 & 0.1627 & 0.1255 & 0.1255 & 0.1627 & 0 \end{pmatrix}.$$

Thus x and y are the eigenvectors associated to the eigenvalues $\lambda_3 = 0.9$ and $\lambda_2 = 0.8$ respectively, that is $k = 2$. Finally we prove that z is a linear combination of x and y. We have

$$z = x \bullet x - y \bullet y = \frac{1}{154}\begin{pmatrix}-57\\-8\\27\\-8\\-8\\27\\27\end{pmatrix} = \alpha x + \beta y,$$

where

$$\alpha = \langle z, x\rangle = \frac{150}{11\sqrt{924}}, \qquad \beta = \langle z, y\rangle = -\frac{4\sqrt{3}}{11\sqrt{11}}.$$

Appendix D

Miscellaneous results

This chapter of the appendix is dedicated to some minor results that are used in the thesis.

Proposition D.0.1. *Let $f : \mathbb{C}^2 \to \mathbb{C}$ be a smooth function such that*

$$f(z, \overline{z}) = f(\overline{z}, z) \in \mathbb{R}, \qquad \forall z \in \mathbb{C}.$$

Then we have

$$\overline{\frac{\partial f(z,\overline{z})}{\partial z}} = \frac{\partial f(z,\overline{z})}{\partial \overline{z}}$$

Proof. We recall that, for $z = x + \mathrm{i}y$, the Wirtinger derivatives are defined as

$$\frac{\partial}{\partial z} = \frac{1}{2}\left(\frac{\partial}{\partial x} - \mathrm{i}\frac{\partial}{\partial y}\right), \qquad \frac{\partial}{\partial \overline{z}} = \frac{1}{2}\left(\frac{\partial}{\partial x} + \mathrm{i}\frac{\partial}{\partial y}\right)$$

and since $f(z, \overline{z}) = \overline{f(z, \overline{z})}$ we have

$$\overline{\frac{\partial f(z,\overline{z})}{\partial z}} = \overline{\frac{1}{2}\left(\frac{\partial f(z,\overline{z})}{\partial x} - \mathrm{i}\frac{\partial f(z,\overline{z})}{\partial y}\right)} = \frac{1}{2}\left(\overline{\frac{\partial f(z,\overline{z})}{\partial x}} + \mathrm{i}\overline{\frac{\partial f(z,\overline{z})}{\partial y}}\right) = \frac{\partial f(z,\overline{z})}{\partial \overline{z}}.$$

□

Proposition D.0.2. *Let $v, w \in \mathbb{R}^m$ be two linearly independent vectors, with $\|v\| = 1$ and let*

$$\mathcal{Z} = \{z \in \mathbb{R}^m : \|z\| = 1, \ \langle z, v\rangle = 0\},$$

where $\langle z, w\rangle = z^\top w$ and $\|v\| = \sqrt{v^\top v}$. Then

$$\arg\min_{z \in \mathcal{Z}} \langle w, z\rangle = \frac{-w + \langle v, w\rangle v}{\| - w + \langle v, w\rangle v\|} := z_*.$$

Proof. A possible proof of the claim is that for all $z \in \mathbb{R}^m$, it holds that

$$|\langle z, w\rangle| \le \|z\|\|w\| = \|w\|,$$

by means of Cauchy-Schwarz inequality and the equality is reached only for $z = \pm w$. Hence the global minimizer of the problem without constraints is $-w$. Thus the extremizer sought is the normalized projection onto the space $\{z \in \mathbb{R}^m : \langle z, v\rangle = 0\}$.

Now we give a rigorous proof of the proposition. First of all the vector z_* is an admissible solution since it has unit norm and

$$\langle z_*, v\rangle = \frac{-\langle w, v\rangle + \langle v, w\rangle\|v\|^2}{\| - w + \langle v, w\rangle v\|} = 0.$$

Let us consider a basis $\mathcal{B}$ of $\mathbb{R}^m$ of the form

$$\mathcal{B} = \{v, w, u_1, \dots, u_{m-2}\},$$

where

$$\langle v, u_i \rangle = \langle w, u_i \rangle = 0, \qquad \|u_i\| = 1, \qquad \langle u_i, u_j \rangle = 0 \qquad i, j = 1, \dots, m-2, \quad i \neq j,$$

that can be obtained, for instance, by extending $\{v, w\}$ to a basis of $\mathbb{R}^m$ and then by applying Gram-Schmidt algorithm. Thus each $z \in \mathbb{R}^m$ can be written as a real linear combination

$$z = \alpha v + \beta w + \sum_{i=1}^{m-2} \gamma_i u_i, \qquad \alpha, \beta \in \mathbb{R}, \qquad \gamma = (\gamma_1, \dots, \gamma_{m-2})^T \in \mathbb{R}^{m-2}.$$

Imposing that $z \in \mathcal{Z}$ yields

$$0 = \langle v, z \rangle = \alpha + \beta \langle w, v \rangle,$$

which means $\alpha = -\beta \langle v, w \rangle$. Thus the constraint on the norm and the orthogonality of the vectors of the basis imply

$$1 = \|z\|^2 = \|\alpha v + \beta w\|^2 + \sum_{i=1}^{m-2} \gamma_i^2 = \beta^2 \|w - \langle v, w \rangle v\|^2 + \|\gamma\|^2,$$

that is

$$\beta^2 = \frac{1 - \|\gamma\|^2}{\|w - \langle v, w \rangle v\|^2}.$$

The quantity we want to minimize can be rewritten as

$$\langle z, w \rangle = \alpha \langle v, w \rangle + \beta \|w\|^2 = \beta \left(\|w\|^2 - \langle v, w \rangle^2 \right)$$

and, since $\left(\|w\|^2 - \langle v, w \rangle^2 \right) > 0$ by means of the Cauchy-Schwarz inequality, the optimal choice for β is the negative value

$$\beta_\star(\gamma) = -\sqrt{\frac{1 - \|\gamma\|^2}{\|w - \langle v, w \rangle v\|^2}},$$

which reaches its minimum for $\gamma = 0$. Hence

$$\arg\min_{z \in \mathcal{Z}} \langle z, w \rangle = \arg\min_{\gamma \in \mathbb{R}^{m-2}} \left\{ \beta_\star(\gamma) \left(\|w\|^2 - \langle v, w \rangle^2 \right) \right\}$$

and the unique solution, obtained for $\gamma = 0$, is

$$z_\star = \beta_\star(0) \cdot (-\langle v, w \rangle v + w) = \frac{-w + \langle v, w \rangle v}{\|w - \langle v, w \rangle v\|}.$$

□

Proposition D.0.3. *Let $A \in \mathbb{C}^{n \times n}$. Then*

$$\ker(A)^\perp = \operatorname{range}(A^*).$$

Proof. We have

$$\ker(A) = \{x \in \mathbb{C}^n \ : \ Ax = 0\} = \{x \in \mathbb{C}^n \ : \ y^* Ax = 0, \ \forall y \in \mathbb{C}^n\} =$$

$$= \{x \in \mathbb{C}^n \ : \ z^* x = 0, \ \forall z \in \operatorname{range}(A^*)\} = \operatorname{range}(A^*)^\perp,$$

which yields the claim by taking the orthogonal of the equality. □

Proposition D.0.4. *Let $E, G \in \mathbb{C}^{n\times n}$ be two matrices with the same kernel and range. Consider an SVD-like (see* (2.12)*) $E = USV^*$, with $S \in \mathbb{C}^{r\times r}$. Then*

$$UU^*G = G = GVV^*.$$

Proof. For all $x \in \mathbb{C}^n$, it holds that

$$Gx = GVV^*x + G(I - VV^*)x = GVV^*x,$$

since $(I - VV^*)x \in \ker(E) = \ker(G)$. Similarly,

$$G^*UU^*x = G^*UU^*x + G^*(I - UU^*)x = G^*UU^*x,$$

since $(I - UU^*)x \in \ker(E^*) = \operatorname{range}(E)^\perp = \operatorname{range}(G)^\perp = \ker(G^*)$. Thus the arbitrariness of x implies the claim. □

Proposition D.0.5. *Given a subspace $\mathcal{E} \subset \operatorname{sym}(\mathbb{R}^{n\times n})$, the restricted Laplacian operator $L : \mathcal{E} \to \mathbb{R}^{n\times n}$ is defined as*

$$L(W) = \operatorname{diag}(W\mathbb{1}) - W, \qquad \mathbb{1} = (1, \ldots, 1)^\top, \qquad \forall W \in \mathcal{E}.$$

Let $L^* : \mathbb{R}^{n\times n} \to \mathcal{E}$ be the adjoint of L with respect to the Frobenius inner product such that, for all $V \in \mathbb{R}^{n\times n}$ and for all $W \in \mathcal{E}$

$$\langle W, L^*(V)\rangle = \langle L(W), V\rangle.$$

Then

$$L^*(V) = \Pi_{\mathcal{E}}(\operatorname{diagvec}(V)\mathbb{1}^\top - V),$$

where $\operatorname{diagvec}(V) \in \mathbb{R}^n$ *is the vector of the diagonal entries of V and $\Pi_{\mathcal{E}}$ is the orthogonal projection with respect to the Frobenius inner product onto $\mathcal{E}$.*

Proof. For all $V \in \mathbb{R}^{n\times n}$ and all $W \in \operatorname{sym}(\mathbb{R}^{n\times n})$ it holds that

$$\langle \operatorname{diag}(W\mathbb{1}), V\rangle = \sum_{i=1}^n \left(\sum_{j=1}^n w_{i,j} \right) v_{i,i} = \sum_{i=1}^n \sum_{j=1}^n v_{j,j} w_{j,i} = \langle \operatorname{diagvec}(V)\mathbb{1}^\top, W\rangle$$

and hence

$$\langle L(W), V\rangle = \langle W, \operatorname{diagvec}(V)\mathbb{1}^\top - V\rangle = \left\langle W, \Pi_{\mathcal{E}} \left(\operatorname{diagvec}(V)\mathbb{1}^\top - V \right) \right\rangle,$$

which, by definition, shows the claim for L^*. □

Proposition D.0.6. *With the same notation of Proposition D.0.6, assume that, for all $W = (w_{i,j}) \in \mathcal{E}$, then $w_{i,i} = 0$ for $i = 1, \ldots n$, i.e. W has no self-loops. Then*

$$L^*(L(W)) = \Pi_{\mathcal{E}}(W\mathbb{1}\mathbb{1}^\top) + W.$$

Proof. By assumption $\mathrm{diagvec}(W) = 0 = \Pi_{\mathcal{E}}(\mathrm{diag}(W\mathbb{1}))$ for all $W \in \mathcal{E}$. Thus, since $\mathrm{diagvec}(\mathrm{diag}(v)) = v$ for all $v \in \mathbb{R}^n$, the formula for L^* shown in Proposition D.0.6 yields

$$L^*(L(W)) = \Pi_{\mathcal{E}}\left(\mathrm{diagvec}(\mathrm{diag}(W\mathbb{1}) - W)\mathbb{1}^\top - \mathrm{diag}(W\mathbb{1}) + W\right) =$$

$$= \Pi_{\mathcal{E}}(W\mathbb{1}\mathbb{1}^\top) - \Pi_{\mathcal{E}}(\mathrm{diagvec}(W)\mathbb{1}^\top) - \Pi_{\mathcal{E}}(\mathrm{diag}(W\mathbb{1})) + \Pi_{\mathcal{E}}(W) = \Pi_{\mathcal{E}}(W\mathbb{1}\mathbb{1}^\top) + W.$$

□

Bibliography

[1] Pierre-Antoine Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds.* Princeton University Press, 2008.

[2] Rafikul Alam, Shreemayee Bora, Michael Karow, Volker Mehrmann, and Julio Moro. "Perturbation theory for Hamiltonian matrices and the distance to bounded-realness". In: *SIAM Journal on Matrix Analysis and Applications* 32.2 (2011), pp. 484–514.

[3] Eleonora Andreotti, Dominik Edelmann, Nicola Guglielmi, and Christian Lubich. "Measuring the stability of spectral clustering". In: *Linear Algebra and its Applications* 610 (2021), pp. 673–697.

[4] Joseph E Avron and Barry Simon. "Analytic properties of band functions". In: *Annals of Physics* 110.1 (1978), pp. 85–101.

[5] Ronald F Boisvert, Roldan Pozo, Karin Remington, Richard F Barrett, and Jack J Dongarra. "Matrix Market: a web resource for test matrix collections". In: *Quality of Numerical Software: Assessment and Enhancement* (1997), pp. 125–137.

[6] Nicolas Boumal, Bamdev Mishra, Pierre-Antoine Absil, and Rodolphe Sepulchre. "Manopt, a Matlab toolbox for optimization on manifolds". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1455–1459.

[7] Roger W Brockett. "Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems". In: *Linear Algebra and its applications* 146 (1991), pp. 79–91.

[8] Angelika Bunse-Gerstner, Ralph Byers, Volker Mehrmann, and Nancy K Nichols. "Numerical computation of an analytic singular value decomposition of a matrix valued function". In: *Numerische Mathematik* 60 (1991), pp. 1–39.

[9] James V Burke, Adrian S Lewis, and Michael L Overton. "A nonsmooth, nonconvex optimization approach to robust stabilization by static output feedback and low-order controllers". In: *IFAC Proceedings Volumes* 36.11 (2003), pp. 175–181.

[10] R Cameron and B Kouvaritakis. "Relative stability margins of multivariable systems A characteristic locus approach". In: *International Journal of Control* 30.4 (1979), pp. 629–651.

[11] Stephen L Campbell and Carl D Meyer. *Generalized inverses of linear transformations.* SIAM, 2009.

[12] Emmanuel J Candès and Terence Tao. "The power of convex relaxation: Near-optimal matrix completion". In: *IEEE transactions on information theory* 56.5 (2010), pp. 2053–2080.

[13] Gianluca Ceruti, Jonas Kusch, and Christian Lubich. "A rank-adaptive robust integrator for dynamical low-rank approximation". In: *BIT Numerical Mathematics* (2022), pp. 1–26.

[14] Gianluca Ceruti and Christian Lubich. “An unconventional robust integrator for dynamical low-rank approximation”. In: *BIT Numerical Mathematics* 62.1 (2022), pp. 23–44.

[15] Ricky Tian Qui Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. “Neural ordinary differential equations”. In: *Advances in neural information processing systems* 31 (2018).

[16] Moody T Chu. “Linear algebra algorithms as dynamical systems”. In: *Acta numerica* 17 (2008), pp. 1–86.

[17] Timothy A Davis and Yifan Hu. “The University of Florida sparse matrix collection”. In: *ACM Transactions on Mathematical Software (TOMS)* 38.1 (2011), pp. 1–25.

[18] Vaclav Doležal. “The existence of a continuous basis of a certain linear subspace of E_r which depends on a parameter”. In: *Časopis pro pěstování matematiky* 89.4 (1964), pp. 466–469.

[19] Miroslav Fiedler. “Algebraic connectivity of graphs”. In: *Czechoslovak mathematical journal* 23.2 (1973), pp. 298–305.

[20] Nicolas Gillis and Punit Sharma. “On computing the distance to stability for matrices using linear dissipative Hamiltonian systems”. In: *Automatica* 85 (2017), pp. 113–121.

[21] Miryam Gnazzo, Vanni Noferini, Lauri Nyman, and Federico Poloni. “Riemann-Oracle: A general-purpose Riemannian optimizer to solve nearness problems in matrix theory”. In: *arXiv preprint arXiv:2407.03957* (2024).

[22] Serge Konstantinovich Godunov. *Ordinary differential equations with constant coefficient.* Vol. 169. American Mathematical Soc., 1997.

[23] Anne Greenbaum, Ren-cang Li, and Michael L Overton. “First-order perturbation theory for eigenvalues and eigenvectors”. In: *SIAM review* 62.2 (2020), pp. 463–482.

[24] Nicola Guglielmi. “On the method by Rostami for computing the real stability radius of large and sparse matrices”. In: *SIAM Journal on Scientific Computing* 38.3 (2016), A1662–A1681.

[25] Nicola Guglielmi, Arturo De Marinis, Anton Savastianov, and Francesco Tudisco. “Contractivity of neural ODEs: an eigenvalue optimization problem”. In: *arXiv preprint arXiv:2402.13092* (2024).

[26] Nicola Guglielmi, Mert Gürbüzbalaban, and Michael L Overton. “Fast approximation of the H_∞ norm via optimization over spectral value sets”. In: *SIAM Journal on Matrix Analysis and Applications* 34.2 (2013), pp. 709–737.

[27] Nicola Guglielmi, Daniel Kressner, and Christian Lubich. “Computing extremal points of symplectic pseudospectra and solving symplectic matrix nearness problems”. In: *SIAM Journal on Matrix Analysis and Applications* 35.4 (2014), pp. 1407–1428.

[28] Nicola Guglielmi, Daniel Kressner, and Christian Lubich. “Low rank differential equations for Hamiltonian matrix nearness problems”. In: *Numerische Mathematik* 129.2 (2015), pp. 279–319.

[29] Nicola Guglielmi and Christian Lubich. “Differential equations for roaming pseudospectra: paths to extremal points and boundary tracking”. In: *SIAM Journal on Numerical Analysis* 49.3 (2011), pp. 1194–1209.

[30] Nicola Guglielmi and Christian Lubich. “Low-rank dynamics for computing extremal points of real pseudospectra”. In: *SIAM Journal on Matrix Analysis and Applications* 34.1 (2013), pp. 40–66.

[31] Nicola Guglielmi and Christian Lubich. *Matrix nearness problems and eigenvalue optimization.* 2022.

[32] Nicola Guglielmi and Christian Lubich. “Matrix stabilization using differential equations”. In: *SIAM Journal on Numerical Analysis* 55.6 (2017), pp. 3097–3119.

[33] Nicola Guglielmi, Christian Lubich, and Volker Mehrmann. “On the nearest singular matrix pencil”. In: *SIAM Journal on Matrix Analysis and Applications* 38.3 (2017), pp. 776–806.

[34] Nicola Guglielmi, Christian Lubich, and Stefano Sicilia. “Rank-1 Matrix Differential Equations for Structured Eigenvalue Optimization.” In: *SIAM Journal on Numerical Analysis* 61.4 (2023), pp. 1737–1762.

[35] Nicola Guglielmi and Manuela Manetta. “Approximating real stability radii”. In: *IMA Journal of Numerical Analysis* 35.3 (2015), pp. 1402–1425.

[36] Nicola Guglielmi and Michael L Overton. “Fast algorithms for the approximation of the pseudospectral abscissa and pseudospectral radius of a matrix”. In: *SIAM Journal on Matrix Analysis and Applications* 32.4 (2011), pp. 1166–1192.

[37] Nicola Guglielmi and Stefano Sicilia. “A low-rank ODE for spectral clustering stability”. In: *Linear Algebra and its Applications* (2024).

[38] Nicola Guglielmi and Stefano Sicilia. “Stabilization of a matrix via a low-rank-adaptive ODE”. In: *BIT Numerical Mathematics* 64.4 (2024), p. 38.

[39] Victor Guillemin and Alan Pollack. *Differential topology.* Vol. 370. American Mathematical Soc., 2010.

[40] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. “The elements of statistical learning. Springer series in statistics”. In: *New York, NY, USA* (2001).

[41] Sarah-Alexa Hauschild, Nicole Marheineke, Volker Mehrmann, and Lehrstuhl Modellierung und Numerik. “Model reduction techniques for linear constant coefficient port-Hamiltonian differential-algebraic systems”. In: *Control and Cybernetics* 48.1 (2019).

[42] Uwe Helmke and John B Moore. *Optimization and dynamical systems.* Springer Science & Business Media, 2012.

[43] Joao P Hespanha. “An efficient matlab algorithm for graph partitioning”. In: *University of California* (2004), pp. 1–8.

[44] Nicholas John Higham. “Computing a nearest symmetric positive semidefinite matrix”. In: *Linear algebra and its applications* 103 (1988), pp. 103–118.

[45] Nicholas John Higham. “Computing the nearest correlation matrix—a problem from finance”. In: *IMA journal of Numerical Analysis* 22.3 (2002), pp. 329–343.

[46] Nicholas John Higham. *Matrix nearness problems and applications.* University of Manchester. Department of Mathematics, 1988.

[47] Diederich Hinrichsen and Anthony J Pritchard. *Mathematical systems theory I: modelling, state space analysis, stability and robustness.* Vol. 48. Springer, 2005.

[48] Roger A. Horn and Charles R Johnson. *Matrix analysis.* Cambridge University Press, 1985.

[49] Michael Karow, Effrosyni Kokiopoulou, and Daniel Kressner. “On the computation of structured singular values and pseudospectra”. In: *Systems & control letters* 59.2 (2010), pp. 122–129.

[50] Tosio Kato. *Perturbation theory for linear operators*. Vol. 132. Springer Science & Business Media, 2013.

[51] Othmar Koch and Christian Lubich. “Dynamical low-rank approximation”. In: *SIAM Journal on Matrix Analysis and Applications* 29.2 (2007), pp. 434–454.

[52] Daniel Kressner and Bart Vandereycken. “Subspace methods for computing the pseudospectral abscissa and the stability radius”. In: *SIAM Journal on Matrix Analysis and Applications* 35.1 (2014), pp. 292–313.

[53] Daniel Kressner and Matthias Voigt. “Distance problems for linear dynamical systems”. In: *Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory: Festschrift in Honor of Volker Mehrmann* (2015), pp. 559–583.

[54] Jure Leskovec and Julian Mcauley. “Learning to discover social circles in ego networks”. In: *Advances in neural information processing systems* 25 (2012).

[55] Jan R Magnus. “On differentiating eigenvalues and eigenvectors”. In: *Econometric theory* 1.2 (1985), pp. 179–191.

[56] M Mansour, Eliahu I Jury, and Luis F Chaparro. “Estimation of the margin of stability for linear continuous and discrete systems”. In: *International Journal of Control* 30.1 (1979), pp. 49–69.

[57] Volker Mehrmann and Hongguo Xu. “Perturbation of purely imaginary eigenvalues of Hamiltonian matrices under structured perturbations”. In: *The Electronic Journal of Linear Algebra* 17 (2008), pp. 234–257.

[58] Carl D Meyer and Gilbert W Stewart. “Derivatives and perturbations of eigenvectors”. In: *SIAM Journal on Numerical Analysis* 25.3 (1988), pp. 679–691.

[59] Vanni Noferini and Federico Poloni. “Nearest Ω-stable matrix via Riemannian optimization”. In: *Numerische Mathematik* 148.4 (2021), pp. 817–851.

[60] Francois-Xavier Orbandexivry, Yurii Nesterov, and Paul Van Dooren. “Nearest stable system using successive convex approximations”. In: *Automatica* 49.5 (2013), pp. 1195–1203.

[61] Dimosthenis Pasadakis, Christie Louis Alappat, Olaf Schenk, and Gerhard Wellein. “Multiway p-spectral graph cuts on Grassmann manifolds”. In: *Machine Learning* (2022), pp. 1–39.

[62] Li Qiu, Bo Bernhardsson, Anders Rantzer, Edward Joseph Davison, Peter Michael Young, and John C Doyle. “A formula for computation of the real stability radius”. In: *Automatica* 31.6 (1995), pp. 879–890.

[63] Yousef Saad. *Numerical methods for large eigenvalue problems: revised edition*. SIAM, 2011.

[64] Gilbert W Stewart. “A Krylov–Schur algorithm for large eigenproblems”. In: *SIAM Journal on Matrix Analysis and Applications* 23.3 (2002), pp. 601–614.

[65] Françoise Tisseur and Nicholas John Higham. “Structured pseudospectra for polynomial eigenvalue problems, with applications”. In: *SIAM Journal on Matrix Analysis and Applications* 23.1 (2001), pp. 187–208.

[66] Lloyd Nick Trefethen. “Pseudospectra of matrices”. In: *Numerical analysis* 91 (1991), pp. 234–266.

[67] Lloyd Nick Trefethen. "Spectra and pseudospectra: the behavior of nonnormal matrices and operators". In: (2020).

[68] Ulrike Von Luxburg. "A tutorial on spectral clustering". In: *Statistics and computing* 17 (2007), pp. 395–416.