Optimal Model Regularization for Wafer Alignment in Lithography

by

A. A. Bonke

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday April 25th, 2025

Supervisors:	Dr. E. M. Hulsebos,	ASML FC-061 Wafer Alignment
	Dr. D. de Laat,	DIAM Discrete Mathematics and Optimization
Thesis committee:	Dr. E. M. Hulsebos,	ASML FC-061 Wafer Alignment
	Dr. D. de Laat,	DIAM Discrete Mathematics and Optimization
	Dr. C. A. Urzúa Torres,	DIAM Numerical Analysis

Faculty: Electrical Engineering, Mathematics and Computer Science

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft ASML

Abstract

This thesis addresses the novel double-bounded positive semidefinite Procrustes problem, defined as

$$\begin{aligned} \min_{A} \quad & \|Y - AX\|_F \\ \text{s.t.} \quad & A \succeq 0 \\ & B - A \succeq 0, \end{aligned}$$

which arises from the optimal model regularization problem for wafer alignment

$$R = \underset{R \succeq R_{min}}{\operatorname{argmin}} \quad \left\| M_y Y - M_y (M_X^T M_X + R)^{-1} M_X^T x \right\|_F .$$

Despite convexity in the optimization variable A, solving the problem is challenging due to the presence of both an upper and a lower bound, both of which introduce nonlinearity. Several numerical methods have been proposed, including semidefinite programming, alternating projection, and projected gradient methods. Among these, the projected gradient method proves to be the most efficient: by decomposing $B = QQ^T$ and transforming the variables, the constraints simplify to $I \succeq A' \succeq 0$, allowing for straightforward projection onto the feasible region using eigenvalue decomposition.

The optimal regularization approach was tested on various customer datasets, demonstrating overlay improvements of several tens to hundreds of picometers in most cases. However, the method failed for datasets with significant deformation and high variance among wafers, due to incomplete optimization of the regularization matrix. Future research should focus on developing a more sophisticated scheme for combining data-driven optimal regularization with standard bending energy matrices, and extensively validating the method across diverse datasets for accuracy, consistency, and performance.

Contents

1 Introduction

Photolithography enables the fabrication of intricate semiconductor designs by using light to transfer chip designs onto a silicon wafer. These designs are three-dimensional, requiring a multitude of layers to be manufactured successively, each layer aligned to the previous layer with nanometer accuracy. To achieve this, a model of the wafer is constructed using optical measurements of wafer deformation. Linear regression is employed to this end, with Tikhonov regularization commonly used to suppress measurement errors and noise. This prompts the optimal regularization problem

$$R = \underset{R \geq R_{min}}{\operatorname{argmin}} \quad \left\| M_y Y - M_y (M_x^T M_x + R)^{-1} M_x^T x \right\|_F, \tag{1.1}$$

where R is the regularization matrix, R_{min} represents a minimum regularization matrix, Y contains the optimal model coefficients, $(M_x^T M_x + R)^{-1} M_x^T x$ has the fitted model coefficients and M_y is a model matrix allowing us to evaluate the fitted coefficients. Optimizing over R directly is difficult due to the matrix inverse, prompting the substitution $A = (M_x^T M_x + R)^{-1}$. Simplifying and rearranging, this yields a problem of the form

$$\begin{aligned} \min_A \quad & \|Y' - AX\|_F \\ \text{s.t.} \quad & A \succeq 0 \\ & B - A \succeq 0, \end{aligned} \tag{1.2}$$

which we will call the double-bounded Procrustes[1] problem, after a family of similar matrix nearness problems. Both the symmetric problem, having only the constraint $A = A^T$, and the single-bounded problem, having only the constraint $A \succeq 0$, have been studied in previous literature. The former problem admits a closed-form solution [1], whereas for the latter problem, a closed form solution is only known for some special cases [2]. The double-bounded problem as defined above has not been studied before. A notable aspect of 1.2 is that not only the spectrum of the matrix is constrained (such is the case in the single-bounded version), but also the eigenvectors of the matrix are constrained to satisfy $B \succeq A$. Furthermore, for positive definite B, the double-bounded problem always attains its infimum, whereas the single-bounded problem may not. An obvious transformation would be to set $A = L^T L$ and optimize over L, thus ensuring positive semidefiniteness of A. This, however, has the disadvantage of not being convex in L [3], and not having a straightforward way to deal with the upper bound. In this thesis, we will apply techniques from convex optimization to derive multiple schemes for the solution of the double-bounded Procrustes problem.

1.1. Structure of the thesis

Chapter 2 will explain the process of lithography in more detail.

[1] Procrustes is a character from Greek mythology who invited guests to his home fit his bed exactly by either stretching their limbs or sawing part of them off.

Next, Chapter 3 will provide theoretical background for wafer alignment and formalize the mathematical framework underpinning wafer modelling.

Using this framework, the optimal regularization problem will be posed and formalized in Chapter 4.

Chapter 5 contains an analysis of the problem's properties, which is applied in subsequent chapters.

First, Chapter 6 gives a semidefinite programming solution, which has the advantage of being relatively straightforward to implement using an off-the-shelf solver. However, such solvers are usually slow to solve the problem, and are difficult to port to other platforms, prompting the search for other methods.

In Chapter 7 we develop some theory around the single-bounded and unbounded/symmetric relaxations of the problem, which is required to study the alternative schemes laid out in the following chapters.

Chapter 8 details an alternating projection scheme, solving the single-bounded problem twice at each iteration to alternatingly project on the lower and upper bound. This is already more efficient than the SDP method but still requires solving the single-bounded problem efficiently. As of now, only numerical solutions are available, hindering performance.

For even better performance, Chapter 9 provides a gradient method. We still need to project on the feasible region, which is the intersection of two convex sets. Usually, we would need to use e.g. Dykstra's algorithm to this end, but in our specific case we may apply a transformation such that the projection on the feasible region is simple and only requires one eigendecomposition.

Accompanying the solution methods, Chapter 10 gives details for the pre- and postprocessing necessary to get from alignment/overlay data to a usable regularization matrix.

The final chapters 11, 12 and 13 show test results for the optimal regularization scheme, utilizing several different customer datasets, and discuss the findings. Furthermore, the different solution schemes are compared in terms of performance and other practical aspects.

2

Photolithography

Semiconductor manufacturing involves hundreds, often thousands, of process steps that have to be carried out with accuracy on the nanometer scale. By far the most common process for semiconductor manufacturing is called photolithography, in which light is used to transfer a chip design onto a circular slice of extremely pure silicon called a wafer. A series of chemical and mechanical processes precede and follow the lithography step, etching the design into silicon and changing the electrical properties of the substrate. Modern microchips consist of (hundreds of) billions of transistors on a silicon substrate. These transistors act like electrical switches, such that arrangements of transistors can perform complex tasks. In order to fit billions of transistors in a small package, each transistor has to be only a few nanometers in size. The process which enables the printing of such fine details involves many steps, which can be broadly divided into six parts [4].

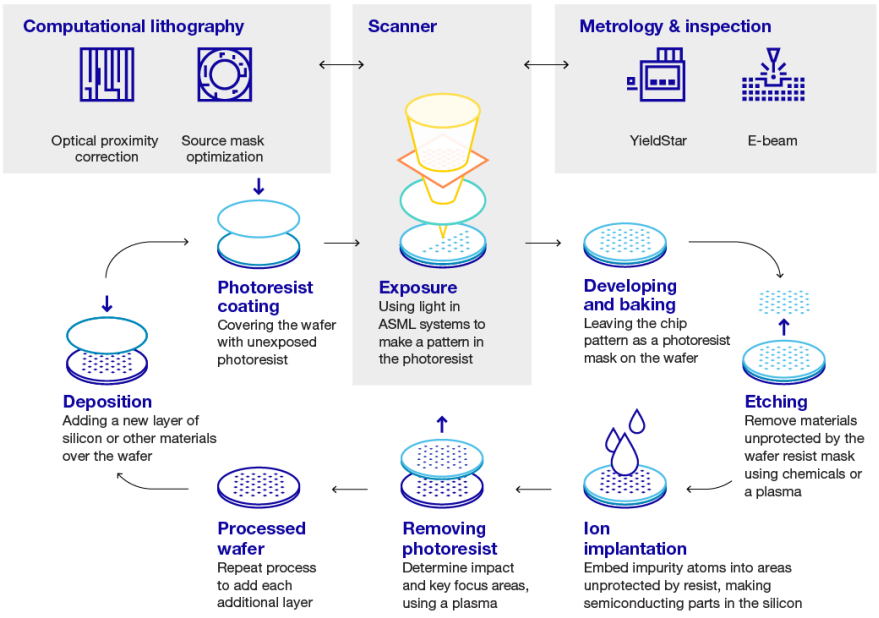

Figure 2.1: Different stages of semiconductor production. (Source: ASML Academy, "1 - Introduction to Lithography", ASML)

2.1. Deposition

Microchips are produced on a silicon substrate known as a wafer. Wafers are sliced from a cylindrical ingot made of extremely pure silicon, produced from refined silica sand. The first step in the production

of microchips is to deposit a very thin layer of a conducting, isolating or semiconducting material onto the wafer. Transistors are then etched into this layer.

2.2. Photoresist coating

Next, a photoresist layer is deposited onto the wafer. A photoresist is a material which is altered chemically by exposure to light. There are two types of photoresists: positive, which become more soluble when exposed to light, and negative, which polymerize and become less soluble when exposed. Positive photoresists are most common because of their higher resolution capability.

2.3. Photolithography

Photolithography or simply lithography, is the process which transfers the chip design onto the wafer. Light is projected onto the wafer through the reticle, which holds the chip design pattern. Optics in the lithography machine shrink and focus the pattern onto the wafer, exposing the photoresist layer in exactly the right places. Lithography presents many challenges. For example, the wavelength of the light used must be extremely small in order to etch ever smaller details. Any deformation, misalignment or foreign particle, however small, may result in the microchip not functioning as intended. Machines that carry out lithography with extreme precision belong to ASML's most crucial and well-known products, including the TWINSCAN EXE and NXE systems, which are used in almost all high-end chip manufacturing today.

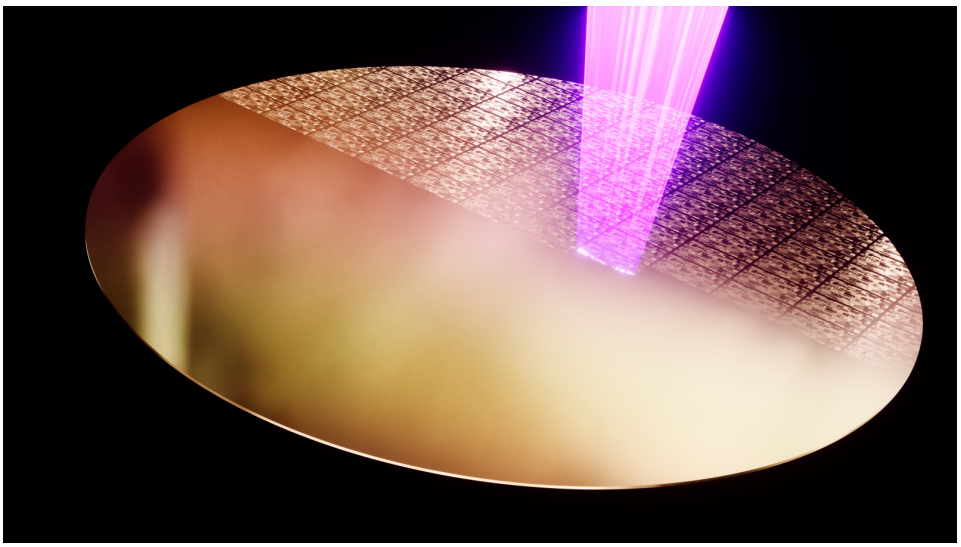

Figure 2.2: Cartoon image of a wafer being exposed. (Source: ASML Media Library, "ASML High NA EUV lithography stills made from animation", ASML)

2.4. Etching

The exposed photoresist layer now needs to be washed away to reveal the produced pattern. The wafer is baked and developed, not unlike the process for developing photographs. Liquid or gaseous chemicals are used to remove the degraded photoresist layer.

2.5. Ion implantation

The electrical properties of silicon, which is a semiconductor, can be changed by implanting either positively or negatively charged ions into the material. The alternating of so called p-doped and n-doped layers enables the transistor to control the flow of electricity. After ion implantation, the remaining photoresist layer that was protecting parts of the wafer is now removed.

2.6. Packaging

The above steps are repeated to produce a chip with multiple layers. To increase throughput, multiple chips at a time are produced on a wafer, which is most commonly 300 mm in size. Individual chips are sawed out from this wafer and encapsulated in a protective container. Metal foils on a substrate known as the 'baseboard' direct electrical signals into and out of the chip.

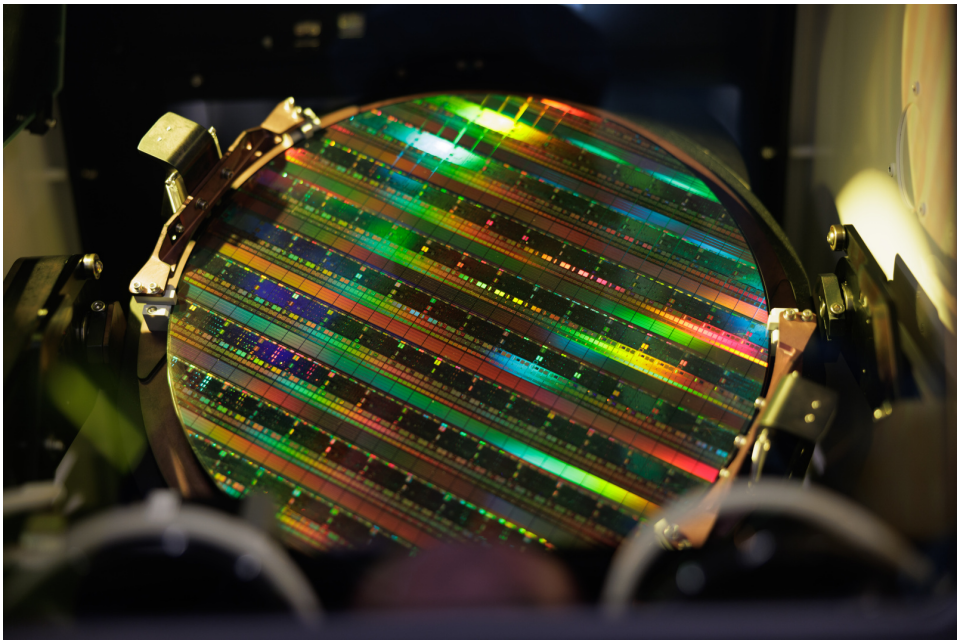

Figure 2.3: A finished wafer. (Source: ASML Media Library, "ASML 2023 wafer insection", ASML)

3

Wafer alignment

Modern integrated circuits consist of many layers, which electrically interconnect to give the chip its intended functionality. Layers have to be manufactured on top of one another with nanometer accuracy: misalignments, however small, can hinder performance of the chip or render it non-functioning. The challenge with this is that the position and shape of the wafer change after every process step: a wafer is loaded into the scanner[1] with varying offset, warped by electrostatic clamping, and deformed by mechanical and chemical processing steps.

This motivates the development of highly precise measurement techniques to measure the position and deformation of the wafer. Modelling the wafer deformation and position precisely allows us to compensate both mechanically (by moving the wafer) and optically (adjusting the light pattern). Before exposure (using light to transfer a chip design onto a wafer), the wafer is measured at multiple points. A model of the shape of the entire wafer is then constructed from these points. There is a critical tradeoff here: measuring more points will give us a better model of the wafer shape, but extensive measurement takes more time. Photolithography machines are extremely expensive to buy and operate, and longer measurement time reduces throughput (the number of wafers processed per unit time) and thus profitability.

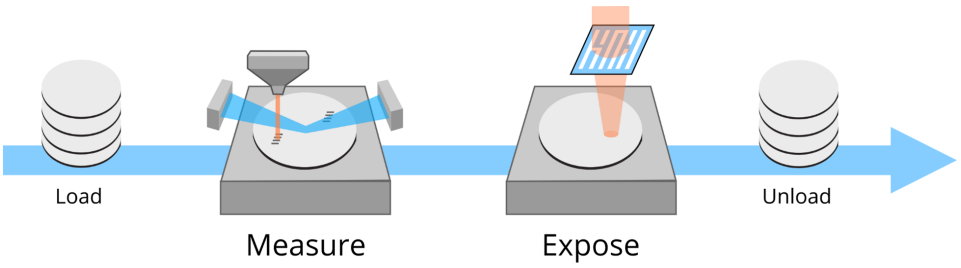

Figure 3.1: The load-measure-expose-unload cycle. (Source: MCPZ, "Wafer Alignment in 30 minutes", ASML)

A multitude of hardware and software technologies are employed to construct the most accurate model of a wafer given a limited number of measurements. This discipline, mapping the deformation and position of a wafer in the x/y-plane, is called wafer alignment. Points on the wafer are measured by etching diffraction gratings (referred to as alignment marks) onto the wafer and diffracting light off of these marks.

[1] A lithography machine which uses light to transfer a chip design onto a wafer.

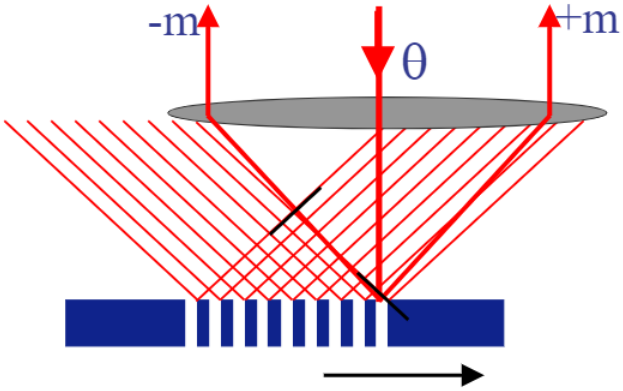

Figure 3.2: Wavefronts diffracted by a grating. (Source: WDRE, "SMASH Optical Module – MEA-AL course", ASML)

When the wafer is moved as the sensor's light beam hits the alignment mark, the phase of the wavefronts changes. Using an interferometer, this phase change can be measured as an intensity signal, with a periodicity proportional to the spacing between the lines within a mark. Measuring this signal in both the x- and y-directions allows us to determine the position of alignment marks on the wafer.

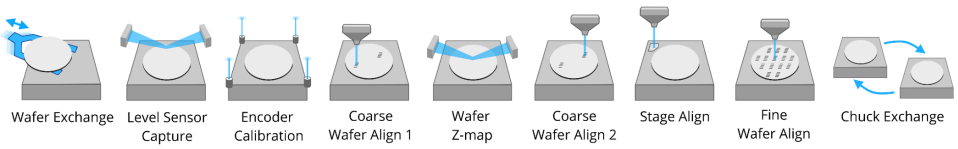

Figure 3.3: Different stages of the wafer alignment process. (Source: MCPZ, "Wafer Alignment in 30 minutes", ASML)

We cannot place marks just anywhere on the wafer, as this could disturb the pattern being etched to make a functioning chip. Typically, 80 mark pairs (each pair having one x-directional and one y-directional mark) distributed over the wafer are measured to enable approximation of the deformation everywhere on the wafer.

A model is fit to these measurements for two reasons. Firstly, a model describing the entire wafer allows us to interpolate/extrapolate the measurements to all other locations on the wafer. Secondly, fitting suppresses measurement noise and errors. To this end, parameterized functions are used which are evaluated on the nominal positions (i.e. where the marks should be in the pattern design) of the marks. We can model the aligned position deviations, i.e. the displacement of the marks relative to their respective nominal positions, as follows:

$$\begin{bmatrix} \mathrm{apd}_x \\ \mathrm{apd}_y \end{bmatrix} = X_1 \begin{bmatrix} f_{1,x}(x_{\mathrm{nom}}, y_{\mathrm{nom}}) \\ f_{1,y}(x_{\mathrm{nom}}, y_{\mathrm{nom}}) \end{bmatrix} + \cdots + X_m \begin{bmatrix} f_{m,x}(x_{\mathrm{nom}}, y_{\mathrm{nom}}) \\ f_{m,y}(x_{\mathrm{nom}}, y_{\mathrm{nom}}) \end{bmatrix} + \varepsilon \tag{3.1}$$

$$= \begin{bmatrix} f_{1,x}(x_{\mathrm{nom}}, y_{\mathrm{nom}}) & \cdots & f_{m,x}(x_{\mathrm{nom}}, y_{\mathrm{nom}}) \\ f_{1,y}(x_{\mathrm{nom}}, y_{\mathrm{nom}}) & \cdots & f_{m,y}(x_{\mathrm{nom}}, y_{\mathrm{nom}}) \end{bmatrix} \begin{bmatrix} X_1 \\ \vdots \\ X_m \end{bmatrix} + \varepsilon, \tag{3.2}$$

where $x_{\mathrm{nom}}, y_{\mathrm{nom}}$ are the nominal x- and y-coordinates of the mark (i.e. where the mark is on the design to be etched) and ε is a noncorrectable residual term. Evaluating this model on n alignment mark pairs can be written as

$$\begin{bmatrix} \overline{\mathrm{apd}_x} \\ \overline{\mathrm{apd}_y} \end{bmatrix} = M_x X, \tag{3.3}$$

where

$$M_x = \begin{bmatrix} f_{1,x}(x_{nom,x_1}, y_{nom,x_1}) & \cdots & f_{m,x}(x_{nom,x_1}, y_{nom,x_1}) \\ \vdots & \ddots & \vdots \\ f_{1,x}(x_{nom,x_n}, y_{nom,x_n}) & \cdots & f_{m,x}(x_{nom,x_n}, y_{nom,x_n}) \\ f_{1,y}(x_{nom,y_1}, y_{nom,y_1}) & \cdots & f_{m,y}(x_{nom,y_1}, y_{nom,y_1}) \\ \vdots & \ddots & \vdots \\ f_{1,y}(x_{nom,y_n}, y_{nom,y_n}) & \cdots & f_{m,y}(x_{nom,y_n}, y_{nom,y_n}) \end{bmatrix}. \tag{3.4}$$

In this matrix, mark x_i may be different from mark y_i if the marks are unidirectional, meaning that the marks have a grating which can only be used to measure position in one direction. For example, x_1 indicates the first mark we use to measure position in the x-direction, and y_1 indicates the first mark we use to measure position in the y-direction. For bidirectional marks, that is, marks that can be used to measure both x- and y-directional position deviations, x_i and y_i refer to the same mark.

Least-squares optimal $\bar{X}$ is then readily given by

$$\bar{X} = (M_x^T M_x)^{-1} M_x^T \begin{bmatrix} apd_x \\ apd_y \end{bmatrix}, \tag{3.5}$$

where x, y are the measured x- and y-coordinates of the mark.

Usually, we write x instead of $\begin{bmatrix} apd_x \\ apd_y \end{bmatrix}$ to denote both the x- and y-directional position deviations, stacked on top of each other:

$$x = M_x X + \varepsilon, \tag{3.6}$$

giving

$$\bar{X} = (M_x^T M_x)^{-1} M_x^T x. \tag{3.7}$$

This framework can also be used to fit coefficients for multiple wafers at once: x and X will be matrices, with columns corresponding to individual wafers.

This process can be roughly divided into two stages: COarse Wafer Alignment (COWA) and FIne Wafer Alignment (FIWA). The simplest version of COWA measures 4 marks: 2 in the x-direction and 2 in the y-direction. These 4 measurements allow us to fit 4 unknowns: translation in the x-direction, translation in the y-direction, symmetric magnification and symmetric rotation. This symmetric model is however lacking when the wafer is deformed asymmetrically, we need an extra 2 marks (one x, one y) to additionally measure asymmetric magnifcation and non-orthogonality.

After COWA, FIWA refines the wafer position measurement and measures wafer deformation at a much finer level than COWA. This thesis will focus on techniques used in FIWA.

4

Problem formulation

The key performance indicator for wafer alignment is called overlay: this refers to the degree of misalignment between successive layers printed on a wafer. In addition to machines that process and modify wafers, semiconductor foundries have a suite of advanced tools that measure overlay. To measure how well on-machine wafer alignment managed to capture the wafer position/deformation, ASML YieldStar [5] systems can be used. YieldStar systems measure overlay targets optically, using a method similar to that employed in alignment. Note the difference between alignment and overlay measurements: whereas the TwinScan measures alignment, which is the absolute location of a layer, YieldStar systems measure overlay, which is a relative measurement between two layers. TwinScan measurements are carried out before exposure, and YieldStar measurements take place after exposure.

Another major difference is that overlay metrology samples 20 to 30 times more points than alignment. The larger set of measurements allows for a very accurate model of wafer deformation, which can also be used during production as feedback into the lithography machine. This comes with the tradeoff of taking longer to measure: this is why overlay is typically measured off-scanner, in a separate machine. Note that the overlay we measure after exposure is not directly comparable to the wafer shape model we would construct on-scanner before exposure: we have already corrected for what we have measured using the alignment marks. To construct the 'best' wafer model we could have constructed before exposure, we take the measured overlay and undo the alignment corrections giving so-called decorrected overlay.

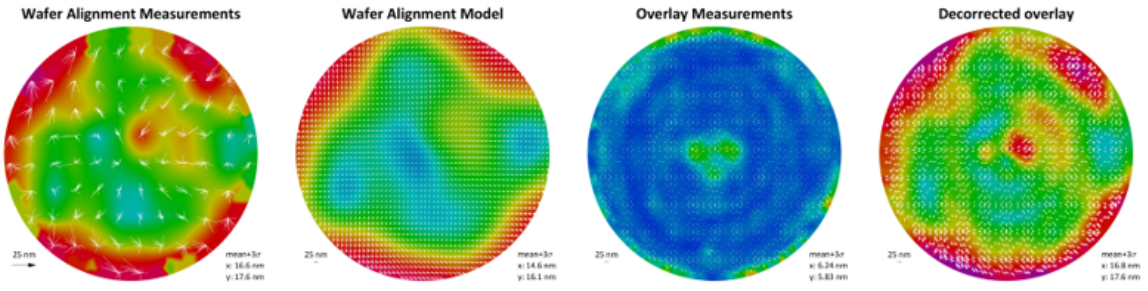

Figure 4.1: Example of wafer alignment measurements vs overlay measurements.

If we knew the decorrected overlay before exposure, we could correct for it optimally and achieve minimal overlay. The goal is thus to construct a model, using a limited number of alignment marks, that is as close as possible to the decorrected overlay when evaluating on the overlay marks.

As described in the previous section, the wafers are modeled using FIWA measurements as follows:

$$x = M_x X + \varepsilon, \tag{4.1}$$

where $x \in \mathbb{R}^{n_{\text{marks,fiwa}} \times n_{\text{wafers}}}$ contains the FIWA measurements, $X \in \mathbb{R}^{n_{\text{params}} \times n_{\text{wafers}}}$ is the weight matrix and $M_x \in \mathbb{R}^{n_{\text{marks,fiwa}} \times n_{\text{params}}}$ is the model matrix. Note that $n_{\text{marks,fiwa}}$ denotes the total number of marks in both the x- and y-directions (i.e. twice the number of mark pairs).

We aim to minimize the squared residual term ε.[1] To improve noise suppression and interpolation/extrapolation behavior, it is common to use Tikhonov regularization:

$$\min \|M_x X - x\|_F^2 + \|\Gamma X\|_F^2 \tag{4.2}$$

for a given matrix Γ. This has closed form solution

$$\bar{X} = (M_x^T M_x + \Gamma^T \Gamma)^{-1} M_x^T x, \tag{4.3}$$

or equivalently

$$\bar{X} = (M_x^T M_x + R)^{-1} M_x^T x, \tag{4.4}$$

where R is positive semidefinite.[2] We will call this R the regularization matrix.

The optimal weights $\bar{X}$, in combination with the basis functions, completely characterize our model of the wafer in FIWA. To assess how good this model is, we can evaluate it on the overlay marks used in the YieldStar measurement, and compare the modeled predictions against the actual measurements.

Similarly to the FIWA case, we can model overlay as follows. Let $y \in \mathbb{R}^{n_{\text{marks,ovl}} \times n_{\text{wafers}}}$ be the decorrected overlay, and $M_y \in \mathbb{R}^{n_{\text{marks,ovl}} \times n_{\text{params}}}$ the model matrix. Then

$$y = M_y Y + \varepsilon, \tag{4.5}$$

where $Y \in \mathbb{R}^{n_{\text{params}} \times n_{\text{wafers}}}$ is the matrix holding the weights assigned to each basis function for the different wafers. Minimizing $\|\varepsilon\|_F^2$:

$$\min \left\|M_y Y - y\right\|_F^2 \tag{4.6}$$

This admits the closed form solution

$$\bar{Y} = (M_y^T M_y)^{-1} M_y^T y. \tag{4.7}$$

Here, we do not require regularization because the system is sufficiently overdetermined.

Now we are ready to formulate the optimal regularization problem. We would like to find a regularization matrix R such that the model we fit on the FIWA measurements, evaluated on the overlay targets, is as close as possible to the actual overlay measurements. We require that $R \geq 0$ in order to have a valid penalty on the model parameters, giving the closed form solution 4.4. In fact, in addition to requiring that our regularization matrix R merely be positive semidefinite, we might impose the stronger condition that $R \geq R_{\text{min}}$ for some appropriately chosen minimum regularization matrix R_{min} to ensure that $M_x^T M_x + R$ is invertible and sufficiently well-conditioned. If $M_x^T M_x$ by itself is invertible and well-conditioned, we may choose $R_{\text{min}} = 0$. This gives the problem

$$R = \underset{R \geq R_{\text{min}}}{\text{argmin}} \quad \left\|y - M_y \bar{X}\right\|_F \tag{4.8}$$

$$= \underset{R \geq R_{\text{min}}}{\text{argmin}} \quad \left\|y - M_y (M_x^T M_x + R)^{-1} M_x^T x\right\|_F, \tag{4.9}$$

[1] A typical least squares problem would minimize $\|Ax - b\|_2$ where x, b are column vectors. For our case, the columns of y, C correspond to different wafers and we would like to minimize the sum of squared errors over the wafers. This corresponds to taking the Frobenius norm $\|A\|_F^2 = \text{Tr}(A^T A) = \sum_{i,j} a_{ij}^2$ of the matrix.

[2] This equivalence can easily be seen from $x^T \Gamma^T \Gamma x = (\Gamma x)^T \Gamma x \geq 0 \; \forall x$ in one direction, and taking the square root of R in the other direction.

i.e. find the regularization matrix R that minimizes the difference between the decorrected overlay y and the FIWA model evaluated on the overlay marks $M_y\bar{X}$.

Since $y = M_y Y + \varepsilon$, we can rewrite this as

$$R = \underset{R \geq R_{min}}{\operatorname{argmin}} \quad \left\| M_y Y - M_y (M_x^T M_x + R)^{-1} M_x^T x \right\|_F \tag{4.10}$$

The term ε drops out here because ε is noncorrectable: $M_y^T \varepsilon = 0$ such that we may take it outside the norm.[3] This is a complicated problem to solve for multiple reasons. Firstly, the optimization variable R is placed within a term that is inverted. We can solve this by setting $A = (M_x^T M_x + R)^{-1}$ and optimizing over A. The constraint $R \geq R_{min}$ is transformed as follows:

$$\begin{aligned} & R \geq R_{min} && (4.11)\\ \iff & M_x^T M_x + R \geq M_x^T M_x + R_{min} && (4.12)\\ \iff & A^{-1} \geq M_x^T M_x + R_{min} && (4.13)\\ \iff & (M_x^T M_x + R_{min})^{-1} \geq A && (4.14) \end{aligned}$$

In addition, we will have $A = (M_x^T M_x + R)^{-1} \geq 0$. Note that we are not imposing a constraint on invertibility of A here. In fact A will be singular in the general case. We will discuss this case further in Chapter 10.

Now for the constrained problem

$$\begin{aligned} A = \underset{A \geq 0}{\operatorname{argmin}} \quad & \left\| M_y Y - M_y A M_x^T x \right\|_F \\ \text{s.t.} \quad & (M_x^T M_x + R_{min})^{-1} \geq A \end{aligned} \tag{4.15}$$

both terms within the norm are left-multiplied by M_y. Because the first dimension M_y is equal to the number of overlay marks, which is typically 20-30 times greater than either the number of parameters or the number of FIWA marks, we are optimizing over a very large matrix. This is disadvantageous for computational performance reasons.

A solution to this is to decompose M_y using a QR-decomposition:

$$M_y = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1 R_1, \tag{4.16}$$

where $\begin{pmatrix} Q_1 & Q_2 \end{pmatrix}$ is an orthogonal matrix and R_1 is an upper triangular matrix. We can orthogonalize M_y in this manner and perform the same transformation on M_x (i.e. right-multiply with R_1^{-1}):

$$\begin{aligned} A' = \underset{A' \geq 0}{\operatorname{argmin}} \quad & \left\| Q_1 Y' - Q_1 A' M_x'^T x \right\|_F \\ \text{s.t.} \quad & (M_x'^T M_x' + R_{min})^{-1} \geq A', \end{aligned} \tag{4.17}$$

where $M_x' = M_x R_1^{-1}$. We have to perform the same transformation on both matrices so that the fitted coefficients still correspond to the same parameter space. Now for any matrix Q with orthogonal columns,

$$\|QA\|_F = \sqrt{\operatorname{Tr}(A^T Q^T Q A)} = \sqrt{\operatorname{Tr}(A^T A)} = \|A\|_F, \tag{4.18}$$

[3] $M_y^T \varepsilon = 0$ follows directly from the normal equation $M_y^T M_y Y = M_y^T y = M_y^T M_y Y + M_y^T \varepsilon$. This means that the other terms inside the norm, which are left-multiplied with M_y, will be orthogonal to ε.

so Q_1 drops out of the equation and we are left with

$$\begin{aligned} A' = \underset{A' \geq 0}{\operatorname{argmin}} \quad & \left\| Y' - A'(M'_x)^T x \right\|_F \\ \text{s.t.} \quad & (M'^T_x M'_x + R_{min})^{-1} \geq A', \end{aligned} \tag{4.19}$$

In summary, we have transformed the problem from overlay space to parameter space, allowing for more efficient optimization.

5

Convex Analysis

In its most general form, the problem is stated as follows:

$$\begin{aligned} A = \underset{A}{\operatorname{argmin}} \quad & \|Y - AX\|_F \\ \text{s.t.} \quad & A \succeq 0 \\ & B - A \succeq 0 \end{aligned} \tag{P}$$

where $B \succeq 0$. Call the feasible region of the problem K. Then the following lemmas hold:

Lemma 1 K *is convex.*

Proof. Let $A_1, A_2 \in K$. For $\lambda \in [0,1]$, set $A = \lambda A_1 + (1-\lambda)A_2$. Noting that

$$B - A = \lambda B - \lambda A_1 + (1-\lambda)B - (1-\lambda)A_2, \tag{5.1}$$

we see that the set $A : B - A \succeq 0$ is convex. K is then the intersection of this set with the (convex) cone of PSD matrices and is therefore convex. □

Lemma 2 K *is closed.*

Proof. We can write

$$\{A : B - A \succeq 0\} = \cap_{x \in \mathbb{R}^n} \{A \in S_+^n : \left\langle A, xx^T \right\rangle \leq \left\langle B, xx^T \right\rangle\}, \tag{5.2}$$

i.e. an intersection of closed halfspaces which is itself closed. K is the intersection of this set with the (closed) cone of PSD matrices, so that K is closed as well. □

Lemma 3 K *has nonempty interior if and only if* $B \succ 0$.

Proof. Sufficiency. Proof from contrapositive as follows. Suppose $\neg(B \succ 0)$. Since $B \succeq 0$,

$$\exists x \neq 0 : x^T B x = 0. \tag{5.3}$$

But then, for $A \in K$,

$$x^T(B - A)x = -x^T A x \geq 0. \tag{5.4}$$

But since $A \succeq 0$ implies $x^T A x \geq 0$, we must have $x^T A x = 0$. Thus A is singular and therefore not in the interior of K. This holds for any $A \in K$ so the interior of K is empty.

Necessity. Conversely, suppose $B \succ 0$. Choose

$$A := \frac{1}{2} B \succ 0. \tag{5.5}$$

Then also

$$B - A = \frac{1}{2}B \succ 0. \tag{5.6}$$

Thus we have explicitly given an element of the interior of K. □

We will always ensure the matrix B is positive definite, so in the following we may assume K has nonempty interior.

Lemma 4 *The eigenvalues of* A *are bounded below by 0 and bounded above by the largest eigenvalue* $\lambda_{\max}^{B}$ *of* B.

Proof. Bound below is immediate from $A \succeq 0$. For the upper bound, let $u_{\max}$ the (normalized) eigenvector of B corresponding to $\lambda_{\max}$ and $\mu_{\max}, v_{\max}$ the largest eigenvalue of A and its corresponding (normalized) eigenvector. For any $x \in \mathbb{R}^n$,

$$x^T(B - A)x \geq 0 \tag{5.7}$$

such that we have

$$0 \leq v_{\max}^T(B - A)v_{\max} = v_{\max}^T B v_{\max} - \mu_{\max}. \tag{5.8}$$

Since

$$v_{\max}^T B v_{\max} \leq \lambda_{\max} \,|\, u_{\max} \,|_2^2 = \lambda_{\max}, \tag{5.9}$$

we have $\lambda_{\max} - \mu_{\max} \geq 0$. □

Lemma 5 *The map* $A \mapsto \|Y - AX\|_F$ *is continuous for all* Y, X.

Proof. This follows readily from continuity of norms combined with continuity of linear maps. □

Lemma 6 *The map* $A \mapsto \|Y - AX\|_F$ *is convex for all* Y, X.

Proof. Let $A_1, A_2 \in \mathbb{R}^{n\times n}$. Now

$$\begin{aligned} \|Y - (\lambda A_1 + (1-\lambda)A_2)X\|_F &= \|\lambda Y + (1-\lambda)Y - \lambda A_1 X - (1-\lambda)A_2 X\|_F && (5.10)\\ &\leq \|\lambda Y - \lambda A_1 X\|_F + \|(1-\lambda)Y - (1-\lambda)A_2 X\|_F && (5.11)\\ &= \lambda \|Y - A_1 X\|_F + (1-\lambda)\|Y - A_1 X\|_F && (5.12) \end{aligned}$$

as required. Alternatively, this readily follows from convexity of norms and convexity of linear maps.□

An important consequence of both the objective and feasible set being convex is that every local minimizer is a global optimizer.

6

Semidefinite Programming Formulation

The presence of eigenvalue constraints suggests semidefinite programming as an appropriate approach to the problem. However, the objective is quadratic in our optimization variable. This can be resolved by rewriting the objective to show that we are, in fact, dealing with (a problem equivalent to) a semidefinite program.

6.1. Primal

Consider the problem P:

$$\begin{aligned} \min_{A} \quad & \|Y - AX\|_F \\ \text{s.t.} \quad & A \succeq 0 \\ & B - A \succeq 0. \end{aligned} \tag{6.1}$$

We can remove the matrix multiplication from the objective like so:

$$\begin{aligned} \min_{A} \quad & \|Z\|_F \\ \text{s.t.} \quad & Z = Y - AX \\ & A \succeq 0 \\ & B - A \succeq 0. \end{aligned} \tag{6.2}$$

We can transform the Frobenius norm objective by placing the matrix Z on the off-diagonal block of an expanded variable matrix [6]. Consider the problem

$$\begin{aligned} \min_{\Lambda, Z} \quad & \mathrm{Tr}(\Lambda) \\ \text{s.t.} \quad & \begin{pmatrix} I & Z \\ Z^T & \Lambda \end{pmatrix} \succeq 0. \end{aligned} \tag{6.3}$$

By Schur complement, if I is nonsingular (trivially satisfied),

$$\begin{pmatrix} I & Z \\ Z^T & \Lambda \end{pmatrix} \succeq 0 \iff I \succeq 0 \wedge \Lambda - Z^T I^{-1} Z \succeq 0. \tag{6.4}$$

Simplified this reads

$$\begin{pmatrix} I & Z \\ Z^T & \Lambda \end{pmatrix} \succeq 0 \iff \Lambda - Z^T Z \succeq 0. \tag{6.5}$$

Lemma 7 *If it exists, the optimal value ρ of problem 6.3 satisfies $\rho = \|Z\|_F^2$.*

Proof. $\rho \leq \|Z\|_F^2$. This is immediate from the fact that $\Lambda = Z^T Z$ is feasible since

$$\Lambda - Z^T Z = 0 \succeq 0. \tag{6.6}$$

The corresponding objective value is then

$$\mathrm{Tr}(Z^T Z) = \|Z\|_F^2, \tag{6.7}$$

bounding the optimum from above. $\rho \geq \|Z\|_F^2$.

Proof by contradiction. Suppose

$$\mathrm{Tr}(\Lambda) < \mathrm{Tr}(Z^T Z) = \|Z\|_F^2. \tag{6.8}$$

But then

$$\mathrm{Tr}(\Lambda - Z^T Z) < 0, \tag{6.9}$$

so that $\Lambda - Z^T Z$ has at least one negative eigenvalue. But this contradicts the constraint

$$\Lambda - Z^T Z \succeq 0, \tag{6.10}$$

which is equivalent with

$$\begin{pmatrix} I & Z \\ Z^T & \Lambda \end{pmatrix} \succeq 0. \tag{6.11}$$

□

Thus we have established that problem 6.3 minimizes $\|Z\|_F$. Going back to the original problem, we can write

$$\begin{aligned} \min_{A,\Lambda} \quad & \mathrm{Tr}(\Lambda) \\ \text{s.t.} \quad & \begin{pmatrix} I & Z \\ Z^T & \Lambda \end{pmatrix} \succeq 0 \\ & Z = Y - AX \\ & A \succeq 0 \\ & B - A \succeq 0 \end{aligned} \tag{6.12}$$

To show that this is in fact a semidefinite program, we can write it in canonical form

$$\begin{aligned} \min_{X} \quad & \langle C, X \rangle \\ \text{s.t.} \quad & \langle A_i, X \rangle = b_i, i \in [m] \\ & X \succeq 0. \end{aligned} \tag{6.13}$$

as follows:

$$\begin{aligned} \min_{A,\Lambda} \quad & \mathrm{Tr}(\Lambda) \\ \text{s.t.} \quad & \begin{pmatrix} I & Z & 0 & 0 \\ Z^T & \Lambda & 0 & 0 \\ 0 & 0 & A & 0 \\ 0 & 0 & 0 & F \end{pmatrix} \succeq 0 \\ & Z = Y - AX \\ & A + F = B \end{aligned} \tag{6.14}$$

we can enforce $Z = Y - AX$ with one equality per entry of Z:

$$Z_{ij} + (AX)_{ij} = Y_{ij} \tag{6.15}$$

or

$$Z_{ij} + \left\langle A, \widetilde{X^{ij}} \right\rangle = Y_{ij} \tag{6.16}$$

where $\widetilde{X^{ij}} = \frac{X^{ij}+(X^{ij})^T}{2}$ and X^{ij} is the matrix with the jth column of X as its ith row. We can ensure the bottom right block matrix equals $B - A$ with $\frac{n(n+1)}{2}$ constraints such that

$$A_{ij} + F_{ij} = B_{ij} \tag{6.17}$$

These, and constraints on constant elements of the matrix, can easily be written as trace inner products. The objective C becomes

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{6.18}$$

6.2. Dual

We have written the primal in (a) canonical form

$$\begin{aligned} \min_X \quad & \langle C, X \rangle \\ \text{s.t.} \quad & \langle A_i, X \rangle = b_i, i \in [m] \\ & X \succeq 0. \end{aligned} \tag{6.19}$$

which is easily related to its dual

$$\begin{aligned} \max_y \quad & b^T y \\ \text{s.t.} \quad & C - \sum_{j=1}^{m} y_j A_i \succeq 0 \\ & y \in \mathbb{R}. \end{aligned} \tag{6.20}$$

This readily gives the dual of the primal formulated above:

$$
\begin{aligned}
\max_{y} \quad & \sum_{\substack{i\in[N_{params}],j\in[N_{wafers}],\\ k=i+N_{params},l=j+N_{params}}} Y_{ij}y^1_{ij} + \sum_{i\in[N_{params}],j\in[N_{params}],j\geq i} B_{ij}y^2_{ij} + \mathrm{Tr}(\Delta) \\
\text{s.t.} \quad & \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
& - \sum_{\substack{i\in[N_{params}],j\in[N_{wafers}],\\ k=i+N_{params},l=j+N_{params}}} y^1_{ij} \begin{pmatrix} E_{kl} & 0 \\ 0 & \begin{pmatrix} \widetilde{X_{ij}} & 0 \\ 0 & 0 \end{pmatrix} \end{pmatrix} \\
& - \sum_{i\in[N_{params}],j\in[N_{params}],j\geq i} y^2_{ij} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & E_{ij} & 0 \\ 0 & 0 & 0 & E_{ij} \end{pmatrix} \\
& - \begin{pmatrix} \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
& - \begin{pmatrix} 0 & Q_1 \\ Q_1^T & \begin{pmatrix} 0 & Q_2 \\ Q_2^T & \end{pmatrix} \end{pmatrix} \succeq 0 \\
& y \in \mathbb{R}.
\end{aligned}
\tag{6.21}
$$

Simplified, this reads

$$
\begin{aligned}
\max \quad & 2\langle Y, \Gamma\rangle + \langle B, \Theta\rangle + \langle I, \Delta\rangle \\
\text{s.t.} \quad & -\begin{pmatrix} \begin{pmatrix} \Delta & \Gamma \\ \Gamma^T & -I \end{pmatrix} & Q_1 \\ Q_1^T & \begin{pmatrix} \Gamma X^T + X\Gamma^T + \Theta & Q_2 \\ Q_2^T & \Theta \end{pmatrix} \end{pmatrix} \succeq 0,
\end{aligned}
\tag{6.22}
$$

or equivalently

$$
\begin{aligned}
\max \quad & -2\langle Y, \Gamma\rangle - \langle B, \Theta\rangle - \langle I, \Delta\rangle \\
\text{s.t.} \quad & \begin{pmatrix} \begin{pmatrix} \Delta & \Gamma \\ \Gamma^T & I \end{pmatrix} & Q_1 \\ Q_1^T & \begin{pmatrix} \Gamma X^T + X\Gamma^T + \Theta & Q_2 \\ Q_2^T & \Theta \end{pmatrix} \end{pmatrix} \succeq 0,
\end{aligned}
\tag{6.23}
$$

As a basic check whether the dual is correct, we can fill in the complementary slackness condition for optimality with $p^* = d^*$:

$$
\left\langle \begin{pmatrix} I & Z & 0 & 0 \\ Z^T & \Lambda & 0 & 0 \\ 0 & 0 & A & 0 \\ 0 & 0 & 0 & F \end{pmatrix}, \begin{pmatrix} \begin{pmatrix} \Delta & \Gamma \\ \Gamma^T & I \end{pmatrix} & Q_1 \\ Q_1^T & \begin{pmatrix} \Gamma X^T + X\Gamma^T + \Theta & Q_2 \\ Q_2^T & \Theta \end{pmatrix} \end{pmatrix} \right\rangle = 0,
\tag{6.24}
$$

which simplifies to

$$
\langle I, \Delta\rangle + 2\langle Z, \Gamma\rangle + \langle \Lambda, I\rangle + \left\langle A, \Gamma X^T + X\Gamma^T + \Theta \right\rangle + \langle F, \Theta\rangle = 0.
\tag{6.25}
$$

This can in turn be simplified using $Z = Y - AX$ and $A + F = B$:

$$\langle I, \Delta \rangle + 2 \langle Y, \Gamma \rangle - 2 \langle AX, \Gamma \rangle + \langle \Lambda, I \rangle + \left\langle A, \Gamma X^T + X \Gamma^T + \Theta \right\rangle + \langle B, \Theta \rangle = 0 \tag{6.26}$$

$$\langle I, \Delta \rangle + 2 \langle Y, \Gamma \rangle + \langle \Lambda, I \rangle + \langle B, \Theta \rangle = 0, \tag{6.27}$$

which is easily observed to be the primal objective $\langle \Lambda, I \rangle$ minus the dual objective, which must then be 0 under the condition $p^* = d^*$.

6.3. Conditions for optimality

The primal bounds the dual, so it is sufficient to show strict feasibility of the dual. For the dual, let $\Delta, \Theta = I$ and $\Gamma, Q_1, Q_2 = 0$. Then the optimization variable is the identity matrix which is positive definite.

Slater's condition then tells us that the optimum of the primal is attained. This justifies the use of minima and maxima instead of infima and suprema in the preceding sections. We can also see from this why the single bounded problem does not always attain its optimum. Its dual is given by

$$\begin{aligned} \sup \quad & -2 \langle Y, \Gamma \rangle - \langle I, \Delta \rangle \\ \text{s.t.} \quad & \begin{pmatrix} \begin{pmatrix} \Delta & \Gamma \\ \Gamma^T & I \end{pmatrix} & Q \\ Q^T & \Gamma X^T + X \Gamma^T \end{pmatrix} \succeq 0. \end{aligned} \tag{6.28}$$

We cannot simply set $\Gamma = 0$ now to obtain a strictly feasible point.

Again we must have

$$\begin{pmatrix} \Delta & \Gamma \\ \Gamma^T & I \end{pmatrix} \succ 0, \tag{6.29}$$

which is equivalent with

$$\Delta - \Gamma^T \Gamma \succ 0 \tag{6.30}$$

as described above, and

$$\Gamma X^T + X \Gamma^T \succ 0. \tag{6.31}$$

In fact, finding Γ, Δ that satisfy these conditions is sufficient for a strictly feasible point since we can set $Q = 0$. The second condition is the only one which cannot be trivially satisfied, since for a given Γ we can choose $\Delta = (\lambda_{max}(\Gamma^T \Gamma) + \varepsilon) I, \varepsilon > 0$ to satisfy the first constraint. So we only need to find

$$\Gamma : \Gamma X^T + X \Gamma^T \succ 0. \tag{6.32}$$

This corresponds with X having rank N_{params}: if this were not the case, the kernel of X would have positive dimension and the above condition could not hold.

The matrix X having full rank is indeed one of the sufficient conditions given in [2] for the single-bounded problem having a solution.

Note that Slater's condition is sufficient for optimality, not necessary, but the above does give an intuition for the difference between the problems.

7

The unbounded and single bounded cases

For the generalized problem P, a natural relaxation would either be

$$\begin{aligned} \min_{A} \quad & \|Y - AX\|_F \\ \text{s.t.} \quad & A \succeq 0 \end{aligned} \tag{7.1}$$

or

$$\begin{aligned} \min_{A} \quad & \|Y - AX\|_F \\ \text{s.t.} \quad & B - A \succeq 0. \end{aligned} \tag{7.2}$$

Setting $A' = B - A$, $Y' = BX - Y$ in the second of these problems, we see that these are equivalent. The single bounded problem admits a closed form solution in some special cases, outlined by e.g. Gillis and Sharma [2]. In a different study, Jingjing et al. [7] propose a solution for the general case, which we will show to be incorrect in Section 7.3.

Relaxing the single bounded case even further, we obtain the unbounded case, known as the symmetric procrustes problem:

$$\begin{aligned} \min_{A} \quad & \|Y - AX\|_F \\ \text{s.t.} \quad & A = A^T. \end{aligned} \tag{7.3}$$

7.1. The symmetric/unbounded problem

This case always admits a closed form solution based on the singular value decomposition [1]. In Higham [1], a result is given only for tall matrices $m \leq n$. We prove the general case below. Note that the roles of X and A are reversed with respect to the paper.

First, we reformulate the problem as follows. We can write X as its singular value decomposition:

$$X = U \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} V^T. \tag{7.4}$$

Using this, we can rewrite the optimization problem as

$$\begin{aligned} \min_{A} \quad & \left\| U^T Y V - U^T A U \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} V^T V \right\|_F \\ \text{s.t.} \quad & A = A^T. \end{aligned} \tag{7.5}$$

Writing $A' = U^T AU$, $Y' = U^T YV$, we can simplify as follows:

$$\begin{aligned} \min_{A'} \quad & \left\| Y' - A' \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \right\|_F \\ \text{s.t.} \quad & A' = A'^T. \end{aligned} \tag{7.6}$$

Now writing $A' = \begin{pmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{pmatrix}$, $Y' = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix}$ according to the decomposition of X, we obtain

$$\begin{aligned} \min_{A_{11}, A_{12}} \quad & \|Y_1 - A_{11}\Sigma_1\|_F + \left\| Y_2 - A_{12}^T \Sigma_1 \right\|_F \\ \text{s.t.} \quad & A_{11} = A_{11}^T. \end{aligned} \tag{7.7}$$

Contrary to the (double) bounded case, A_{11} and A_{12} are independent from each other and can be optimized separately. Note that there is no constraint on the elements of A_{12} so we have the unconstrained solution

$$A_{12}^T = Y_2 \Sigma_1^{-1}. \tag{7.8}$$

We can expand the term containing A_{11}:

$$\|Y_1 - A_{11}\Sigma_1\|_F^2 = \sum_i (a_{ii}\sigma_i - Y_{ii})^2 + \sum_{j>i} ((\sigma_i A_{ij} - Y_{ij})^2 + (\sigma_i A_{ij} - Y_{ji})^2), \tag{7.9}$$

where we directly incorporated the symmetry $A_{ij} = A_{ji}$. This poses an easy optimization problem where all the A_{ij} can be optimized separately by taking the derivative of the expression with respect to each. This yields

$$a_{ij} = \frac{\sigma_i Y_{ij} + \sigma_j Y_{ji}}{\sigma_i^2 + \sigma_j^2}. \tag{7.10}$$

Writing the matrix $\Phi = (\varphi_{ij})$, $\varphi_{ij} = \frac{1}{\sigma_i^2 + \sigma_j^2}$, we can write this concisely as

$$A_{11} = \Phi * (Y_1\Sigma + \Sigma Y_1^T), \tag{7.11}$$

where $*$ denotes the Hadamard product. Finally, we can 'reassemble' the matrix A:

$$A = U \begin{pmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{pmatrix} U^T \tag{7.12}$$

where A_{22} is arbitrary and can be chosen as the zero matrix.

7.2. The single-bounded problem

The same reformulation as in the previous section can be used, but now with the constraint $A \succeq 0$ instead of $A = A^T$: Noting that unitary transformations preserve positive semidefiniteness[1], the problem reads

$$\begin{aligned} \min_{A_{11}, A_{12}} \quad & \|Y_1 - A_{11}\Sigma_1\|_F + \left\| Y_2 - A_{12}^T \Sigma_1 \right\|_F \\ \text{s.t.} \quad & A' \succeq 0. \end{aligned} \tag{7.13}$$

We can write the constraint using the generalized Schur complement [8]

Lemma 8 ([8] Theorem 1) *A block matrix* $\begin{pmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{pmatrix}$ *is positive semidefinite if and only if:*

[1] This follows directly from $x^T Ax \geq 0 \iff (Ux)^T A(Ux) \geq 0 \iff x^T(U^T AU)x \geq 0 \forall x$.

1. $A_{11} \succeq 0$
2. $\ker(A_{11}) \subseteq \ker(A_{12}^T)$
3. $A_{22} - A_{12}^T A_{11}^+ A_{12} \succeq 0$

where + denotes the Moore-Penrose pseudoinverse.

This gives the following formulation:

$$\begin{aligned} \min_{A_{11},A_{12}} \quad & \|Y_1 - A_{11}\Sigma_1\|_F + \left\|Y_2 - A_{12}^T\Sigma_1\right\|_F \\ \text{s.t.} \quad & A_{11} \succeq 0 \\ & \ker(A_{11}) \subseteq \ker(A_{12}^T) \\ & A_{22} - A_{12}^T A_{11}^+ A_{12} \succeq 0 \end{aligned} \tag{7.14}$$

Note that the third constraint is redundant since A_{22} does not appear in the objective and thus we can always choose $A_{22} = A_{12}^T A_{11}^+ A_{12}$. Obviously a solution $\{A_{11}, A_{12}\}$ obtained by optimizing each objective separately, subject to the constraint $A_{11} \succeq 0$ for the first problem, will provide a lower bound to the objective value. Literature on this subject [9] shows that this is actually the highest such lower bound, and therefore the infimum. The infimum of the full problem is attained, then, if and only if the kernel constraint holds for these matrices.

As before, the optimization problem over A_{12} is unconstrained and can be easily solved. The problem for A_{11} is basically a version of the original problem where the matrix X is full rank, diagonal and positive definite. These subproblems both have unique optima [2] but the solution to the full problem, if it exists, may not be unique (i.e. if X is not full rank). In the literature, many authors have given the solution to the general problem (the (possibly) rank-deficient case), contingent on having a solution for the full-rank subproblem. Several numerical schemes were proposed for this [2], but the first paper that claimed to have a closed-form solution only came out in 2019 [7].

7.3. An incorrect solution

Here, the authors propose the following method of solution. Decompose X as

$$X = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} \tag{7.15}$$

Then, let

$$\hat{S} = \Phi * (U_1^T Y V_1 \Sigma + \Sigma V_1^T Y^T U_1). \tag{7.16}$$

This is the unconstrained solution as presented before. Now if the eigendecomposition of $\hat{S}$ reads

$$\hat{S} = N \begin{pmatrix} \delta_1 & & & \\ & \delta_2 & & \\ & & \ddots & \\ & & & \delta_r \end{pmatrix} N^T, \tag{7.17}$$

the authors claim

$$A_{11} = N \begin{pmatrix} \delta_1 & & & \\ & \delta_2 & & \\ & & \ddots & \\ & & & \delta_r \end{pmatrix}_+ N^T, \tag{7.18}$$

where + denotes taking the positive part of the matrix elements, is the solution to the full-rank subproblem. This would mean that the solution to the full-rank positive semidefinite/single-bounded

problem is effectively the solution of the symmetric problem, projected onto S_+^n. This is indeed the intuitive answer, but unfortunately relies on a mistaken line of reasoning.

Jingjing et al. [7] use the following approach to the full-rank subproblem. First, the problem is reduced to the case where X is diagonal by unitarily transforming according to the SVD of X.

$$\begin{aligned} \min_A \quad & \|Y - AD\|_F \\ \text{s.t.} \quad & A \succeq 0. \end{aligned} \tag{7.19}$$

The diagonal matrix D contains the singular values of (full rank) X, so that D is positive definite. The authors then show that the above problem is equivalent to

$$\begin{aligned} \min_A \quad & \|(Y_1 - A)D\|_F \\ \text{s.t.} \quad & A \succeq 0, \end{aligned} \tag{7.20}$$

where

$$Y_1 = \Phi * (YD + DY^T) \tag{7.21}$$

with $\Phi = (\varphi_{ij}), \varphi_{ij} = \frac{1}{\sigma_i^2 + \sigma_j^2}$ as before. Notice that Y_1 is exactly the solution to the symmetric/unbounded problem. Then, Y_1 is decomposed as the difference of two positive semidefinite matrices $Y_1 = \bar{Y_1} - \bar{Y_2}$:

$$Y_1 = W \begin{pmatrix} \gamma_1 & & & \\ & \gamma_2 & & \\ & & \ddots & \\ & & & \gamma_n \end{pmatrix} W^T = W \begin{pmatrix} \gamma_1 & & & \\ & \gamma_2 & & \\ & & \ddots & \\ & & & \gamma_n \end{pmatrix}_+ W^T - W \begin{pmatrix} -\gamma_1 & & & \\ & -\gamma_2 & & \\ & & \ddots & \\ & & & -\gamma_n \end{pmatrix}_+ W^T. \tag{7.22}$$

Notice that $\bar{Y_1}, \bar{Y_2}$ are orthogonal under the trace inner product. Filling in in 7.20:

$$\|(Y_1 - A)D\|_F^2 = \left\|(\bar{Y_1} - A)D\right\|_F^2 - 2\langle(\bar{Y_1} - A)D, \bar{Y_2}D\rangle + \left\|\bar{Y_2}D\right\|_F^2 \tag{7.23}$$

$$= \left\|(\bar{Y_1} - A)D\right\|_F^2 + 2\mathrm{tr}(D\bar{Y_2}^T AD) + \left\|\bar{Y_2}D\right\|_F^2. \tag{7.24}$$

The third term is constant and the first term is obviously minimized for $A = \bar{Y_1}$. The authors then make the following mistake: from 'noting that D is a positive definite diagonal matrix', they conclude that $A = \bar{Y_1}$ is the minimizer of the expression. No explanation is given, but supposedly the authors assume the middle term cannot be negative. Indeed in that case,

$$2\mathrm{tr}(D\bar{Y_2}^T\bar{Y_1}D) = 0 \tag{7.25}$$

so that $\bar{Y_1}$ would be the optimal solution. This is not the case however, as the trace of the product of more than two positive semidefinite matrices may very well be negative.

Going back to the subproblem where the mistake originated, we want to find $X \succeq 0$ that minimizes $\|(F - X)D\|_F$, where $F = F^T$ and D is diagonal positive definite. The problem here is that FD and XD may not be symmetric: otherwise we could take for X the projection of F onto S_+^n. Writing $A := F - X$ and noticing that we can equivalently minimize the square norm, we can rewrite the objective:

$$\min_X \|AD\|_F^2. \tag{7.26}$$

Now we can rewrite this objective using the identity derived below:

$$\|AD - DA\|_F^2 = \mathrm{tr}((AD - DA)^T(DA - AD)) \tag{7.27}$$

$$= \|AD\|_F^2 + \|DA\|_F - \mathrm{tr}(ADAD) - \mathrm{tr}(DADA). \tag{7.28}$$

$DA = (AD)^T$ so the norms are equal, and we can rewrite the traces with the cyclic property:

$$= 2\|AD\|_F^2 - 2\mathrm{tr}(D^{\frac{1}{2}}ADAD^{\frac{1}{2}}) \tag{7.29}$$

$$= 2\|AD\|_F^2 - 2\left\|D^{\frac{1}{2}}AD^{\frac{1}{2}}\right\|_F^2. \tag{7.30}$$

By rewriting this result we can conclude

$$\|AD\|_F^2 = \frac{1}{2}\|AD - DA\|_F^2 + \left\|D^{\frac{1}{2}}AD^{\frac{1}{2}}\right\|_F^2. \tag{7.31}$$

Replacing $A := F - X$ again gives

$$\|(F - X)D\|_F^2 = \frac{1}{2}\|(F - X)D - D(F - X)\|_F^2 + \left\|D^{\frac{1}{2}}(F - X)D^{\frac{1}{2}}\right\|_F^2. \tag{7.32}$$

We see here that, if $(F - X)D$ is symmetric, the equation simplifies to

$$\|(F - X)D\|_F^2 = \left\|D^{\frac{1}{2}}(F - X)D^{\frac{1}{2}}\right\|_F^2 \tag{7.33}$$

which is minimized by $D^{-\frac{1}{2}}\mathcal{P}_{S_+}(D^{\frac{1}{2}}FD^{\frac{1}{2}})D^{-\frac{1}{2}}$. However, this fails to hold in general. This solution already provides a better approximation than the 'unweighted' projection proposed by the authors, but becomes more inaccurate as the antisymmetric term becomes larger. Intuitively, the more dispersed the singular values of X are, the further D will be from a scalar multiple of the identity matrix, and the larger the antisymmetric term will be.

8

Projection methods

One way to view the problem is as a projection onto a convex set. If we define the sets

$$V = \{AX : A \succeq 0\} \tag{8.1}$$

and

$$W = \{AX : B - A \succeq 0\}, \tag{8.2}$$

the problem

$$\begin{aligned} \min_{A} \quad & \|Y - AX\|_F \\ \text{s.t.} \quad & A \succeq 0 \\ & B - A \succeq 0 \end{aligned} \tag{8.3}$$

can be viewed as a projection of Y onto the intersection of the two convex sets V and W:

$$\begin{aligned} \min_{A} \quad & \|Y - (AX)\|_F \\ & AX \in V \cap W. \end{aligned} \tag{8.4}$$

This restatement of the problem allows us to take advantage of existing methods for such problems, some of which are detailed in [10]. The simplest form of a projection method, simply called projection onto convex sets (POCS), alternatingly projects the initial point y_0 onto V and W, converging to a point in the intersection $V \cap W$ [11]. However, this point is not necessarily the projection of y_0 onto $V \cap W$.

Other methods, such as Dykstra's projection algorithm or ADMM [12], are known to converge to the projection of the initial point. To use these methods effectively, we require a fast (preferably closed-form) projection function for each of the sets V and W. Because the structure of the two sets making up our feasible region are essentially the same, we really only require one projection function which is the single bounded Procrustes problem. The problem here is that, while the double bounded problem is guaranteed to attain its infimum, the single bounded problem might not.

As detailed in e.g. Woodgate [9], a sufficient condition for attaining the infimum in the single bounded problem is that X is full rank. This is not the case in general: indeed in our setting, $X \in \mathbb{R}^{n_{\text{params}} \times n_{\text{wafers}}}$ so that X is always rank deficient if we have more parameters than wafers.

A simple modification to the problem is

$$\begin{aligned} \min_{A} \quad & \left\| \begin{pmatrix} Y & 0 \end{pmatrix} - A \begin{pmatrix} X & \lambda I \end{pmatrix} \right\|_F \\ \text{s.t.} \quad & A \succeq 0 \\ & B - A \succeq 0. \end{aligned} \tag{8.5}$$

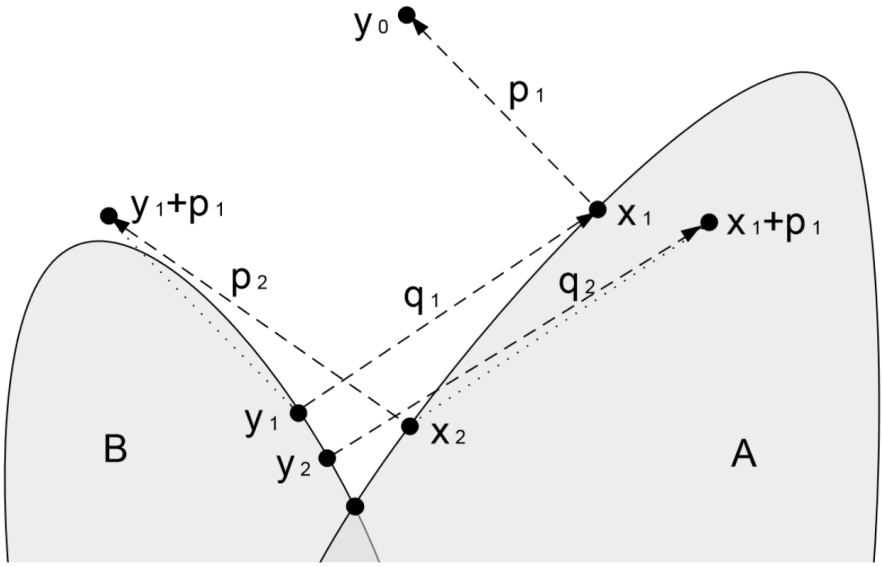

Figure 8.1: Illustration of Dykstra's algorithm for finding the projection onto the intersection of two convex sets. Source: [10]

Squaring the objective, we can separate the left and right parts:

$$\begin{aligned} \min_{A} \quad & \|Y - AX\|_F^2 + \lambda^2 \|A\|_F^2 \\ \text{s.t.} \quad & A \geq 0 \\ & B - A \geq 0. \end{aligned} \tag{8.6}$$

We have essentially added a 'regularization term' to the problem, penalizing the norm of A. This ensures that the matrix $\begin{pmatrix} X & \lambda I \end{pmatrix}$ is full rank and that the projection of $\begin{pmatrix} Y & 0 \end{pmatrix}$ onto V and W respectively exists.

The projection method we will use is Dykstra's projection algorithm, because it is simple, converges well and does not require a choice of parameters. The convergence was found to be comparable to ADMM both in [10] and specifically for our problem setting.

Algorithm 1 Dykstra's projection algorithm [10]

Require: $Y \in \mathbb{R}^{n \times m}, X \in \mathbb{R}^{n \times m} \lambda \geq 0$
Ensure: $AX \approx \mathcal{P}_{V \cap W}(Y)$
  $Y \leftarrow \begin{pmatrix} Y & 0 \end{pmatrix}$
  $X \leftarrow \begin{pmatrix} X & \lambda I \end{pmatrix}$
  $P \leftarrow \mathbf{0}$
  $Q \leftarrow \mathbf{0}$
  while not converged **do**
    $AX \leftarrow \mathcal{P}_V(Y + P)$
    $P \leftarrow P + Y - AX$
    $Y \leftarrow \mathcal{P}_W(AX + Q)$
    $Q \leftarrow Q + AX - Y$
  end while

For the projection steps $\mathcal{P}_V$ and $\mathcal{P}_W$ we can use, e.g. the fast gradient method solution by Gillis and Sharma [2]. We can use that

$$\begin{aligned} \min_{A} \quad & \|Y - AX\|_F \\ \text{s.t.} \quad & B - A \geq 0 \end{aligned} \tag{8.7}$$

is equivalent to

$$\begin{aligned} B - \min_{C} \quad & \|(BX - Y) - CX\|_F \\ \text{s.t.} \quad & C \geq 0 \end{aligned} \tag{8.8}$$

to solve both projections in a similar manner.

9

Projected gradient methods

9.1. A basic scheme

Projected gradient methods are a well-studied approach to solving constrained optimization problems [13]. For our problem P, a projected gradient method can be applied as follows. First, a gradient step is taken, after which the (possibly infeasible) result is projected onto the feasible region. For the specific problem at hand we can easily compute

$$\begin{aligned} \nabla_A f(A) &= \nabla_A \|Y - AX\|_F^2 && (9.1)\\ &= \nabla_A \mathrm{Tr}((Y - AX)^T(Y - AX)) && (9.2)\\ &= 2(Y - AX)X^T, && (9.3) \end{aligned}$$

where we used the squared objective $f(A) = \|Y - AX\|_F^2$. After a gradient step $A'_{k+1} = A_k - t_k \nabla f(A)$, the new iterate A'_{k+1} is not guaranteed to be positive semidefinite or even symmetric. Positive semidefiniteness is hard to enforce directly, but symmetry can be ensured quite easily by symmetrizing the gradient:

$$A'_{k+1} = A_k - t_k \frac{\nabla f(A) + \nabla f(A)^T}{2}. \tag{9.4}$$

Now we can obtain the next feasible iterate by projecting onto the feasible region:

$$A_{k+1} = \mathcal{P}_K(A'_{k+1}), \tag{9.5}$$

where

$$K = \{A \in S_+^n : B - A \succeq 0\}. \tag{9.6}$$

It is natural to view K as the intersection of two convex sets

$$K = \{A \succeq 0\} \cap \{A : B - A \succeq 0\} = K_1 \cap K_2. \tag{9.7}$$

The projection onto each of these sets is simple: projecting on the first set entails truncating all negative eigenvalues to 0, and projecting on the second set simply requires truncating all negative eigenvalues of $B - A$, and subtracting the result from B to get back A. We might iteratively project onto the intersection of these sets using Dykstra's algorithm.

9.2. Improving on the basic scheme

Another approach, however, avoids the need for iterative schemes at this level altogether: noting that B is positive definite, we can write

$$B = UDU^T = UD^{\frac{1}{2}}D^{\frac{1}{2}}U^T. \tag{9.8}$$

Note that D is diagonal and has strictly positive diagonal entries, since B is positive definite. We can transform the problem so that B is the identity matrix using the matrix $Q := UD^{\frac{1}{2}}$:

$$A' = Q^{-1}A(Q^{-1})^T \tag{9.9}$$
$$Y' = Q^{-1}Y \tag{9.10}$$
$$X' = Q^T X, \tag{9.11}$$

Note that this transformation is essentially a change of basis, so this leaves positive semidefiniteness intact: $A' \succeq 0 \iff A \succeq 0$. Furthermore

$$B - A \succeq 0 \tag{9.12}$$
$$\iff QQ^T - QA'Q^T \succeq 0 \tag{9.13}$$
$$\iff I - A' \succeq 0. \tag{9.14}$$

Now $I \succeq A \succeq 0$ precisely when the eigenvalues of A lie in the closed interval $[0, 1]$. This suggests that we can produce the projection of A' onto the set $\{A' \succeq 0 : I - A' \succeq 0\}$ by truncating the eigenvalues to this interval. To show this is true, consider the following:

$$\min_{A^* \in K} \quad \|A^* - A'\|_F . \tag{9.15}$$

Writing $A' = N\Sigma N^T$,

$$\min_{A^* \in K} \quad \left\|A^* - N\Sigma N^T\right\|_F \tag{9.16}$$
$$\min_{A^* \in K} \quad \left\|N^T A^* N - \Sigma\right\|_F \tag{9.17}$$

Since this is essentially an elementwise optimalization, it is evident that the optimal solution to this problem is $N^T A^* N = \Sigma_{[0,1]}$, the diagonal matrix with the ith diagonal element being $\max\{0, \min\{1, \sigma_i\}\}$. So then $A^* = N\Sigma_{[0,1]}N^T$.

Under this transformation, the objective reads

$$\|Y - AX\|_F = \left\|QY' - QA'Q^T Q^{-T} X'\right\|_F \tag{9.18}$$
$$= \|QY' - QA'X'\|_F \tag{9.19}$$
$$= \left\|VD^{\frac{1}{2}}Y' - VD^{\frac{1}{2}}A'X'\right\|_F \tag{9.20}$$
$$= \left\|D^{\frac{1}{2}}Y' - D^{\frac{1}{2}}A'X'\right\|_F , \tag{9.21}$$

giving the new gradient

$$\nabla_{A'} f'(A') = \nabla_{A'} \left\|D^{\frac{1}{2}}(Y' - A'X')\right\|_F^2 \tag{9.22}$$
$$= \nabla_{A'} \mathrm{Tr}((Y' - A'X')^T (Y' - A'X')) \tag{9.23}$$
$$= \nabla_{A'} \mathrm{Tr}((Y' - A'X')^T D(Y' - A'X')) \tag{9.24}$$
$$= 2D(Y - AX)X^T. \tag{9.25}$$

Now for a projected gradient method, we may combine the projection and gradient steps in different ways. One approach is described above, taking a gradient step and directly projecting on the feasible set:

$$A_{k+1} = \mathcal{P}_K(A_k - t_k \frac{\nabla f(A_k) + \nabla f(A_k)^T}{2}), \tag{9.26}$$

where t_k is a (possibly time-varying) step size.

Notice that the search direction controlled by t_k may exit the feasible region: only after projecting are we assured of a feasible iterate. Another possible scheme is the following:

$$d_{k+1} = \mathcal{P}_K(A_k - t_k \frac{\nabla f(A_k) + \nabla f(A_k)^T}{2}) - A_k \tag{9.27}$$

$$A_{k+1} = A_k + t_{k+1} d_{k+1}. \tag{9.28}$$

Here, we project the search direction onto the feasible set first. Notice that, as long as $t_k \in [0, 1]$, A_{k+1} will be feasible since K is convex. For $t_k = 1$, this reduces to the previous scheme, so we assume without loss of generality that $0 < t_k < 1$ for all k. This guarantees that every iterate is strictly feasible, making this an inner point method.

These two methods have very different convergence properties. For example, suppose the minimizer is on the boundary of the feasible set and at that point, the (symmetrized) gradient is perpendicular to the boundary. Then, the first exterior point method can 'bounce' around the minimum and diverge, or converge very slowly.

9.3. Convergence of the gradient method

Lipschitz continuity is an important property in the study of (projected) gradient methods. A function g is L-Lipschitz continuous if $\forall x, y$ in its domain:

$$\left\|g(x) - g(y)\right\| \leq L \left\|x - y\right\|. \tag{9.29}$$

For our gradient, the Lipschitz constant can be derived as follows. Writing $y = A$, $x = A + H$, we require L to have the property

$$\|\nabla f(A + H) - \nabla f(A)\| \leq L \|A + H - A\| = L \|H\|. \tag{9.30}$$

Filling in the (symmetrized) gradient $\nabla f(A) = D(Y - AX)X^T + X(Y - AX)^T D$:

$$\|\nabla f(A + H) - \nabla f(A)\| = \left\|D(Y - (A + H)X)X^T + X(Y - (A + H)X)^T D - D(Y - AX)X^T - X(Y - AX)^T D\right\| \tag{9.31}$$

$$= \left\|D(A + H)XX^T + X((A + H)X)^T D - DAXX^T - X(AX)^T D\right\| \tag{9.32}$$

$$= \left\|DHXX^T + XHX^T D\right\| \tag{9.33}$$

$$= \left\|DHXX^T + (DHXX^T)^T\right\| \tag{9.34}$$

$$\leq 2\left\|DHXX^T\right\| \tag{9.35}$$

$$\leq 2\|D\|_2 \left\|XX^T\right\|_2 \|H\|, \tag{9.36}$$

So that $L = 2\|D\|_2 \left\|XX^T\right\|_2 = 2\|D\|_2 \|X\|_2^2$. It should be noted that this is an upper bound, since

$$\left\|DHXX^T + (DHXX^T)^T\right\| \leq 2\left\|DHXX^T\right\| \tag{9.37}$$

only holds with equality when $DHXX^T$ is symmetric.

A key consequence of this is the smoothness condition [14] which reads

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\left\|y - x\right\|^2. \tag{9.38}$$

Now consider taking gradient steps with size α according to equation 9.26. To combine this with the projection step after the gradient step, we use the fact that projection onto a convex set is a nonexpansive operation. For a gradient step $\tilde{x}^{k+1} = x^k - \alpha \nabla f(x)$, the next iterate is $x^{k+1} = \mathcal{P}_K(\tilde{x}^{k+1})$.

Now projections onto convex sets always have the property that, for all $y \in K$:

$$\langle \tilde{x}^{k+1} - x^{k+1}, y - x^{k+1} \rangle \leq 0. \tag{9.39}$$

Setting $y = x^k$ gives

$$\langle \tilde{x}^{k+1} - x^{k+1}, x^k - x^{k+1} \rangle \leq 0 \tag{9.40}$$

$$\langle x^k - \alpha \nabla f(x^k) - x^{k+1}, x^k - x^{k+1} \rangle \leq 0 \tag{9.41}$$

$$\left\| x^k - x^{k+1} \right\|^2 \leq \alpha \langle \nabla f(x^k), x^k - x^{k+1} \rangle \tag{9.42}$$

$$\frac{1}{\alpha} \left\| x^k - x^{k+1} \right\|^2 \leq \langle \nabla f(x^k), x^k - x^{k+1} \rangle. \tag{9.43}$$

. Now using the smoothness inequality with $x = x^k$ and $y = x^{k+1}$,

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L}{2} \left\| x^{k+1} - x^k \right\|^2 \tag{9.44}$$

$$f(x^{k+1}) \leq f(x^k) - \langle \nabla f(x^k), x^k - x^{k+1} \rangle + \frac{L}{2} \left\| x^{k+1} - x^k \right\|^2 \tag{9.45}$$

$$\tag{9.46}$$

Substituting the bound found previously:

$$f(x^{k+1}) \leq f(x^k) - \frac{1}{\alpha} \left\| x^k - x^{k+1} \right\|^2 + \frac{L}{2} \left\| x^{k+1} - x^k \right\|^2 \tag{9.47}$$

$$f(x^{k+1}) \leq f(x^k) - (\frac{1}{\alpha} - \frac{L}{2}) \left\| x^{k+1} - x^k \right\|^2. \tag{9.48}$$

This already shows the familiar result that the objective value is guaranteed to decrease at each iteration if $\alpha < \frac{2}{L}$. In practice this is a conservative step size for many problems. We can use nonexpansiveness of projections on closed convex sets [15] to derive a bound that is not dependent on x^{k+1} itself. Now

$$\tilde{x}^{k+1} = x^k - \alpha \nabla f(x^k) \tag{9.49}$$

$$x^k - \tilde{x}^{k+1} = \alpha \nabla f(x^k) \tag{9.50}$$

$$\tag{9.51}$$

$$\left\| x^k - x^{k+1} \right\| = \left\| \mathcal{P}_K(x^k) - \mathcal{P}_K(\tilde{x}^{k+1}) \right\| \leq \left\| x^k - \tilde{x}^{k+1} \right\| = \alpha \left\| \nabla f(x^k) \right\| \tag{9.52}$$

where we used nonexpansiveness of projections and the fact that x^k is feasible and thus equals its projection. Plugging this into the previous bound on the function value decrease gives

$$f(x^{k+1}) \leq f(x^k) - \frac{L}{2} \alpha^2 \left\| \nabla f(x^k) \right\|^2 \tag{9.53}$$

$$f(x^{k+1}) \leq f(x^k) - \frac{\alpha L}{2} \left\| \nabla f(x^k) \right\|^2, \tag{9.54}$$

given that $\alpha < \frac{2}{L}$.

10

Implementation

The semidefinite programming method from Chapter 6 was implemented in Julia using JuMP [16] in combination with the solvers CSDP [17], Hypatia [18], ProxSDP [19], SCS [20] and COSMO [21]. Of these solvers, only COSMO managed to solve the problem in under 1 minute. An added benefit is that COSMO natively supports SDPs with a quadratic objective, so we do not have to use the reformulation shown before.

The alternating projection method from Chapter 8 and the projected gradient method from 9 were implemented in Julia without any special dependencies.

The 'pipeline' from alignment and overlay data to the optimal regularization matrix is now as follows:

- The data is preprocessed. This includes removing invalid marks and removing the average fingerprint.
- A model matrix is generated from the model specification and alignment mark positions. Separate model matrices are generated for alignment and overlay.
- The model matrices are transformed based on the orthogonalization of M_y.
- The matrices X and Y are constructed from the model matrices and position deviations for alignment and overlay respectively.
- The upper bound B is calculated from M_x and possibly a minimum regularization matrix.
- The data is scaled appropriately, and split in the x- and y-directions if the problem allows this.
- The semidefinite program is solved, giving optimal matrix A.
- In general, A is not invertible so the eigendecomposition $A = S\Sigma S^T$ is used.
- $R_{opt} = \Sigma^{-1} - S^T M_x^T M_x S$ and the corresponding subspace matrix S is returned.

These last two steps address the problem of having a singular A such that

$$A = (M_x^T M_x + R_{opt})^{-1}. \tag{10.1}$$

If we take the eigendecomposition of A, we will have

$$A = \begin{pmatrix} S_0 & S \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & \Sigma \end{pmatrix} \begin{pmatrix} S_0^T \\ S^T \end{pmatrix} = S\Sigma S^T. \tag{10.2}$$

To be precise, we will see very small but nonzero eigenvalues in the eigendecomposition of A. Even very small negative values may appear due to the finite infeasibility tolerance of the solver. To handle this, we consider everything below a certain cutoff (e.g. 10^{-8}) to be a zero value, and calculate the subspace matrix accordingly.

We thus use the pseudoinverse $A^+ = S\Sigma^{-1}S^T$:

$$\Sigma^{-1} = S^T(M^TM + R_{opt})S, \tag{10.3}$$

allowing for extraction of the optimal regularization matrix R_{opt}.

The subspace associated with this matrix can be viewed as a model reduction, reducing the amount of parameters. This subspace matrix must then be right-multiplied with the model matrix when using the fitted regularization matrix.

11

Results

In this section, the improvements in overlay resulting from optimal regularization will be demonstrated for different customer datasets. A dataset consists of a number of wafers, in which each wafer has the same number and layout of alignment and overlay marks. For a given dataset, we optimize the regularization matrix with the gradient method described in Chapter 9 and use this regularization matrix, together with measurements of the alignment marks, to construct a model of wafer deformation. In the scanner, this model would be used to correct optimally for the shape of the wafer when exposing the next layer. After this, the wafer is measured with a much more dense layout of overlay targets. This determines the overlay: mismatch between pattern layers on the wafer. The lower overlay is, the more accurately we have constructed a model of wafer deformation from the alignment marks.
To summarize overlay in one statistic, mean-3-sigma (m3s/mean + 3σ) is often used, i.e. the mean of the position deviation measurements plus three times the standard deviation. This is reported separately for the x- and y-directions.
The default regularization that will be compared against is the bending energy regularization. Bending energy for a model function u is defined as follows:

$$E_b(u) = \int_{\text{wafer}} (\frac{\partial^2 u}{\partial x^2})^2 + 2(\frac{\partial^2 u}{\partial x \partial y})^2 + (\frac{\partial^2 u}{\partial y^2})^2 \, dxdy, \tag{11.1}$$

with a corresponding inner product[1]

$$\langle u, v \rangle_b = \int_{\text{wafer}} \frac{\partial^2 u}{\partial x^2}\frac{\partial^2 v}{\partial x^2} + 2\frac{\partial^2 u}{\partial x \partial y}\frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2}\frac{\partial^2 v}{\partial y^2} \, dxdy. \tag{11.2}$$

We can construct the covariance matrix of the model functions p_i under this inner product:

$$[M_{cov}]_{ij} = \left\langle p_i, p_j \right\rangle. \tag{11.3}$$

Now $R_{bending} = M_{cov}$ is used as a regularization matrix by multiplying it with a prefactor: $R = \lambda R_{bending}$. The best value of λ for a given dataset is found by line search.

A typical waferplot, showing decorrected overlay, is given in Figure 11.1. As mentioned before, decorrected overlay is obtained by measuring overlay and undoing the FIWA alignment corrections. [2]

[1] Specifically, this form is only an inner product on the space of functions on the wafer that are twice differentiable and have a nonzero second derivative, i.e. i.e. excluding constant and linear model functions. These components would be nonzero on the wafer but have zero bending energy, such that the form has a nontrivial kernel and thus fails to be an inner product. In practice, we already fit these components with a 4PAR/6PAR model, which does not need to be regularized.

[2] More precisely, what is shown in figure 11.1 is overlay after applying only a 4PAR model. This model has four parameters and fits translation, symmetric magnification and symmetric rotation.

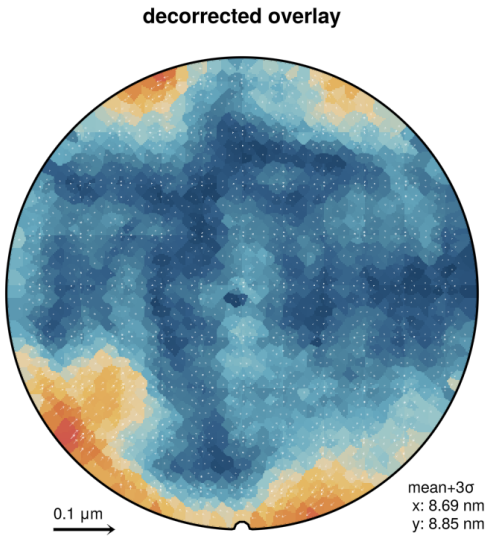

Figure 11.1: A sample waferplot. Data from Bonding.

At the bottom of the wafer we see a small notch depicted: this notch is actually present on the wafer (although not at the same scale). The notch is used as a reference point for orientation and alignment of the wafer in various lithography processes. Wafers are always plotted with the notch at the bottom.
A multitude of arrows are plotted at locations corresponding to the alignment/overlay marks on the wafer. The x-component and y-component of each arrow indicate the deformations in those respective directions. An arrow at the bottom is given for scale. A colorscale provides added visualization, where blue represents low deformation and red represents high deformation. Below, we will mostly show stacked waferplots, which show arrows corresponding to all wafers in the dataset plotted on top of each other. The colormap and m3s statistic are derived from an average over all the wafers.

To safeguard customers' interests, datasets will be referred to by generic names such as 'Foundry' or 'Bonding'. [3]

[3]'Foundry' (also called 'fab') refers to a factory where integrated circuits are manufactured. 'Bonding' refers to wafer bonding, in which two or more wafers are attached to each other through chemical and mechanical processes.

11.1. Foundry

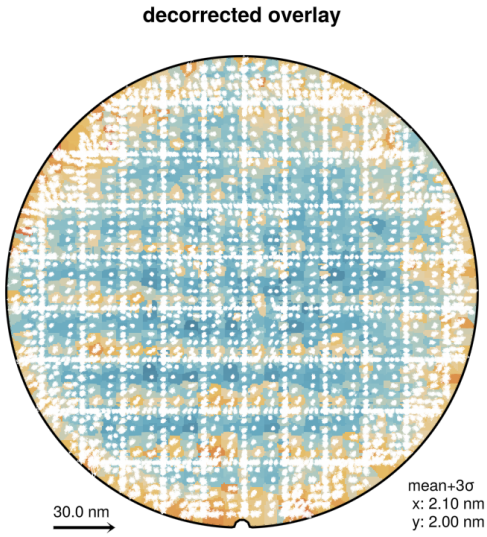

Figure 11.2: Decorrected overlay for Foundry.

For this first set of figures, optimal regularization was trained on all wafers using the gradient method in Chapter 9 and then used for a least-squares fit:

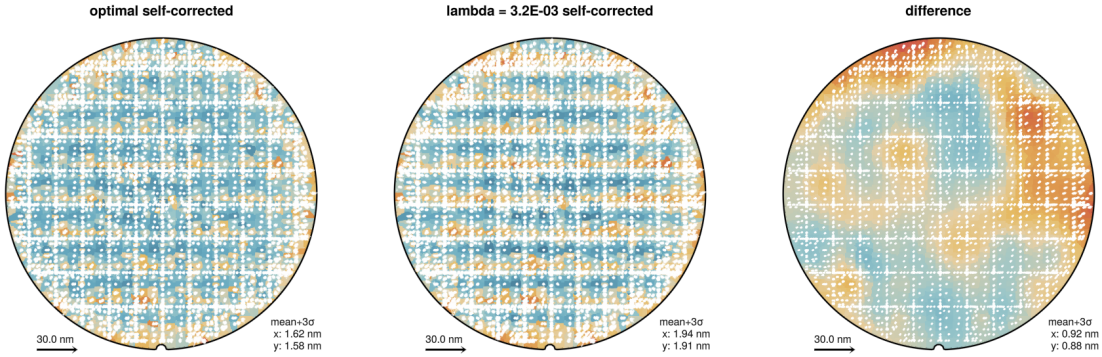

Figure 11.3: Self-corrected residuals for Foundry.

Here, we are comparing optimal regularization against bending energy regularization with $\lambda = 3.2e-3$, which was found to be the optimal parameter for this dataset with a line search. We see a very significant improvement in m3s in both the x- and y-directions. However, these results are self-corrected: we 'train' or optimize our regularization matrix on the same data that we are using to evaluate its performance. A fairer way to judge performance would be to use a test/train split, where we use separate datasets for testing and training:

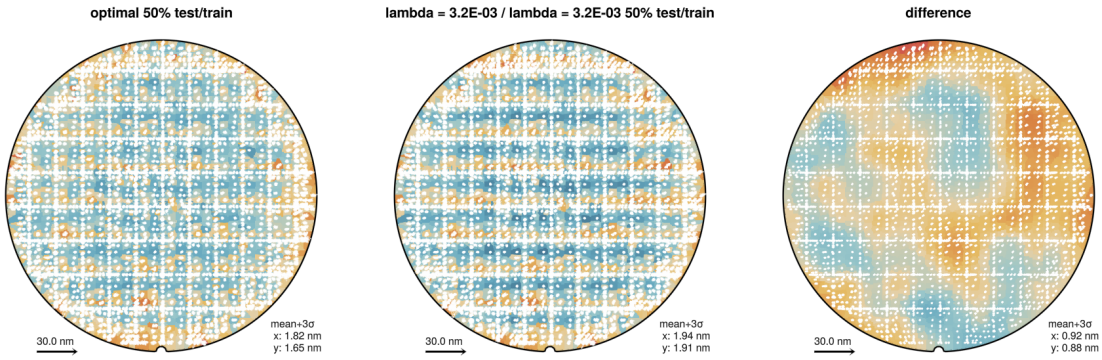

Figure 11.4: 50% test/train split residuals for Foundry.

The above data were produced using a 50% test/train split. The optimal λ was determined for each half separately. In this case we still see a significant improvement in m3s, although not as significant as the self-corrected case. This is to be expected: for this specific dataset, we have 120 model parameters. Since we are optimizing over $A \in \mathbb{R}^{120\times120}$, we have $\frac{120(120+1)}{2} = 7260$ free parameters to be determined by a dataset of 75 wafers. Thus we can expect overfitting to at least some degree when using any realistic amount of wafers.

The third plot in each of the above figures shows the difference in residuals between the optimal and bending energy regularization. We can see that optimal regularization is especially effective around the edge of the wafer, where higher deformations tend to take place.

It should be mentioned that we are using the Frobenius norm (corresponding to the RMS) of the residuals in optimization, but we are often interested in $\mu + 3\sigma$ as a statistic. Although these do not correspond exactly, using the Frobenius norm/RMS is a good approximation and above all easy to express in a semidefinite program. In fact, the Frobenius norm already corresponds closely to the column-wise standard deviation: if we have a vector $X \in \mathbb{R}^{n\times1}$ such that its columns have mean zero,

$$\text{std}(X) = \sqrt{\frac{1}{n-1}\text{Tr}(X^T X)}, \tag{11.4}$$

which is proportional to $\|X\|_F$. The Frobenius norm of a matrix then corresponds to the square root of the summed column-wise variances (if the matrix has column-wise mean 0). Regarding this last condition, if we fit the translation (4par/6par) component well, we expect the residuals to have close to zero mean. This is indeed the case for the alignment residuals (mean of order $1e-22$) but less so for the overlay residuals (mean of order $1e-11$).

It is instructive to visualize what the regularization is actually doing. To this end, we can use the following scheme. We artificially create a FIWA measurements vector with all entries set to zero, except one, which is set to 1.0nm. This vector is used to fit a model of the wafer deformation, using the regularization matrix we want to study. This model is then evaluated on the overlay marks and plotted, giving the following result:

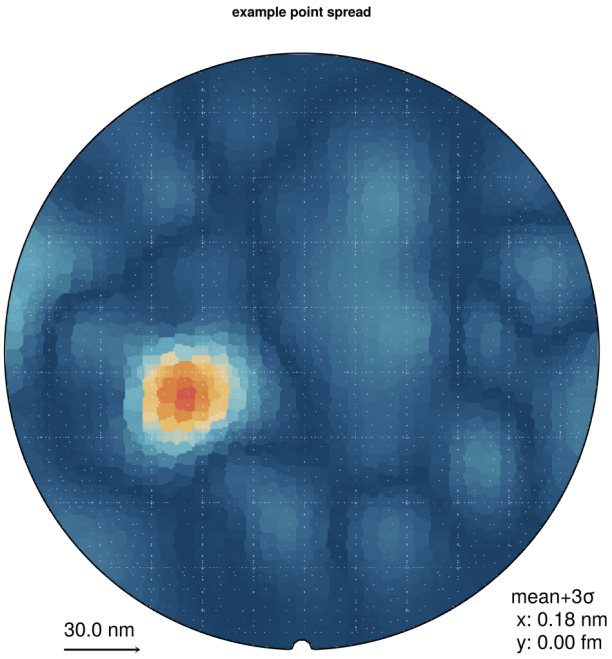

Figure 11.5: Example point spread for Foundry.

This shows us, in essence, to what extent a perturbation/measurement on a single mark 'propagates'. If the point spread at a mark is large, a measurement on that mark influences our model greatly. We would like to view this for all marks to visualize how regularization influences the propagation of measurements across the entire wafer. However, simply superimposing the point spreads as shown above would obfuscate which perturbation is coming from which wafer. Therefore we adopt the following methodology: for each point spread, we compute a notion of dispersion as follows. Starting from the model evaluated on the overlay marks, for mark i

$$\sigma_i = \frac{\sum_m (x_{nom,m} - x_{nom,i}) * x_i}{\sum_m x_i}. \tag{11.5}$$

This is similar to a standard deviation (hence the letter σ), which can be plotted per mark and superimposed to give the following image:

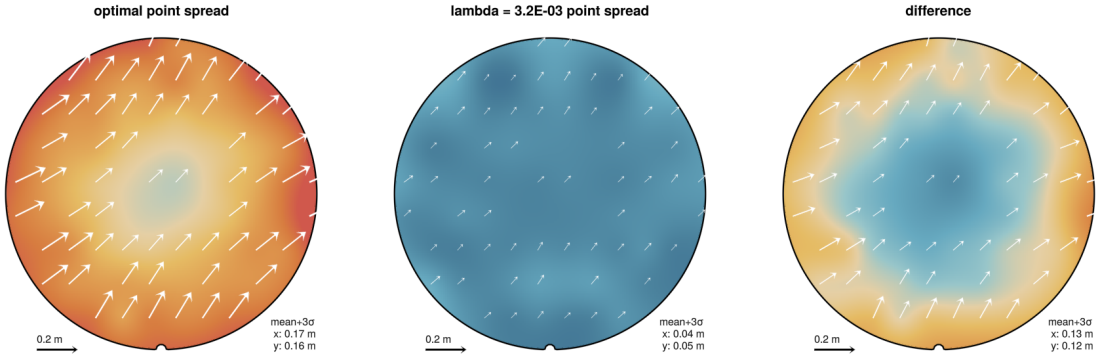

Figure 11.6: Superimposed point spreads for self-corrected optimal regularization on Foundry.

This illuminates the difference between optimized regularization and bending energy regularization.

The latter is comparatively uniform across the wafer, whereas the optimal regularization clearly shows a difference as we move across the wafer in a radial direction. Point spreads in the center seem to be much more concentrated, highlighting that wafer deformations in this area tend to be more localised. Conversely, the edge shows a higher degree of smoothing/propagation of measurements, reflecting the fact that wafers tend to show more deformations around the edges.

11.2. DRAM

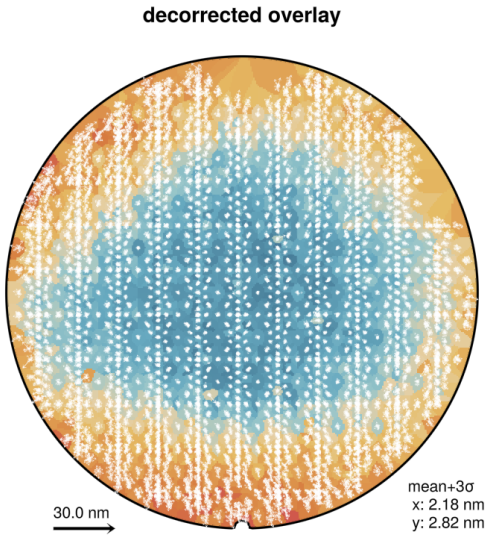

Figure 11.7: Decorrected overlay for DRAM.

Next we will look at optimal regularization for a DRAM[4] dataset.

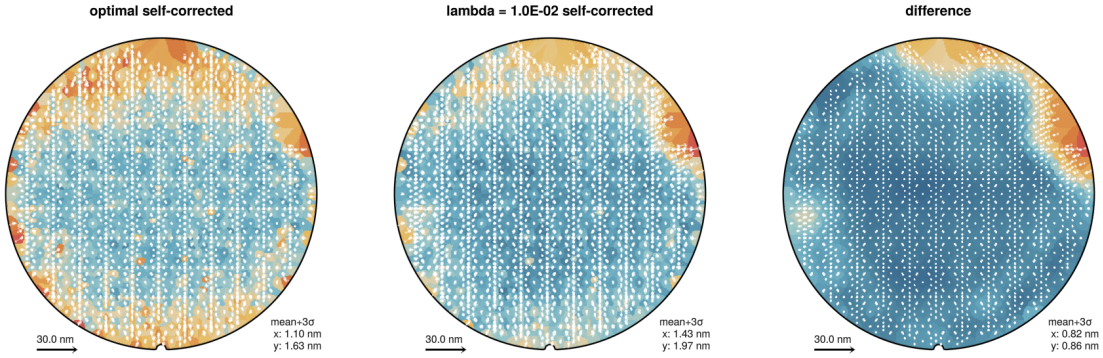

Figure 11.8: Self-corrected residuals for DRAM.

[4]Dynamic Random Access Memory

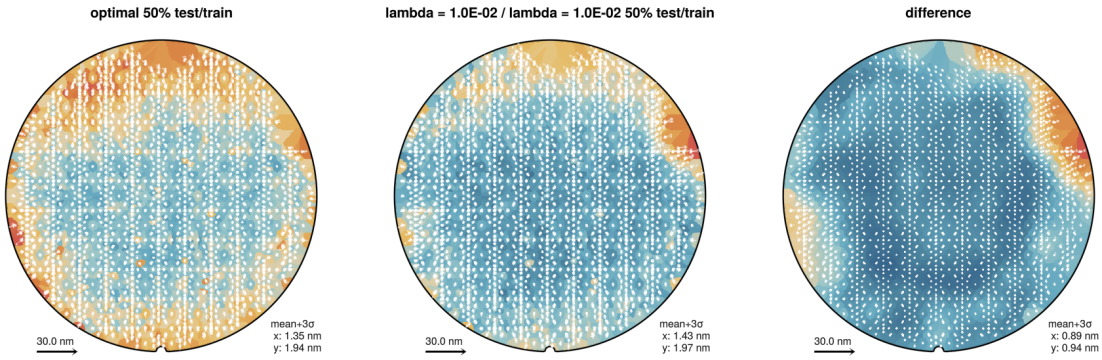

Figure 11.9: 50% test/train split residuals for DRAM.

This dataset is characterized by highly warped and variable wafers. Although the self-corrected optimal regularization still provides an improvement over bending energy, the effect is minimal. As with the previous dataset, the most pronounced effect of optimal regularization occurs around the edge.

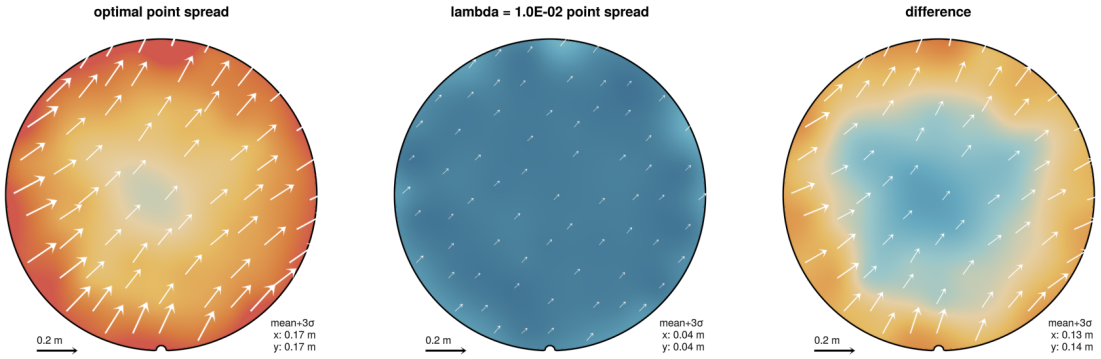

Figure 11.10: Superimposed point spreads for self-corrected optimal regularization on DRAM

11.3. Bonding

The next dataset is a bonding dataset, containing wafers that underwent a series of chemical and mechanical processes to bond them to one another in the x/y-plane. Such datasets are characterized by large deformations around the edges, where wafers are bonded, as well as large deformations in the center due to other process steps.

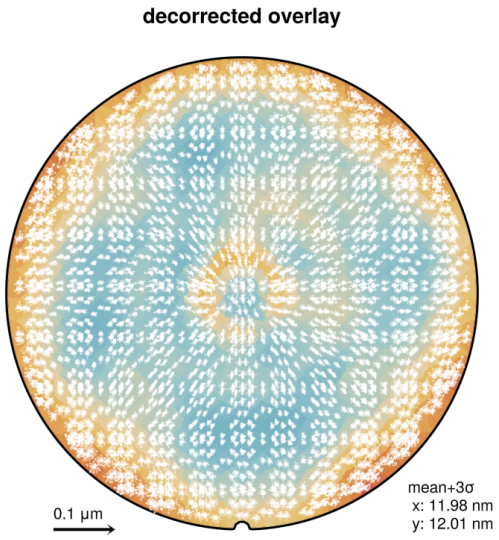

Figure 11.11: Decorrected overlay for Bonding.

This is clearly visible in the above plot of decorrected overlay for the dataset. Notice the larger scale of the arrows compared to previous plots.

The self-corrected residuals are as we would expect, and show an improvement of several hundreds of picometers:

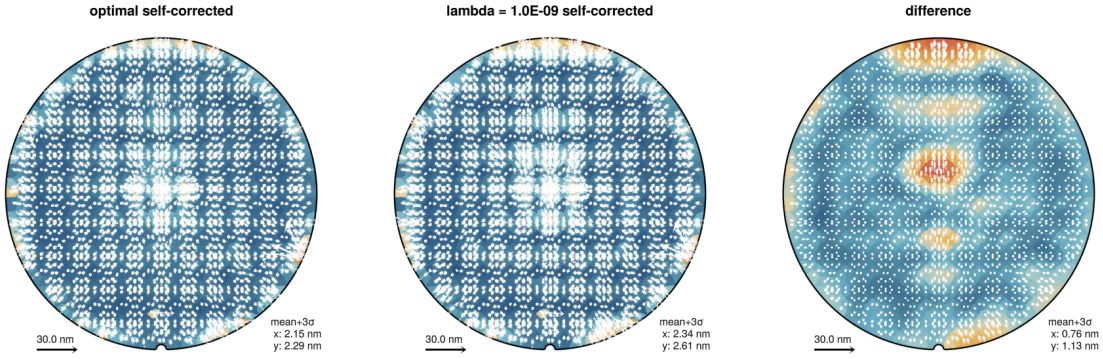

Figure 11.12: Self-corrected residuals for Bonding.

However, the 50% test/train split shows that the fitted regularization is inadequate when applied to new wafers:

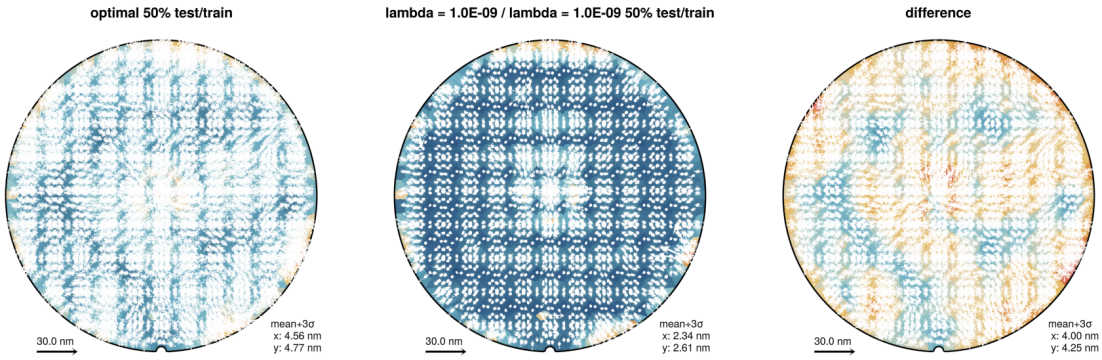

Figure 11.13: 50% test/train split residuals for Bonding.

The reason this happens is that while we are decreasing the objective function $\|Y - AX\|_F$ in the optimization, but this only fits the matrix according to the data we are providing, which consists of only a few dozen wafers. This has as a result that every component of A that is in the left null space of X does not get optimized, and is dependent on the initial value with which we start the optimization.

This problem holds for all datasets, not only bonding, but the difference here is that bonding datasets are much more sensitive to regularization due to the high deformation. This shows that an additional scheme is necessary to ensure the regularization matrix has a default value. A very simple option is to initialize the optimization with the bending energy regularization matrix that works best for the dataset:

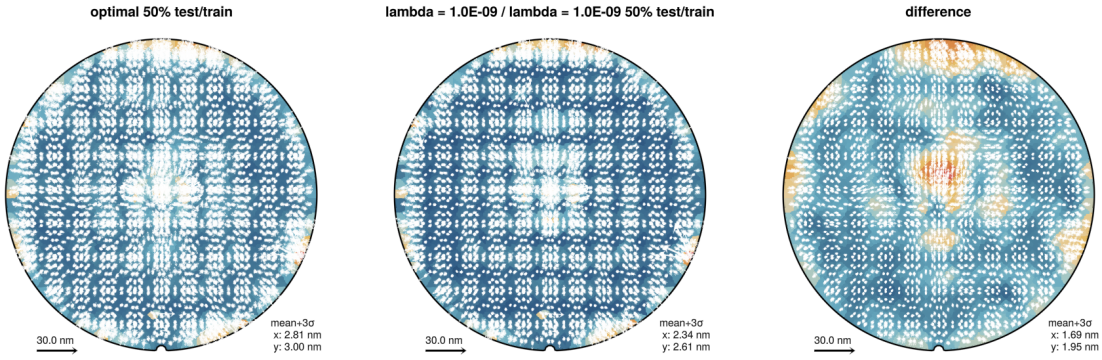

Figure 11.14: 50% test/train split residuals, initialized with bending energy regularization, for Bonding.

This already shows a very large improvement over the previous set of waferplots, but there is still an unacceptable worsening of overlay. This suggests that we are either still overfitting the training data, or erroneously fitting (linear combinations of) model parameters that are not reflected in the dataset.

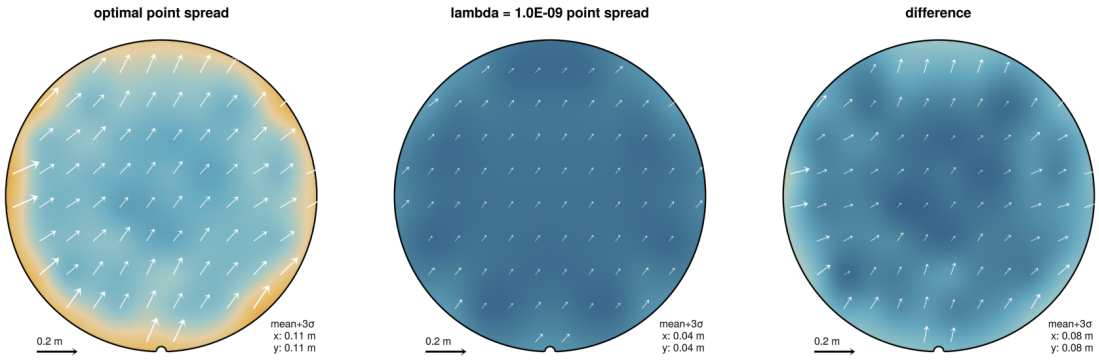

Figure 11.15: Superimposed point spreads for self-corrected optimal regularization on Bonding.

Point spreads for this dataset are also wider at the edge of the wafer, but the difference in the middle, compared to default regularization is less pronounced. Notice also that for both types of regularization, point spreads are lower than for previous datasets, reflecting that the deformations we want to fit are more localized.

12 Discussion

12.1. Different solution methods

In this thesis we have outlined three viable solution schemes for the positive semidefinite Procrustes problem:

1. Reformulation as a semidefinite program, allowing solution by off-the-shelf solvers,
2. Using Dykstra's alternating projection algorithm, solving the full-rank subproblem with a gradient method,
3. Using a gradient method, reformulating the problem such that feasibility coincides with the eigenvalues lying within the interval $[0, 1]$.

The advantages and disadvantages of these approaches are listed below.

12.1.1. Semidefinite programming

A major advantage of reformulating as a semidefinite program is the possibility of using off-the-shelf commercial or open-source solvers. The codebase for such solvers is often well-maintained and highly optimized, making the implementation of the optimization problem comparatively simple. However, most SDP solvers were found to be too slow (solution time > 1 minute) to be used. COSMO [21] did provide reasonable performance, because quadratic objectives can be natively supplied to the solver and because it is a first-order method. This removes the need for a reformulation as detailed in 6 and thus allows for a smaller variable size. Still, COSMO was found to be relatively slow in comparison to the other proposed methods, and may provide issues with portability to other platforms (e.g. on-machine hardware).

12.1.2. Dykstra's projection algorithm

Using Dykstra's projection algorithm allows us to iteratively solve the double-bounded Procrustes problem by solving two instances of the single-bounded problem at each iteration. This really only provides an efficient solution scheme if we are able to solve the single-bounded problem with low computational cost. As of now, a correct closed-form solution to the full-rank single bounded Procrustes problem is not known, forcing us to resort to (projected) gradient methods to this end. However, such methods usually employ an eigenvalue decomposition at each iteration to project the iterate onto the feasible set, which is computationally expensive. Empirically, Gillis and Sharma's Fast Gradient Method [2] on average required no fewer than hundred iterations to converge within reasonable tolerance. Since Dykstra's projection algorithm projects first on the upper, then on the lower bound at each iteration, each iteration of the outer loop will require no fewer than a few hundred eigenvalue decompositions. Even with comparatively fast convergence, this is computationally expensive and unfavorable compared to a gradient method that projects onto both bounds at once. Of course, Dykstra's projection method would again be an interesting option if a correct closed-form solution to the single-bounded Procrustes problem becomes available.

12.1.3. Projected gradient method

The projected gradient method has the advantage of being simple to understand and implement. The reformulation using the decomposition of the upper bound matrix B adds complexity and may cause our algorithm to converge more slowly, but provides the major advantage of only requiring one eigenvalue decomposition per projection onto the feasible set. If we did not perform this transformation, a scheme like Dykstra's algorithm would be necessary at each iteration. Furthermore, projected gradient methods are known to converge well and we can give theoretical bounds on convergence relatively easily.

12.2. Robustness of optimal regularization

We have seen in the results (Chapter 11) that the optimal regularization matrix, despite mathematical optimality, may fail to provide benefit when fitting a model to never-seen-before wafers. This happens especially for highly warped and/or highly variable datasets, which may require a greater number of wafers as input to the algorithm to derive a sufficiently stable regularization matrix. In such cases, we would want to stay close to the bending energy regularization. Further research could also focus on determining an appropriate minimum number of wafers to optimize regularization for diffferent types of datasets. Lastly, the results described in the previous section are preliminary: a great deal of validation and testing needs to be carried out to determine whether optimal regularization methods offer a significant and consistent benefit across a large variety of datasets.

13

Conclusion

We have posed the double-bounded positive semidefinite Procrustes problem

$$\begin{aligned} \min_{A} \quad & \|Y - AX\|_F \\ \text{s.t.} \quad & A \succeq 0 \\ & B - A \succeq 0, \end{aligned} \tag{13.1}$$

originating from the optimal model regularization problem for wafer alignment:

$$R = \underset{R \succeq R_{min}}{\operatorname{argmin}} \quad \left\| M_y Y - M_y (M_x^T M_x + R)^{-1} M_x^T x \right\|_F . \tag{13.2}$$

Problem 13.1 is convex and always attains its optimum, but is difficult to solve due to the upper and lower bounds, introducing nonlinearity.

Several numerical methods have been introduced, including a semidefinite programming approach, an alternating projection method and a projected gradient method. Out of these, the projected gradient method is the most efficient: applying the decomposition $B = QQ^T$, the transformation

$$A' = Q^{-1} A (Q^{-1})^T \tag{13.3}$$

$$Y' = Q^{-1} Y \tag{13.4}$$

$$X' = Q^T X \tag{13.5}$$

allows us to write the constraints in 13.1 as

$$I \succeq A' \succeq 0, \tag{13.6}$$

which corresponds exactly with the condition that all eigenvalues of A' lie in the interval $[0, 1]$. This enables a relatively simple projection onto the feasible region, employing only one eigenvalue decomposition per iteration.

Optimal regularization was tested on different customer datasets and was shown to have a benefit of several tens to hundreds of picometers on most data. The scheme failed however for data with a high degree of deformation and with high variance among the different wafers in the dataset. Essentially, this is caused by the data supplied to the optimization not spanning the full model parameter space, causing a part of the regularization matrix to be left unoptimized and dependent on initial values. A suggestion for extended research on this topic would include a more sophisticated scheme for combining the data-driven optimal regularization matrix with the 'standard' bending energy matrix we know to work well for the dataset at hand. In addition, the scheme for optimal regularization needs to be validated on more datasets and tested extensively for accuracy, consistency and performance.

References

[1] Nicholas J Higham. "The symmetric Procrustes problem". In: *BIT Numerical Mathematics* 28 (1988), pp. 133–143.

[2] Nicolas Gillis and Punit Sharma. "A semi-analytical approach for the positive semidefinite Procrustes problem". en. In: *Linear Algebra and its Applications* 540 (Mar. 2018), pp. 112–137. ISSN: 00243795. DOI: 10.1016/j.laa.2017.11.023. URL: https://linkinghub.elsevier.com/retrieve/pii/S0024379517306511 (visited on 09/10/2024).

[3] K.G. Woodgate. "A new algorithm for the positive semi-definite Procrustes problem". In: *Proceedings of 32nd IEEE Conference on Decision and Control*. San Antonio, TX, USA: IEEE, 1993, pp. 3596–3601. ISBN: 978-0-7803-1298-2. DOI: 10.1109/CDC.1993.325890. URL: http://ieeexplore.ieee.org/document/325890/ (visited on 09/10/2024).

[4] Alison Li and Jessica Timings. *6 crucial steps in semiconductor manufacturing*. Oct. 2023. URL: https://www.asml.com/en/news/stories/2021/semiconductor-manufacturing-process-steps.

[5] *See ASML's metrology and inspection systems*. URL: https://www.asml.com/en/products/metrology-and-inspection-systems.

[6] Lieven Vandenberghe and Stephen Boyd. "Semidefinite programming". In: *SIAM review* 38.1 (1996), pp. 49–95.

[7] Peng Jingjing et al. "Solution of symmetric positive semidefinite Procrustes problem". In: *The Electronic Journal of Linear Algebra* 35 (Dec. 2019), pp. 543–554. ISSN: 1081-3810. DOI: 10.13001/ela.2019.5167. URL: https://journals.uwyo.edu/index.php/ela/article/view/5167 (visited on 09/10/2024).

[8] Arthur Albert. "Conditions for positive and nonnegative definiteness in terms of pseudoinverses". In: *SIAM Journal on Applied Mathematics* 17.2 (1969), pp. 434–440.

[9] Keith G. Woodgate. "Least-squares solution of F = PG over positive semidefinite symmetric P". In: *Linear Algebra and its Applications* 245 (Sept. 1996), pp. 171–190. ISSN: 0024-3795. DOI: 10.1016/0024-3795(94)00238-X. URL: https://www.sciencedirect.com/science/article/pii/002437959400238X (visited on 09/10/2024).

[10] Zuzana Bílková and Michal Šorel. "Projection methods for finding intersection of two convex sets and their use in signal processing problems". In: *Electronic Imaging* 33 (2021). Publisher: Society for Imaging Science and Technology, pp. 1–6.

[11] Ward Cheney and Allen A Goldstein. "Proximity maps for convex sets". In: *Proceedings of the American Mathematical Society* 10.3 (1959), pp. 448–450.

[12] Stephen Boyd et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends® in Machine learning* 3.1 (2011), pp. 1–122.

[13] Alfredo N Iusem. "On the convergence properties of the projected gradient method for convex optimization". In: *Computational & Applied Mathematics* 22 (2003), pp. 37–52.

[14] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. Chap. 9, p. 461.

[15] Heinz H Bauschke and Jonathan M Borwein. "On projection algorithms for solving convex feasibility problems". In: *SIAM review* 38.3 (1996), pp. 367–426.

[16] Miles Lubin et al. "JuMP 1.0: Recent improvements to a modeling language for mathematical optimization". In: *Mathematical Programming Computation* (2023). DOI: 10.1007/s12532-023-00239-3.

[17] Brian Borchers. "CSDP, A C library for semidefinite programming". In: *Optimization Methods and Software* 11.1-4 (1999), pp. 613–623. DOI: 10.1080/10556789908805765. eprint: https://doi.org/10.1080/10556789908805765. URL: https://doi.org/10.1080/10556789908805765.

[18] Chris Coey, Lea Kapelevich, and Juan Pablo Vielma. "Solving natural conic formulations with Hypatia.jl". In: *INFORMS Journal on Computing* 34.5 (2022), pp. 2686–2699. DOI: https://doi.org/10.1287/ijoc.2022.1202.

[19] Mario Souto, Joaquim D. Garcia, and Álvaro Veiga. "Exploiting low-rank structure in semidefinite programming by approximate operator splitting". In: *Optimization* (2020), pp. 1–28. DOI: 10.1080/02331934.2020.1823387. URL: https://doi.org/10.1080/02331934.2020.1823387.

[20] Brendan O'Donoghue et al. "Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding". In: *Journal of Optimization Theory and Applications* 169.3 (June 2016), pp. 1042–1068. URL: http://stanford.edu/~boyd/papers/scs.html.

[21] Michael Garstka, Mark Cannon, and Paul Goulart. "COSMO: A Conic Operator Splitting Method for Convex Conic Problems". In: *Journal of Optimization Theory and Applications* 190.3 (2021), pp. 779–810. DOI: 10.1007/s10957-021-01896-x. URL: https://doi.org/10.1007/s10957-021-01896-x.