A mathematical Language for MRI Reconstructions

Version 0.0: “Little Wolf”

Previous Versions: none

Date: 2023, June 20

Author: Bastien Milani

Contacts:

bastien.milani@yahoo.fr

Supporting Institutions

University of Lausanne (UNIL), Lausanne, Switzerland

Center for Biomedical Imaging (CIBM), Lausanne, Switzerland

Centre Hospitalier Universitaire Vaudois (CHUV), Lausanne, Switzerland

Bistro de l’Atlantic, Malley, Switzerland

Founding

Swiss National Science Foundation (SNSF) grant number 320030B_201292

Dotation of Professor Matthias Stuber (UNIL), Lausanne, Switzerland

Table of Contents

Introduction

Dear Reader

Please allow me to introduce myself, I am Bastien, the author of this text. Enjoy this reading with a cup of coffee, or any substance you like if it the side effects are not too damageable. Because life should not be boring, we allow sometimes an informal language, in particular when giving some intuitive indication in order to help the understanding. But the goal is really to present a formal mathematical material, or as formal as possible whenever approximations or hypothesis needs to be done. The author considers that there is something human in the contrast between the form and the content. And please excuse my inexact English, I am not a native English speaker. If you want to write me to propose some corrections, you are welcome.

By the way, this very first version of the text is released only with part I (Mathematical Background) and part II (Algorithms). And also with the first chapter of part III (Discretization). The other chapters of part III and the next parts are in construction: part IV (MRI-reconstructions) and part V (Implementations). I don't really know when they will be ready. The reason why I publish the first parts of that text is that I am in a situation where I need it to serve as supplementary material for scientific publications.

The present text can be considered as something between a course and a book. If you found it, you probably know that "MRI" means "magnetic resonance imaging", or "nuclear magnetic resonance imaging" for purists. This text was originally written for the image reconstructions of MRI data acquired along non-Cartesian trajectories (non-Cartesian MRI), which are essentially iterative reconstructions, with a few exceptions such as gridded reconstructions. The fact that a trajectory is non-Cartesian makes the reconstructions more subtle, more complicated than for a Cartesian trajectory. One of the motivations for writing this text came in fact from a feeling of the author, that non-Cartesian MRI reconstruction deserved a specific or dedicated mathematical formulation because the usual mathematical tools used in Cartesian reconstruction are possibly not suited for implementing non-Cartesian MRI reconstructions. That is why the text was originally named "A Language for non-Cartesian MRI reconstructions". However, all the mathematical material and algorithms described in the present text are valid for any trajectory, including partially sampled Cartesian-trajectories and fully sampled Cartesian trajectories, hence the title "A Language for MRI reconstructions". Another source of motivation for writing this text was the fact that mathematical algorithms for solving the optimization problems we encounter in MRI reconstruction are usually described in a general abstract (not practically

oriented) way by mathematician who are specialists in optimization, while the implementations of these algorithms for MRI reconstructions are often done by engineers who speak a different language than mathematicians. The misunderstanding between these two worlds was one of the reasons why this text was written, with the hope that it could help them to collaborate. Whenever I had the feeling that well-known mathematical results, definitions or algorithms could be generalized, stated from a different view-point, reformulated or described in details, in some way that could be beneficial for MRI reconstruction scientists, I tried to do my best to achieve it.

One last thing: we break a few conventions of the mathematician community in this text. Notably, I don't put any dot nor comma after any equation. My mathematician friends threated me with a knife because of that. But I will not put those dots. They are too ugly. I prefer to go to hell. Another thing that mathematicians will never do in a text is to propose a pronunciation for some symbols. Here we do that. Mathematicians are not aware that when engineers encounter some symbol such as "$A^\dagger$" or "ξ", they may stop reading the text because they don't know how to pronounce the symbol in their head. We are human and we think with our senses, even if we construct mathematical objects. The author believes that neglecting that is a problem.

This text is written (currently in development) in the intentions to reach following aims:

- to develop a notation, a formal mathematical description, a language for MRI reconstruction,
- to describe examples of MRI reconstruction algorithms with the developed language as formally as possible,
- to point out where approximation and hypothesis need to be done in algorithms because the mathematical formalism cannot be practically achieved.
- To present some real-world implementations.

In this first version however, we only present some mathematical background and algorithms, which includes in particular

- a detailed description of the conjugate-gradient-descent-algorithm for solving different instances of the least-square problem,
- the ADMM-algorithm for solving the generalized-LASSO problem, including the generalized-LASSO problem with multiple 1-norm terms,

Plan of the text

The text is divided into parts, the parts are divided into chapters and chapters are divided into sections.

Part 1 is about mathematics only and contains the following chapters:

- In chapter 1 we define mathematical notations and review some results coming mostly from linear algebra and sometimes from convex analysis or optimization theory. In particular, we develop a notation in order to represent any vector space isomorphic to $\mathbb{C}^n$ as a vector space isomorphic to $\mathbb{R}^{2n}$. All vector spaces we encounter are finite-dimensional and all the presented linear algebra material exploits the presence of a real Euclidean product on each vector space, which may be different from the canonical Euclidean product on $\mathbb{C}^n$ and $\mathbb{R}^{2n}$. If A is the matrix of a linear map between vector spaces X and Y, we will explain that the matrix $A^\dagger$ of the adjoint map verifies

$$A^\dagger = H_X^{-1} A^* H_Y \quad (*)$$

where A^* is the complex-conjugate-transpose of matrix A and H_X resp. H_Y are suitable matrices related to the real-valued Euclidean products on X resp. Y (details are given in section 1.3).
We also present in chapter 1 a different definition of the gradient of a function. Namely, we set

$$grad_x(f) = H_X^{-1} \nabla f \quad (**)$$

where H_X is the matrix present in $(*)$. This definition is particularly suited for vector spaces with 2-norms (defined later in the text) that are different from the standard 2-norms. In particular, given the squared-norm function

$$x \longmapsto \|Ax - y\|_{Y,2}^2 \quad (***)$$

where A is any linear map from X into a (finite dimensional) vector space Y and where $\|\cdot\|_{Y,2}^2$ is a 2-norm on Y, its gradient (as defined in $**$) has the familiar expression

$$A^\dagger(Ax - y)$$

It is worth to point out that this alternative definition makes the negative gradient ($-grad_x(f)$) parallel to the normalized direction of steepest descent for any differentiable function.

- In chapter 2, we demonstrate that a version of the conjugate gradient descent method (CGD-method or CGD-algorithm), can be applied to the least-square problem and always converges to a minimizer of the squared norm function $(***)$. In fact this is true for any linear map A in $(***)$, while most manuals restrict usually their description to the case where the symmetric map $A^{\dagger}A$ is positive definite. In order to demonstrate this convergence, we first reformulate the least-square problem as an invertible problem that can be solved abstractly (free of concrete evaluation) with CGD-algorithm, and we then show that the CGD applied to the original least-square problem generate the same sequence as the invertible problem, up to a translation.
- In chapter 3, we describe the generalized-LASSO problem.

Part 2 is about algorithms and is implementation oriented. We describe CGD-algorithm for 1-term, 2-terms and multiple-terms least-square problems, and we write some pseudo-code of those algorithms in order to help their implementation. We describe then the ADMM-algorithm for solving the generalized-LASSO problem (including multiple 1-norm terms), which is one of the central algorithms for today's MRI reconstruction.

Part 3 contains only its first chapter for the moment. It describes how the real Euclidean spaces we work on arise from a discrete approximation of the physical continuous world.

As this version 0.0 is the very first version of the book, we encourage the reader to contact the author if he or she thinks that corrections or missing references, additional material, … should be brought to the text.

Acknowledgments

I thank in first place my colleague and friend **Jean-Baptiste Ledoux** for fighting next to me with his unequaled skills in MRI acquisitions. The only tech who find the errors in my acquisition protocols.

I would like to thank my friend **Gianluca Giacchi** for proofreading the text, for his suggestions, fruitful discussions and (more importantly) unique sense of humor. I propose to solve the problem of dot-after-equations with a sword duel.

An energy without comparison was brought by **Benedetta Franceschiello** who's force spreads in researchers. What we cannot prove we have to believe in.

I thank **Jerome Yerly** for proofreading the text and for his pioneering abilities in implementations.

I am grateful to **Pina Marziliano** who can read between the lines. Thank for the support.

Finally I would like to gratefully thank **Matthias Stuber**, who believed in me since he never fired me, even if I have spent the three first years of my postdoc writing theories without producing a single image. Thanks again.

Part I: Mathematical Background

In this part I, we describe mathematical notions and algorithms. This part contains no approximation nor physical hypothesis. It contains no physics and there is no direct implication to MRI. It is a pure mathematical description. We limit to point out, sometimes, how some mathematical notions are linked to practical imaging concepts in order to help the intuition.

1. Euclidean spaces

1.1 Some language and definitions

Any vector-space we encounter in this text is finite dimensional. We recall the definition of a vector space over a field:

Let be X a non-empty set and $(\mathbb{K},\ \oplus\ ,\odot\ )$ a field ($\oplus$ is the addition in $\mathbb{K}$ and $\odot$ is the multiplication in $\mathbb{K}$). Then is $(X, \mathbb{K}, +,\ \cdot)$ a vector space over the field $(\mathbb{K},\ \oplus\ ,\odot\ )$ if the following axioms are satisfied:

- " $+$ " is a binary operation (called "addition") from $X \times X$ to X which is so that $(X, +)$ is an abelian group (i.e. a commutative group). That is
  - (a) $\forall\ x, y, z \in X$ holds $x + (y + z) = (x + y) + z$ (associativity),
  - (b) there exist a unique neutral element $0 \in X$ such that $x + 0 = 0 + x = x\ \ \forall\ x \in X$,
  - (c) for any $x \in X$ there exist a unique element $x^{-1} \in X$ (the additive inverse of x) so that $x + \ x^{-1} = x^{-1} + x = 0$,
  - (d) it holds $x + y = y + x\ \ \forall\ x, y \in X$ (commutativity).
- " $\cdot$ " is a binary operation (called "scalar multiplication") from $\mathbb{K} \times X$ to X that verifies the following axioms: $\forall \alpha, \beta \in \mathbb{K}$ and $\forall x, y \in X$ holds
  - (a) $\alpha \cdot (\beta \cdot x) = (\alpha \odot \beta) \cdot x$
  - (b) $\alpha \cdot (x + y) = \alpha \cdot x + \alpha \cdot y$
  - (c) $(\alpha \oplus \beta) \cdot x = \alpha \cdot x + \beta \cdot x$
  - (d) the neutral element $\mathbf{1}$ of the multiplication in $\mathbb{K}$ verifies $\mathbf{1} \cdot x = x$.

In order to simplify the notation, we will just write $\alpha\, x$ instead of $\alpha \cdot x$, we will write $\alpha\, \beta$ instead of $\alpha \odot \beta$, and we will write $\alpha + \beta$ instead of $\alpha \oplus \beta$. We use then same symbol for the addition in $\mathbb{K}$ and in X but the arguments make it clear which addition is written. The same remark holds for the multiplication. In this text, we will only encounter the field of complex numbers $\mathbb{C}$ with complex addition and multiplication, and the field of real numbers $\mathbb{R}$ with real addition and multiplication.

A linear function between two vectors spaces will often be designated as a "**vector space homomorphism**" of just "**homomorphism**" for short. We write it for example as

$$A: X \longrightarrow Y$$

$$x \longmapsto Ax$$

We will also call it a "**linear map**" or just a "**map**". An homomorphism from a vector space to itself is called an "**endomorphism**" and if it is invertible it is called and "**isomorphism**". We recall that a linear map A from X to Y is any function such that

$$A(\alpha\, a + \beta\, b) = \alpha\, Aa + \beta\, Ab$$

In most of the sections, we assume that every vector space is readily endowed with a vector basis. When that is true, a vector-space homomorphism is thus uniquely represented by a matrix and a vector (i.e. an element of a vector space) is uniquely represented as a column vector of coordinates. By abuse of notation, if A is the matrix of a homomorphism, we will then also write the homomorphism itself with letter A, the context being in principle clear enough to allow disambiguation. In order to state that a vector space X is a real n-dimensional vector space, we will write $X \simeq \mathbb{R}^n$. That means the choice of a basis on X allows to write each vector as a coordinate vector in $\mathbb{R}^n$. Similarly, in order to state that a vector space X is a complex n-dimensional vector space, we will write $X \simeq \mathbb{C}^n$. That means the choice of a basis on X allows to write each vector as a coordinate vector in $\mathbb{C}^n$.

If $X_1 \simeq \mathbb{R}^n$ and $X_2 \simeq \mathbb{R}^n$, we will like to say that X_1 and X_2 are two "**copies**" of the same vector space $\mathbb{R}^n$, although that language is not a formal mathematical convention. In fact are X_1 and X_2 different (they differ at least by their name), but both are isomorphic to the same structure $\mathbb{R}^n$. The author think that "being a copy of $\mathbb{R}^n$ or $\mathbb{C}^n$" is less barbaric than "…being isomorphic to…".

In some of the sections, we break the convention of assuming the presence of a vector basis and we treat homomorphisms on vector spaces in an abstract way i.e. without any choice of vector basis. In that context, any choice of a basis allows writing any homomorphism as a matrix, and any vector as a vector of coordinates, but we let the choice of the basis open. The use of that convention will be mentioned at the beginning of the sections in question.

We will use the symbol " ≔ " in order to write "equal by definition".

The **complex-conjugate-transpose** of any complex-valued matrix A will be written A^* (pronounced "A **star**") and its **transpose** matrix will be written A^T (pronounced "A **transpose**"). The real-valued matrix consisting of the real part of A will be written $\mathrm{real}(A)$ or just rA , and the real-valued matrix consisting of the imaginary part of A will be written $\mathrm{imag}(A)$ or just iA. We note that iA is real-valued and is a non-dissociable symbol. It does NOT mean "i times A. "

The **complex-conjugate-transpose** of column vector x is a raw vector that will be written x^* (pronounced "x **star**"). The transpose of column vector x is a raw vector that will be written x^T (pronounced "x **transpose**"). The real-valued vector consisting of the real part of a column vector x will be written $\mathrm{real}(x)$ or just rx. The real-valued vector consisting of the imaginary part of a column vector x will be noted $\mathrm{imag}(x)$ or just ix.

The matrix-matrix multiplication of matrices A and B will be written AB. The matrix-vector multiplication of matrix A and vector v will be written Av. If v is a raw-vector and x a column vector, their product $v\,x$ is given by interpreting v as a matrix with one single line and x as a matrix with one single column. Typically, given two real-valued column vectors x and w, then is $w^T x$ their standard Euclidean product. If x and w are complex-valued is $w^* x$ their standard Hermitian product.

The **complex-conjugate** of a complex number α will be written α^* (and not $\bar{\alpha}$ because we keep the bar-symbol for another notion).

Excepted homomorphisms, any function f will be written as $f(\cdot)$ in order to stress the fact that it is a function and not a vector or a number. This didactic notation is more cumbersome than just writing f but we gain in clarity and it allows us to write vectors like f, without an arrow on the top. For example, if the function is complex-valued, $f(\cdot)$ describes the function while $f(x)$ describes the complex value obtained by evaluating function $f(\cdot)$ on argument x. Sometimes, we will have to break that convention because the notation would become unpractical. We will always mention when that convention is broken.

Given a homomorphism

$$A : X \longrightarrow V$$

$$x \longmapsto A\,x$$

from a vector space X to a vector space V, the kernel (or null-space) of A will be written $\mathrm{Ker}(A)$ (an other notation is $\aleph(A)$) and is defined as the sub-linear space of X given by

$$\mathrm{Ker}(A) := \{x \in X \mid A\,x = 0\}$$

The image of A (or range of A) will be written $\mathrm{Im}(A)$ (an other notation is $\mathrm{Rg}(A)$) and is the sub-linear space of V given by

$$\mathrm{Im}(A) = \{v \ \in V \mid \exists\, x \in X : A\,x = v\} = \{Ax \mid x \in X\}$$

We will write the square root of -1 with symbol j so that holds

$$A = rA + j\,iA \qquad x = rx + j\,ix \qquad \alpha = r\alpha + j\,i\alpha$$

$$A^* = rA^T - j\,iA^T \qquad x^* = rx^T - j\,ix^T \qquad \alpha^* = r\alpha - j\,i\alpha$$

Speaking about this symbols, we will pronounce rA as "are A" or "**real-part of** A" or just "**real-part** A". We will pronounce iA as "I A" or "**imaginary part of** A" or just "**imag-part** A". In general, we will pronounce $\mathrm{real}(\dots)$ as the **"real-part of** (…)" or just "**real-part** (…)". We will pronounce $\mathrm{imag}(\dots)$ as the "**imaginary-part of** (…)" or just "**imag-part** (…)".

1.2 Real and complex description of a complex n-dimensional vector-space

Given two vector spaces $X \simeq \mathbb{C}^n$ and $Y \simeq \mathbb{C}^m$, we assume the existence of a vector basis in each of them. All vectors $x \in X$ and $y \in Y$ are thus uniquely represented by vectors of complex-valued coordinates and any homomorphism from X to Y is uniquely represented by a complex-valued matrix. We present in this section two different but equivalent descriptions of such vectors and matrices. We will call them the "real-description" and the "complex-description". In fact, each one is a mathematical "representation" of the other. But we will not use the term "representation" in this text because we don't want to go into the theory of mathematical representations, which is something defined very formally in mathematics.

For any vector $x \in X \simeq \mathbb{C}^n$, we define its complex description as the direct one i.e. x itself. In terms of real and imaginary parts it means

$$x = rx + j\,ix$$

The same holds on Y. For any matrix $A \in \mathbb{C}^{m\times n}$ of an homomorphism between two vector spaces $X \simeq \mathbb{C}^n$ and $Y \simeq \mathbb{C}^m$ we define its complex description as the direct one, i.e. A itself. In term of real and imaginary part it means

$$A = rA + j\ iA$$

We define the real description of a vector $x \in \mathbb{C}^n$ respectively a matrix $A \in \mathbb{C}^{m\times n}$ as the block vector resp. block matrix

$$\begin{bmatrix} rx \\ ix \end{bmatrix} \in \mathbb{R}^{2n} \qquad \text{resp.} \qquad \begin{bmatrix} rA & -iA \\ iA & rA \end{bmatrix} \in \mathbb{R}^{2m\times 2n}$$

Formally, this defines two functions defined by

$$\mathcal{R}(\cdot) : \mathbb{C}^n \longrightarrow \mathbb{R}^{2n}$$

$$x \longmapsto \mathcal{R}(x) \coloneqq \begin{bmatrix} rx \\ ix \end{bmatrix}$$

and

$$\mathcal{R}(\cdot) : \mathbb{C}^{m\times n} \longrightarrow \mathbb{R}^{2m\times 2n}$$

$$A \longmapsto \mathcal{R}(A) \coloneqq \begin{bmatrix} rA & -iA \\ iA & rA \end{bmatrix}$$

We do an abuse of notation by choosing the same symbol $\mathcal{R}$ for two different functions. But the argument makes it clear, which function is selected (programmers would talk about "overload of function name"). Therefore, given a vector $y \in \mathbb{C}^m$ in another vector space, we also write $\mathcal{R}(y)$ for its real description.

Instead of writing $\mathcal{R}(A)$ resp. $\mathcal{R}(x)$ we will also write $\mathcal{R}A$ resp. $\mathcal{R}x$ in order to simplify the notation. We will pronounce $\mathcal{R}(A)$ as "**real** A" or "**real description of** A" or just "**real of** A". This has to be discriminated from rA which is pronounced "are A" or "real-part of A" or just "real-part A". In general we will pronounce $\mathcal{R}(\dots)$ as the "**real description of** (…)" or just "**real of** (…)". The symbol $\mathcal{R}$ will be pronounced "**real**", while symbol "r" will be pronounced "are".

For the sake of completeness, consider the following simple example: hereafter on the left is the **complex description** of a 4-dimensional complex vector. On the right stands the **real description** of the same complex vector. We want to interpret it as two different description of the same vector.

$$x = \begin{bmatrix} 4 + j\,6 \\ 3 - j\,3 \\ 7 + j\,0 \\ 0 + j10 \end{bmatrix} \qquad \mathcal{R}x = \begin{bmatrix} 4 \\ 3 \\ 7 \\ 0 \\ 6 \\ 3 \\ 0 \\ 10 \end{bmatrix}$$

The inverse of $\mathcal{R}$ will be written $\mathcal{R}^{-1}$ and will be pronounced "**real-inverse**". It verifies obviously

$$\mathcal{R}^{-1}\begin{bmatrix} rx \\ ix \end{bmatrix} = rx + j\,ix \qquad \mathcal{R}^{-1}\begin{bmatrix} rA & -iA \\ iA & rA \end{bmatrix} = rA + j\,iA$$

It can be shown that all vector operations commutes with $\mathcal{R}$ and $\mathcal{R}^{-1}$. That means that these two descriptions are fully equivalent. They are two descriptions of the same thing.

For example the complex matrix-vector multiplication verifies

$$A \cdot x = (rA + j\,iA)(rx + j\,ix) = (rA \cdot rx - iA \cdot ix) + j\,(iA \cdot rx + rA \cdot ix)$$

whereas in the real description it reads

$$\mathcal{R}A \cdot \mathcal{R}x = \begin{bmatrix} rA & -iA \\ iA & rA \end{bmatrix} \cdot \begin{bmatrix} rx \\ ix \end{bmatrix} = \begin{bmatrix} rA \cdot rx - iA \cdot ix \\ iA \cdot rx + rA \cdot ix \end{bmatrix}$$

This demonstrates that

$$\mathcal{R}(A \cdot x) = \mathcal{R}A \cdot \mathcal{R}x$$

It can be shown similarly that the product of matrices A and B verifies

$$\mathcal{R}(A \cdot B) = \mathcal{R}A \cdot \mathcal{R}B$$

As well,

$$A^* = rA^T - j\,iA^T$$

and

$$\begin{bmatrix} rA & -iA \\ iA & rA \end{bmatrix}^T = \begin{bmatrix} rA^T & iA^T \\ -iA^T & rA^T \end{bmatrix}$$

shows that

$$\mathcal{R}(A^*) = \mathcal{R}(A)^T$$

It can also be shown that A is invertible if and only if $\mathcal{R}A$ is invertible and it holds in that case

$$\mathcal{R}(A^{-1}) = (\mathcal{R}A)^{-1} =: \mathcal{R}A^{-1}$$

We avoid the definition of the symbol $\mathcal{R}(x^T)$ or $\mathcal{R}(x^*)$ for any raw vector x^T or x^*.

We will note $Id_{\mathbb{R}^n}$ the identity matrix on $\mathbb{R}^n$ and more generally Id_X the identity matrix of any finite-dimensional vector space X. The reader may verifies that the complex multiplication of a vector by the complex value α is equivalent to the matrix-multiplication by the matrix

$$\begin{bmatrix} \alpha & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \alpha \end{bmatrix} = r\alpha \; id_{\mathbb{C}^n} + j \; i\alpha \; id_{\mathbb{C}^n}$$

in the complex description, or by the matrix

$$\mathcal{R}(\alpha) \coloneqq \begin{bmatrix} r\alpha \cdot id_{\mathbb{R}^n} & -i\alpha \cdot id_{\mathbb{R}^n} \\ i\alpha \cdot id_{\mathbb{R}^n} & r\alpha \cdot id_{\mathbb{R}^n} \end{bmatrix}$$

in the real description. This defines the real- and complex-description of a complex value. In particular, the multiplication by j (the square root of -1) is represented by the multiplication by the matrix

$$\begin{bmatrix} 0 & -id_{\mathbb{R}^n} \\ id_{\mathbb{R}^n} & 0 \end{bmatrix}$$

in the real description. We finally notice that any real matrix of the form

$$\begin{bmatrix} rA & -iA \\ iA & rA \end{bmatrix} \in \mathbb{R}^{2m \times 2n}$$

[1.2.1]

defines a map from $\mathbb{R}^{2n}$ to $\mathbb{R}^{2m}$ as well as a map from $\mathbb{C}^n$ to $\mathbb{C}^m$ by taking its complex description. But an arbitrary matrix in $\mathbb{R}^{2m \times 2n}$ is not necessarily of this form and its complex description is not necessarily defined in general. All linear maps we encounter must have a real description of the form [1.2.1] in order to make our algebraic framework valid. In practice, the linear map we encounter will always match that criteria.

It is possible to build a formal theory of the **real** $2n$-dimensional vector space $(\mathbb{C}^n, \mathbb{R}, +\ \cdot)$ i.e. the vector space of n-components complex-valued vectors over the field of real numbers. The set of that vector

space is made of complex valued vectors while the scalars that act on them are purely real numbers. In that vector space is j not a scalar anymore, it is a matrix (a linear map) as described above. If X is a vector space isomorphic to $\mathbb{C}^n$ by the choice of a vector basis $(\overrightarrow{e_1}, \dots, \overrightarrow{e_n})$ (i.e. $X \simeq \mathbb{C}^n$) then the list of the $2n$ vectors $(\overrightarrow{e_1}, \dots, \overrightarrow{e_n}, j\overrightarrow{e_1}, \dots, j\overrightarrow{e_n},)$ is a basis of the $2n$-dimensional vector space X over the field of real numbers and the multiplication by j is a linear map on X (which depends on the choice of the basis). The multiplication by a complex value also becomes a linear map in that real vector space. This $2n$-dimensional vector space over the field of real numbers is in fact our real description of $X \simeq \mathbb{C}^n$. We did not introduce it as formally as a mathematician would do because it is cumbersome and we would not gain any better understanding of the tools we need to describe MRI reconstruction. We leave that exercise to the interested reader.

We finish this section with an example. Let be $\mathbf{\Omega}$ the $n \times n$ matrix of the discrete Fourier transform (DFT) on $\mathbb{C}^n$, which contains the complex-valued trigonometric coefficients. In the real description, this matrix represents an isomorphism from the real vector space $\mathbb{R}^{2n}$ to itself. That may sound counter-intuitive because the matrix $\mathbf{\Omega}$ is complex valued. But in fact, $\mathbf{\Omega}$ can be written in the real description:

$$\mathcal{R}\mathbf{\Omega} = \begin{bmatrix} r\mathbf{\Omega} & -i\mathbf{\Omega} \\ i\mathbf{\Omega} & r\mathbf{\Omega} \end{bmatrix}$$

It can perfectly be described as a real matrix. The DFT of a vector x can be evaluated as

$$\begin{bmatrix} r\mathbf{\Omega} & -i\mathbf{\Omega} \\ i\mathbf{\Omega} & r\mathbf{\Omega} \end{bmatrix} \cdot \begin{bmatrix} rx \\ ix \end{bmatrix} = \begin{bmatrix} r\mathbf{\Omega} \cdot rx - i\mathbf{\Omega} \cdot ix \\ i\mathbf{\Omega} \cdot rx + r\mathbf{\Omega} \cdot ix \end{bmatrix}$$

If on prefers the complex description (for some implementation purpose for example), we can use the complexification of the previous result insetead:

$$(r\mathbf{\Omega} \cdot rx - i\mathbf{\Omega} \cdot ix) + j\,(i\mathbf{\Omega} \cdot rx + r\mathbf{\Omega} \cdot ix) = (r\mathbf{\Omega} + j\, i\mathbf{\Omega}) \cdot (rx + j\, ix) = \mathbf{\Omega}\, x$$

The right-hand side of the previous equation is the complex-description of the DFT and can be evaluated with a common fast-Fourier-transform (FFT) implementation. The DFT can either be seen as acting on $\mathbb{C}^n$ or on $\mathbb{R}^{2n}$. There is nothing wrong in saying the (n-dimensional) DFT with matrix $\mathbf{\Omega}$ is a linear map from the real vector space $\mathbb{R}^{2n}$ to itself.

1.3 Real-valued Euclidean-products, complex-valued Hermitian-products and 2-norms

We present in this subsection the real-valued Euclidean product on $\mathbb{C}^n$. Formally, the following description is the same as considering the vector space $(\mathbb{C}^n, \mathbb{R})$ of complex-valued vectors over the field

of real numbers, and to endow it with a real-valued Euclidean product. But we will here scarify a bit formality to the benefit of simplicity by omitting to speak about the fields of real or complex numbers. It has no consequence on the final results we want to describe. For practical reasons, we will also define the complex-valued Hermitian-product associated to a real-valued Euclidean product.

The round-brackets $(\cdot \mid \cdot)$ will stands for any complex-valued Hermitian-product, or just H-product for short. We recall that a complex-valued Hermitian-product on a complex vector-space X (over the field $\mathbb{C}$) is a function of two variables

$$(\cdot \mid \cdot) : X \times X \longrightarrow \mathbb{C}$$

$$(x, y) \longmapsto (x|y)$$

that verifies for any $\alpha, \beta \in \mathbb{C}$ and for any $x, y, z \in X$

a) $(\alpha\, x + \beta\, y \mid z) = \alpha^*(x \mid z) + \beta^*(y \mid z)$ and $(x \mid \alpha\, y + \beta\, z) = \alpha(x \mid y) + \beta(x \mid z)$

b) $(x|y) = \overline{(y|x)}$

c) $(x|x) \in \mathbb{R}$ and $(x|x) > 0$ whenever $x \neq 0$ and $(x|x) = 0$ for $x = 0$

Property (c) is called positive-definiteness. We note that the choice of defining skew-linearity in the first variable is the physicist convention. The mathematician convention is to define skew-linearity in the second variable.

The triangular brackets $\langle \cdot \mid \cdot \rangle$ will stand for any real-valued Euclidean-product, or just E-product for short. We define here what is a real-valued Euclidean-product on a real vector space X (over the field $\mathbb{R}$). It is a function of two variables

$$\langle \cdot \mid \cdot \rangle : X \times X \longrightarrow \mathbb{R}$$

$$x, y \longmapsto \langle x|y \rangle$$

that verifies for any $\alpha, \beta \in \mathbb{R} \subset \mathbb{C}$ and for any $x, y, z \in X$

a) $\langle \alpha\, x + \beta\, y \,|z\rangle = \alpha\langle x|z\rangle + \beta\langle y|z\rangle$ and $\langle x|\alpha\, y + \beta\, z\,\rangle = \alpha\langle x|y\rangle + \beta\langle x|z\rangle$

b) $\langle x|y\rangle = \langle y|x\rangle$

c) $\langle x|x\rangle \in \mathbb{R}$ and $\langle x|x\rangle > 0$ whenever $x \neq 0$ and $\langle x|x\rangle = 0$ for $x = 0$

For any two vectors $a, b \in \mathbb{R}^{2n}$, the standard E-product of a and b is the real value given by

$$\langle a|b\rangle_{\mathbb{R}^{2n}} \coloneqq a^T \cdot b$$

For any two vectors $a, b \in \mathbb{C}^n$, the standard H-product of a and b is the complex value given by

$$(a|b)_{\mathbb{C}^n} \coloneqq a^* \cdot b = (ra^T \cdot rb + ia^T \cdot ib) + j\,(ra^T \cdot ib - ia^T \cdot rb)$$

For any two vectors $a, b \in \mathbb{C}^n$, the standard E-product of a and b is the real value given by

$$\langle a|b\rangle_{\mathbb{C}^n} \coloneqq real(a^* \cdot b) = ra^T \cdot rb + ia^T \cdot ib = \begin{bmatrix} ra \\ ia \end{bmatrix}^T \cdot \begin{bmatrix} rb \\ ib \end{bmatrix} = \mathcal{R}a^T \cdot \mathcal{R}b = \langle \mathcal{R}a|\mathcal{R}b\rangle_{\mathbb{R}^{2n}}$$

which is nothing else than the standard E-product $\langle \cdot \mid \cdot \rangle_{\mathbb{R}^{2n}}$ on $\mathbb{R}^{2n}$. Or said differently, it is the standard E-product of the vector space we obtain by considering the vector space $(\mathbb{C}^n, \mathbb{R}, +, \cdot)$.

In fact, the real part of any (complex-valued) H-product is always a (real-valued) E-product (while its complex part is a simplectic form). We want to bring to the attention of the reader that the E-product (which is real valued by our definition) will be of interest in our reconstructions, and NOT the H-product (which is complex valued). The vector space of interest in this document is the $2n$-dimensional vector space $\mathbb{C}^n$ (over the field of real numbers).

One of the end-point of this book is the solving convex optimization problems where the domain of the objective function is $\mathbb{R}^{2n} \backsimeq \mathbb{C}^n$. That means that we consider the real- and imaginary-part of the image as independent real decision variables. The notions of convex sets and convex function are naturally defined on real vector spaces with real-valued E-product. As described later in the text, the conjugate-gradient-descent algorithm makes use of a real-valued Euclidean-product in order to create a line-search parameter, which has to be real by nature. That is why the complex-valued H-product do not enter directly in the mathematical framework, but it is really the real-valued E-product enters into consideration.

For the description of our reconstructions, we will need E-products that are more general than the standard ones. For the sake of generality, will consider all existing E-products that are the real part of any H-product on $\mathbb{C}^n$. That means that any of our E-product is the real part of an associated H-product.

Given an E-product $\langle \cdot \mid \cdot \rangle_Y \coloneqq \mathrm{real}(\cdot \mid \cdot)_Y$ with the associated Hermitian-product $(\cdot \mid \cdot)_Y$ on a vector space Y, we will make use of the induced 2-norm $\| \cdot \|_{Y,2}$ given by

$$\|y\|_{Y,2} \coloneqq \sqrt{\langle y|y\rangle_Y} = \sqrt{(y|y)_Y}$$

Both the H-product or E-product can be used to compute the 2-norm because they coincide when both arguments in the product are identical.

The standard 2-norm on $\mathbb{C}^n$ is

$$\|x\|_{\mathbb{C}^n,2} = \sqrt{\sum_{k=1}^{n} |rx_k|^2 + |ix_k|^2}$$

whereas the standard 2-norm on $\mathbb{R}^{2n}$ is

$$\|\mathcal{R}x\|_{\mathbb{R}^{2n},2} = \sqrt{\sum_{k=1}^{2n} |(\mathcal{R}x)_k|^2} = \sqrt{\sum_{k=1}^{n} |rx_k|^2 + |ix_k|^2} = \|x\|_{\mathbb{C}^n,2}$$

1.4 The matrices associated to a H-product and an E-product

For any H-product $(\cdot \mid \cdot)$ on $\mathbb{C}^n$ there exist a unique matrix H so that

$$(a|b) = a^* \, H \, b$$

Matrix H is then positive-definite and is Hermitian (i.e. $H^* = H$).

The converse is also true: if H is any hermitian positive-definite matrix, then previous equation defines a H-product. In term or real and imaginary decomposition $H = rH + j\, iH$ with real-parts rH and imaginary-part iH, the positive-definiteness of H together with Hermiticity are equivalent to

$$rH^T = rH \qquad rH \text{ is positive-definite} \qquad iH^T = -iH$$

Each hermitian product is thus associated to a unique matrix. We will say that H is the matrix of $(\cdot \mid \cdot)$ and that the former is a H-product with matrix H.

For any E-product $\langle \cdot \mid \cdot \rangle$ on $\mathbb{C}^n$ there is a unique matrix S so that

$$\langle a|b\rangle = \mathcal{R}a^T \cdot S \cdot \mathcal{R}b$$

and it follows from the properties of the E-product that S is a real, symmetric and positive-definite.

Conversely, any real, symmetric and positive-definite matrix $S \in \mathbb{R}^{2n\times 2n}$ defines an E-product on $\mathbb{C}^n$ via the previous equation. We say that S is the matrix of $\langle \cdot \mid \cdot \rangle$ and the former is an E-product with matrix S. But all E-products are not necessary the real-part of a H-product.

We will restrict ourselves on E-products that are the real part of a H-product:

$$\langle a|b\rangle = real(a|b)$$

By writing H the matrix of the H-product and writing S the matrix of the E-product, it can be shown that the E-product can be rewritten as

$$\langle a|b\rangle = \begin{bmatrix} ra \\ ia \end{bmatrix}^T \cdot \begin{bmatrix} rH & -iH \\ iH & rH \end{bmatrix} \cdot \begin{bmatrix} rb \\ ib \end{bmatrix} = \mathcal{R}a^T \cdot S \cdot \mathcal{R}b$$

Since

$$(a|b) = a^* \, H \, b$$

it follows that matrix S is given by

$$S = \begin{bmatrix} rH & -iH \\ iH & rH \end{bmatrix} = \begin{bmatrix} rH & iH^T \\ iH & rH \end{bmatrix} = \mathcal{R}H$$

which is a real, symmetric, and positive-definite matrix. We have thus

$$\boxed{S = \mathcal{R}H}$$

This stresses the important fact that an E-product is the real part of a H-product exactly if its matrix S is the real representation of the matrix H of the H-product. In particular, the H-product associated to an E-product is always unique.

We will say that the H-product $(\cdot \mid \cdot)$ is associated to the E-product $\langle \cdot \mid \cdot \rangle$ and inversely. As we will see in the next sub-chapter, this restriction makes the H-adjoint of any matrix A coincide with its E-adjoint, which is a key feature of the algebraic mechanic we need.

1.5 Adjoint-homomorphism and Adjoint-matrix

We now consider a homomorphism with matrix $A \in \mathbb{C}^{m\times n}$ from vector space $X \simeq \mathbb{C}^n$ to vector space $V \simeq \mathbb{C}^m$:

$$A : X \longrightarrow V$$

$$x \longmapsto Ax$$

We assume that X is endowed with a H-product $(\cdot \mid \cdot)_X$ with matrix H_X and we assume that V is endowed with a H-product $(\cdot \mid \cdot)_V$ with matrix H_V. Then it can be shown that there exists exactly one homomorphism of matrix $A_H^\dagger$

$$A_H^\dagger : V \longrightarrow X$$

$$v \longmapsto A_H^\dagger \cdot v$$

such that for any $x \in X$ and $v \in V$ holds

$$(Ax|v)_V = \left(x\middle|A_H^\dagger v\right)_X$$

and it can be shown that the matrix $A_H^\dagger$ is uniquely given by

$$A_H^\dagger = H_X^{-1} \cdot A^* \cdot H_V$$

The matrix $A_H^\dagger$ is the adjoint of matrix A with respect to $(\cdot \mid \cdot)_X$ and $(\cdot \mid \cdot)_V$ and the homomorphism associated to $A_H^\dagger$ is called the adjoint homomorphism.

Let us furthermore endow X with the E-product $\langle \cdot \mid \cdot \rangle_X = \mathrm{real}(\cdot \mid \cdot)_X$ and V with the E-product $\langle \cdot \mid \cdot \rangle_V = \mathrm{real}(\cdot \mid \cdot)_V$. Then it can be shown that there exists exactly one homomorphism of matrix $A_E^\dagger$

$$A_E^\dagger : V \longrightarrow X$$

$$v \longmapsto A_E^\dagger v$$

such that for any $x \in X$ and $v \in V$ holds

$$\langle Ax|v\rangle_V = \left\langle x\middle|A_E^\dagger v\right\rangle_X$$

The matrix $A_E^\dagger$ is the adjoint of matrix A with respect to $\langle \cdot \mid \cdot \rangle_X$ and $\langle \cdot \mid \cdot \rangle_V$.

Because $\langle \cdot \mid \cdot \rangle_X$ is the real part of $(\cdot \mid \cdot)_X$ and $\langle \cdot \mid \cdot \rangle_V$ is the real part of $(\cdot \mid \cdot)_V$, it can be shown that the matrices $A_E^\dagger$ and $A_H^\dagger$ are equal and we will write it $A^\dagger$:

$$A^\dagger := A_E^\dagger = A_H^\dagger$$

The adjoint matrix $A^\dagger$ is thus given by

$$A^\dagger = H_X^{-1} \cdot A^* \cdot H_V$$

Adjoint matrix in the complex description

In the real representation, this translates to

$$\mathcal{R}(A^\dagger) = S_X^{-1} \cdot \mathcal{R}A^T \cdot S_V$$

Adjoint matrix in the real description

The symbol $A^\dagger$ is pronounced "A **adjoint**" or "A **dagger**" or "**dagger** A".

We finally note that for a given homomorphism A, its adjoint homomorphism $A^\dagger$ is independent of any choice of basis. It is the unique homomorphism verifying

$$(Ax|v)_V = \left(x\middle|A^\dagger\, v\right)_X$$

for any vector $v \in V$ and $x \in X$. Homomorphisms A and $A^\dagger$ can therefore be written free of any matrix.

1.6 Gradient and squared-norm functions

Given a real-valued function

$$f(\cdot) : \mathbb{C}^n \longrightarrow \mathbb{R}$$

$$x = rx + j\, ix \longmapsto f(rx + j\, ix)$$

we will say that $f(\cdot)$ is differentiable if and only if the two functions from $\mathbb{R}^n$ to $\mathbb{R}$ given by

$$rx \longmapsto f(rx + j\, ix) \qquad \text{for any fixed value } ix$$

$$ix \longmapsto f(rx + j\, ix) \qquad \text{for any fixed value } rx$$

are differentiable in the sense of real differentiation.

Given a real-valued function $f(\cdot)$ defined on $\mathbb{C}^n$, we consider its real representation $\mathcal{R}f(\cdot)$ defined as

$$\mathcal{R}f(\cdot) : \mathbb{R}^{2n} \longrightarrow \mathbb{R}$$

$$(rx, iy) \mapsto \mathcal{R}f(rx, ix) := f(x) = f(rx + j\, ix)$$

The function $\mathcal{R}f(\cdot)$ is practically the same as the function $f(\cdot)$. We only reformulated the definition domain. It follows that $f(\cdot)$ is differentiable if and only if $\mathcal{R}f(\cdot)$ is differentiable.

Let $f(\cdot)$ be a differentiable function. We define the gradient of $f(\cdot)$ in x as the unique vector $grad_x(f)$ in $X \simeq \mathbb{C}^n$ that satisfies for any $h \in X$

$$\langle grad_x(f)|h\rangle_X = \lim_{\substack{\eta \to 0 \\ \eta \in \mathbb{R}}} \frac{f(x + \eta \cdot h)}{\eta}$$

Note that $\eta \in \mathbb{R}$ holds in the entire section. Using the chain rule we can write

$$\lim_{\eta \to 0} \frac{f(x + \eta \cdot h) - f(x)}{\eta} = \lim_{\eta \to 0} \frac{\mathcal{R}f(rx + \eta \cdot rh, ix + \eta \cdot ih) - \mathcal{R}f(rx, ix)}{\eta}$$
$$= \sum_{v=1}^{n} \frac{\partial \mathcal{R}f}{\partial rx_v} \cdot rh_v + \frac{\partial \mathcal{R}f}{\partial ix_v} \cdot ih_v$$

We define the nabla operators ∇_{rx} (pronounced "**nabla are x**") and ∇_{ix} (pronounced "**nabla i x**") as the column vector operators

$$\nabla_{rx} = \begin{bmatrix} \frac{\partial}{\partial rx_1} \\ \vdots \\ \frac{\partial}{\partial rx_n} \end{bmatrix} \quad \text{and} \quad \nabla_{ix} = \begin{bmatrix} \frac{\partial}{\partial ix_1} \\ \vdots \\ \frac{\partial}{\partial ix_n} \end{bmatrix}$$

It follows

$$\lim_{\eta \to 0} \frac{f(x + \eta \cdot h) - f(x)}{\eta} = \nabla_{rx}\mathcal{R}f^T \cdot rh + \nabla_{ix}\mathcal{R}f^T \cdot ih = \begin{bmatrix} \nabla_{rx}\mathcal{R}f \\ \nabla_{ix}\mathcal{R}f \end{bmatrix}^T \cdot \begin{bmatrix} rh \\ ih \end{bmatrix}$$
$$= \begin{bmatrix} \nabla_{rx}\mathcal{R}f \\ \nabla_{ix}\mathcal{R}f \end{bmatrix}^T \cdot S_X^{-1} \cdot S_X \cdot \begin{bmatrix} rh \\ ih \end{bmatrix}$$

Because S_X is symmetric, S_X^{-1} is symmetric too. It follows

$$\lim_{\eta \to 0} \frac{f(x + \eta \cdot h) - f(x)}{\eta} = \left(S_X^{-1} \cdot \begin{bmatrix} \nabla_{rx}\mathcal{R}f \\ \nabla_{ix}\mathcal{R}f \end{bmatrix}\right)^T \cdot S_X \cdot \begin{bmatrix} rh \\ ih \end{bmatrix}$$

We define

$$\nabla f := \nabla_{rx}\mathcal{R}f + j\, \nabla_{ix}\mathcal{R}f$$

and we will pronounce it "**nabla** f". In order to stress that the differenciation is done with respect to x we will sometimes write $\nabla_x f$ and pronounce it "**nabla X** f".

From

$$S_X^{-1} = \mathcal{R}(H_X^{-1}) \qquad \text{and} \qquad \begin{bmatrix} \nabla_{rx}\mathcal{R}f \\ \nabla_{ix}\mathcal{R}f \end{bmatrix} = \mathcal{R}(\nabla f)$$

we obtain

$$\lim_{\eta \to 0} \frac{f(x + \eta \cdot h) - f(x)}{\eta} = \left(\mathcal{R}(H_X^{-1}) \cdot \ \mathcal{R}(\nabla f)\right)^T \cdot S_X \cdot \mathcal{R}(h) = \mathcal{R}(H_X^{-1} \cdot \nabla f)^T \cdot S_X \cdot \mathcal{R}(h)$$
$$= \langle H_X^{-1} \cdot \nabla f \,|\, h \rangle_X$$

Since that equation is true for any h it follows that the gradient is given by

$$grad_x(f) = H_X^{-1} \cdot \nabla f$$

Gradient in the Complex-description

or in the real representation

$$\mathcal{R}(grad_x(f)) = S_X^{-1} \cdot \begin{bmatrix} \nabla_{rx}\mathcal{R}f \\ \nabla_{ix}\mathcal{R}f \end{bmatrix}$$

Gradient in the Real-description

By the previous construction, the gradient always exists and is always unique. Sometimes, the definition of the gradient is the transpose of the Jacobian matrix, or the Jacobian matrix itself. We note that our definition of the gradient differs from those definitions. Our modified definition takes into account the E-product on X. This has the following two advantages.

It can be shown that the negative gradient $-grad_x(f)$ is parallel to the normalized direction of steepest descent. In fact, the normalized direction of steepest descent is defined by

$$\underset{p \in X}{\operatorname{argmin}} \left\{ \lim_{\eta \to 0} \frac{f(x + \eta \cdot p)}{\eta} \ s.t. \ \|p\|_{X,2} = 1 \right\}$$

which can be rewritten in the real description as

$$\underset{[rp,ip]^T}{\operatorname{argmin}} \left\{ \begin{bmatrix} \nabla_{rx}\mathcal{R}f \\ \nabla_{ix}\mathcal{R}f \end{bmatrix}^T \cdot \begin{bmatrix} rp \\ ip \end{bmatrix} \ s.t. \begin{bmatrix} rp \\ ip \end{bmatrix}^T \cdot S_X \cdot \begin{bmatrix} rp \\ ip \end{bmatrix} = 1 \right\}$$

The solution to this optimization problem is given in [1] on page 476:

$$\begin{bmatrix} rp \\ ip \end{bmatrix} = -\frac{S_X^{-1} \cdot \begin{bmatrix} \nabla_{rx}\mathcal{R}f \\ \nabla_{ix}\mathcal{R}f \end{bmatrix}}{\sqrt{\begin{bmatrix} \nabla_{rx}\mathcal{R}f \\ \nabla_{ix}\mathcal{R}f \end{bmatrix}^T \cdot S_X^{-1} \cdot \begin{bmatrix} \nabla_{rx}\mathcal{R}f \\ \nabla_{ix}\mathcal{R}f \end{bmatrix}}} = -\frac{\mathcal{R}(grad_x(f))}{\|grad_x(f)\|_{X,2}}$$

The normalized steepest descent direction is thus

$$p = -\frac{grad_x(f)}{\|grad_x(f)\|_{X,2}}$$

We now consider a vector space $X \simeq \mathbb{C}^n$ with E-product $\langle \cdot \mid \cdot \rangle_X$ and associated H-product $(\cdot \mid \cdot)_X$ with matrix H_X. We also consider a second vector space $Y \simeq \mathbb{C}^m$ with E-product $\langle \cdot \mid \cdot \rangle_Y$ and associated H-product $(\cdot \mid \cdot)_Y$ with matrix H_Y. We consider a homomorphism, which has matrix M, given by

$$M : X \longrightarrow Y$$

$$x \longmapsto Mx$$

and we consider $y \in Y$. The real-valued squared-norm function

$$x \longmapsto \frac{1}{2}\| Mx - y\|_{Y,2}^2$$

is differentiable and its gradient is given by

$$grad_x(f) = H_X^{-1} M^* H_Y (Mx - y) \in X$$

where the expression for the adjoint can be substituted in order to obtain

$$grad_x(f) = M^\dagger (Mx - y) \in X$$

Gradient of the squared-norm function

In addition, we consider a further vector space $Z \simeq \mathbb{C}^p$ with E-product $\langle \cdot \mid \cdot \rangle_Z$ and associated H-product $(\cdot \mid \cdot)_Z$ with matrix H_Z. We also consider a homomorphism of matrix ϕ given by

$$\phi : X \longrightarrow Z$$

$$x \longmapsto \phi\, x$$

and a vector $z \in Z$. Then the two-terms squared-norm function

$$x \longmapsto \frac{1}{2}\| M\, x - y\|_{Y,2}^2 + \frac{1}{2}\| \phi\, x - z\|_{Z,2}^2$$

is differentiable and its gradient is

$$grad_x(f) = M^\dagger(Mx - y) + \phi^\dagger(\phi\, x - z) \in X$$

This establishes the form of gradient we will encounter in the conjugate gradient-descent algorithm that is part of the ADMM-algorithm presented later.

Our proposed alternative definition of the gradient has therefore two advantages: the negative gradient becomes parallel to the normalized steepest descent direction, and the gradient of a squared norm function becomes similar to the familiar expression

$$M^*(Mx - y)$$

which is the expression for the gradient in the case where the Euclidean products are canonical.

1.7 Orthogonal decompositions and orthogonal projections

Given an arbitrary vector space W with E-product $\langle \cdot \mid \cdot \rangle_W$ with its induced 2-norm $\|\cdot\|_{W,2}$ and given a non-empty, closed, convex set $C \subseteq W$, we define the projection $\Pi_C(w)$ of any $w \in W$ onto C by

$$\Pi_C(w) = \underset{w' \in C}{\operatorname{argmin}} \frac{1}{2}\|w' - w\|_{W,2}^2$$

The existence and uniqueness of that projection is a result of convex analysis.

If P is a vector subspace of W, the vector space W can then be written as the direct sum of P and its orthogonal complement $P^\perp$, which is defined with respect to the E-product $\langle \cdot \mid \cdot \rangle_W$:

$$W = P \oplus P^\perp$$

It can be shown that

$$(P^\perp)^\perp = P$$

for any subspace P. Any vector $w \in W$ can be uniquely decomposed as the orthogonal decomposition

$$w = w_{||} + w_{\perp}$$

with $w_{||} \in P$ and $w_{\perp} \in P^{\perp}$. The vector $w_{||}$ is called the "orthogonal projection" of w onto P and $w_{\perp}$ is called the "orthogonal projection" of w onto $P^{\perp}$. It can be shown that

$$\|w\|_{W,2}^2 = \left\|w_{||}\right\|_{W,2}^2 + \|w_{\perp}\|_{W,2}^2$$

It also holds

$$w_{||} = \Pi_P(w) \quad \text{and} \quad w_{\perp} = \Pi_{P^{\perp}}(w)$$

If C is an affine subspace of W, then there is a unique subspace P of W such that

$$C = c + P = \{c + p \mid p \in P\}$$

for any $c \in C$. For any two c_1 and c_2 in C holds thus

$$c_2 - c_1 \in P$$

Let us chose a fixed $c \in C$ and let be the following orthogonal decomposition of c with respect to P and $P^{\perp}$:

$$c = c_{||} + c_{\perp}$$

with $c_{||} \in P$ and $c_{\perp} \in P^{\perp}$. Since $-c_{||} \in P$ it follows

$$c + \left(-c_{||}\right) = c_{\perp} \in P$$

and therefore

$$C = c_{\perp} + P$$

The set C can be seen as the shift of subspace P in a direction that is orthogonal to P. The vector $c_{\perp}$ is in fact the only vector of W that is both in C and in $P^{\perp}$. In order to see that, we assume that $c_{1,\perp}$ and $c_{2,\perp}$ are both in C and in $P^{\perp}$. Then is

$$c_{1,\perp} - c_{2,\perp} \in P \qquad \text{and} \qquad c_{1,\perp} - c_{2,\perp} \in P^{\perp}$$

hence

$$c_{1,\perp} - c_{2,\perp} = 0$$

Therefore is $c_\perp$ the only vector that is both in C and P, independently of the choice of c.

Let be $w \in W$ and let be C an affine set given by

$$C = c_\perp + P$$

with associated subspace P and $c_\perp$ as defined above. Let be the orthogonal decomposition of w with respect to P and $P^\perp$ given by

$$w = w_{||} + w_\perp$$

We define the orthogonal projection of any $w \in W$ onto C to be equal to $w_{||} + c_\perp$. It can be shown that this orthogonal projection is also given by $\Pi_C(w)$:

$$\Pi_C(w) = w_{||} + c_\perp$$

1.8 The decomposition associated to a homomorphism and the invertible restriction of a homomorphism

In this section, we don't assume any pre-established choice of basis on the vector spaces. Any homomorphism is thus free of matrix and any vector free of coordinates.

Let be A an arbitrary homomorphism between the vector space X with E-product $\langle \cdot \mid \cdot \rangle_X$ and the vector space V with E-product $\langle \cdot \mid \cdot \rangle_V$:

$$A : X \longrightarrow V$$

$$x \longmapsto Ax$$

The adjoint homomorphism $A^\dagger$ is therefore the unique linear map

$$A^\dagger : V \longrightarrow X$$

$$v \longmapsto A^\dagger v$$

such that

$$\langle v|Ax\rangle_V = \left\langle A^\dagger v \middle| x \right\rangle_X \quad \forall v \in V, \forall\, x \in X$$

We write $W^\perp$ for the orthogonal complement of a subspace W with respect to the given E-product in its parent space. It is a fundamental result from linear algebra, sometimes called the decomposition associated to A, that the following is always true:

$$Im(A^\dagger) = Ker(A)^\perp \quad \text{and} \quad Im(A) = Ker(A^\dagger)^\perp$$

Because $(W^\perp)^\perp = W$ for any sub space W, it also holds

$$Im(A^\dagger)^\perp = Ker(A) \quad \text{and} \quad Im(A)^\perp = Ker(A^\dagger)$$

We now define the "restricted spaces"

$$\overline{X} := Im(A^\dagger) = Ker(A)^\perp \subseteq X$$

and

$$\overline{V} := Im(A) = Ker(A^\dagger)^\perp \subseteq V$$

This very important situation is depicted in figure1.

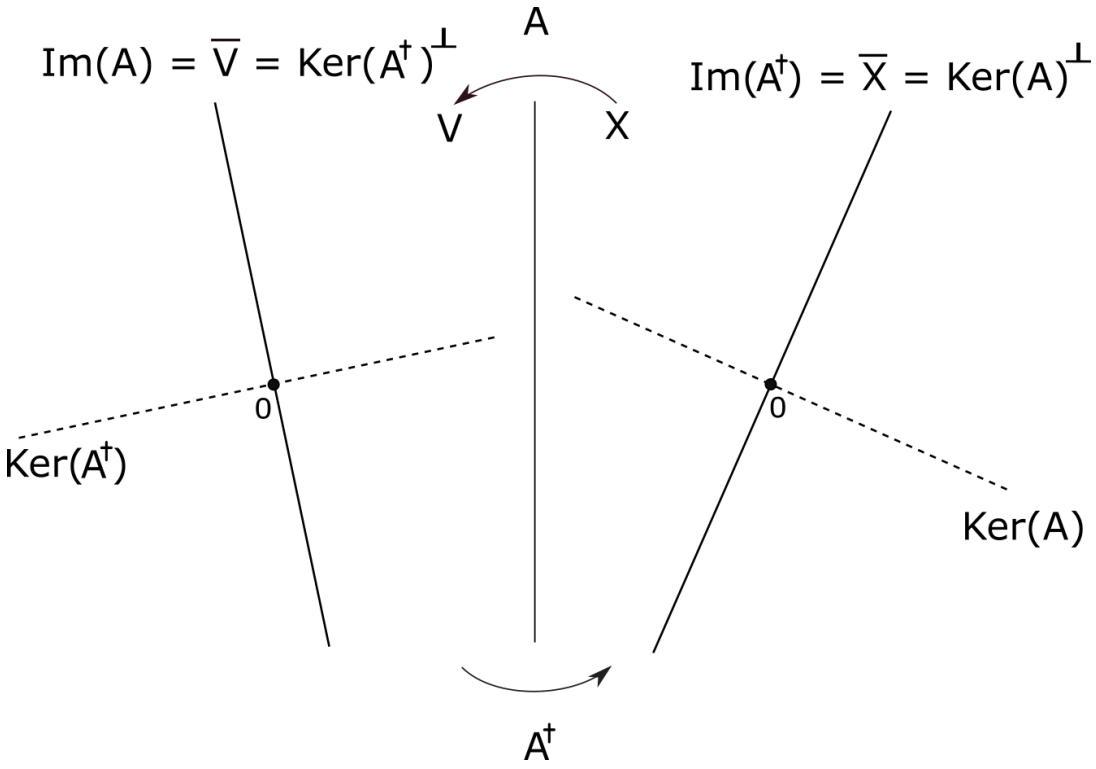

Figure 1: A is a linear map from X to V and $A^\dagger$ is its adjoint from V to X. $Im(A)$ is orthogonal to $Ker(A^\dagger)$ and $Im(A^\dagger)$ is orthogonal to $Ker(A)$. The invertible restriction $\overline{A}$ is an isomorphism between $\overline{X}$ and $\overline{V}$.

We define the homomorphism $\overline{A}$ from $\overline{X}$ to $\overline{V}$ as the restriction of A to $\overline{X}$ and $\overline{V}$:

$$\overline{A} : \overline{X} \longrightarrow \overline{V}$$

$$\bar{x} \longmapsto \overline{A}\,\bar{x} := A\bar{x} \quad \forall \bar{x} \in \overline{X} \subseteq X$$

It is also a fundamental result from linear algebra that $\overline{A}$ is an invertible homomorphism (i.e. an isomorphism) from $\overline{X}$ to $\overline{V}$ and we will call $\overline{A}$ the "invertible restriction" of A. This can be seen as a restatement of the well-known rank-theorem.

We note that

$$\overline{V} = Im(A) = Im(\overline{A})$$

Since $A^\dagger$ is a linear map from V to X in its own right, it also have an invertible restriction $\overline{A^\dagger}$ from $Ker(A^\dagger)^\perp = \overline{V}$ to $Im\left(A^\dagger\right) = \overline{X}$. It follows that

$$\overline{A^\dagger} : \overline{V} \longrightarrow \overline{X}$$

$$\bar{v} \longmapsto \overline{A^\dagger}\bar{v} := A^\dagger\bar{v} \quad \forall \bar{v} \in \overline{V} \subseteq V$$

is also an isomorphism. We note that

$$\overline{X} = Im\left(A^\dagger\right) = Im(\overline{A^\dagger})$$

We now want to describe the adjoint of the invertible restriction $\overline{A}^\dagger = (\overline{A})^\dagger$. For that purpose, we consider the E-products on $\overline{X}$ and $\overline{V}$ naturally inherited from $\langle\cdot\,|\,\cdot\rangle_X$ and $\langle\cdot\,|\,\cdot\rangle_V$. We define the E-product on $\overline{X}$ by

$$\langle\overline{x_1}|\overline{x_2}\rangle_{\overline{X}} := \langle\overline{x_1}|\overline{x_2}\rangle_X \quad \forall\, \overline{x_1}, \overline{x_2} \in \overline{X} \subseteq X$$

and we define the E-product on $\overline{V}$ by

$$\langle\overline{v_1}|\overline{v_2}\rangle_{\overline{V}} := \langle\overline{v_1}|\overline{v_2}\rangle_V \quad \forall\, \overline{v_1}, \overline{v_2} \in \overline{V} \subseteq V$$

The key statement about the adjoint of the invertible restriction $\overline{A}^\dagger$is, that it is equal to the invertible restriction of the adjoint $\overline{A^\dagger}$. The demonstration is as follows. By the definition of the adjoints, which always exist and are always unique, it holds for any $\bar{v} \in \overline{V}$ and $\bar{x} \in \overline{X}$

$$\left\langle\overline{A^\dagger}\bar{v}\middle|\bar{x}\right\rangle_{\overline{X}} = \left\langle\overline{A^\dagger}\bar{v}\middle|\bar{x}\right\rangle_X = \left\langle A^\dagger\bar{v}\middle|\bar{x}\right\rangle_X = \langle\bar{v}|A\bar{x}\rangle_V = \langle\bar{v}|\overline{A}\,\bar{x}\rangle_V = \langle\bar{v}|\overline{A}\,\bar{x}\rangle_{\overline{V}} = \left\langle\overline{A}^\dagger\bar{v}\middle|\bar{x}\right\rangle_{\overline{X}}$$

It follows

$$\overline{A}^{\dagger} = \overline{A^{\dagger}}$$

In particular is $\overline{A}^{\dagger}$ therefore invertible and it holds

$$\overline{A}^{\dagger}\overline{v} = A^{\dagger}\overline{v} \quad \forall \overline{v} \in \overline{V}$$

The bar-symbol on vectors such as $\overline{v}$ has been used until now in order to designate a vector of the restricted space such as $\overline{V}$. We now give the meaning of an operator to the bar $\overline{\cdot}$.

For any $x \in X$ we define the orthogonal decomposition with respect to $\mathrm{Ker}(A) = \overline{X}^{\perp}$ and its orthogonal complement $\overline{X}$ as

$$x = \overline{x} + x_{\perp}$$

where $\overline{x} \in \overline{X}$ and $x_{\perp} \in \overline{X}^{\perp}$ are uniquely determined. The vector $\overline{x}$ is the orthogonal projection of x on $\overline{X}$ and $x_{\perp}$ is the orthogonal projection of x onto $\mathrm{Ker}(A) = \overline{X}^{\perp}$.

For any $v \in V$ we define the orthogonal decomposition with respect to $\mathrm{Ker}\left(A^{\dagger}\right) = \overline{V}^{\perp}$ and its orthogonal complement $\overline{V}$ as

$$v = \overline{v} + v_{\perp}$$

where $\overline{v} \in \overline{V}$ and $v_{\perp} \in \overline{V}^{\perp}$ are uniquely determined. The vector $\overline{v}$ is the orthogonal projection of v on $\overline{V}$ and $v_{\perp}$ is the orthogonal projection of v onto $\mathrm{Ker}\left(A^{\dagger}\right) = \overline{V}^{\perp}$.

It follows immediately that for every $x \in X$ and every $v \in V$ holds

$$A\,x = \overline{A}\,\overline{x} \qquad \text{respectively} \qquad A^{\dagger}v = \overline{A}^{\dagger}\overline{v}$$

Moreover is $Ax \in \mathrm{Im}(A) = \overline{V}$ and is therefore equal to its own orthogonal projection $\overline{Ax}$. The same remark holds for $A^{\dagger}v$ that is equal to $\overline{A^{\dagger}v}$.

We have thus shown the following lemma, that we will use extensively in the subsection about conjugate-gradient-descent algorithm.

Lemma of the invertible restriction

Let be A a homomorphism from X to V. Let be the restricted subspaces $\overline{X}$ and $\overline{V}$ as previously defined. Let be $\overline{A}$ the invertible restriction of A and let be $\overline{A}^{\dagger}$ the invertible restriction of $A^{\dagger}$.

Then, for any $x \in X$ and its orthogonal projection $\overline{x}$ onto $\overline{X}$, respectively for any $v \in V$ and its orthogonal projection $\overline{v}$ onto $\overline{V}$, it holds

$$A\,x = \overline{A}\,\overline{x} = \overline{Ax} \qquad \text{respectively} \qquad A^{\dagger}v = \overline{A}^{\dagger}\overline{v} = \overline{A^{\dagger}v}$$

2. The least-square problem

2.1 The least-square problem and its normal equation

Let V a vector space, $v \in V$ and let be the homomorphism

$$A : X \longrightarrow V$$

$$x \longmapsto Vx$$

We are interested in solving the following least-square problem

$$x^{\#} \in S_{LSQ} \coloneqq \underset{x \in X}{\operatorname{argmin}} \frac{1}{2} \|A\,x - v\|_{V,2}^2$$

Least-Square Problem

The set S_{LSQ} is the set of solutions to this problem. It is the set of minimizers of the squared-norm function

$$x \longrightarrow \frac{1}{2} \|A\,x - v\|_{V,2}^2$$

Squared-Norm Function

The symbol $x^{\#}$ stands for any element of S_{LSQ} i.e. any minimizer of the squared norm function. The least-square problem is to find a minimizer of the squared-norm function. We show as part of the following that the infimum of that optimization problem is always reached i.e. S_{LSQ} is not empty. That problem is thus feasible.

The squared-norm function is differentiable and convex. A necessary and sufficient condition for x to be a solution of least-square problem is therefore

$$\nabla_x \left[\frac{1}{2} \|A\,x - v\|_{V,2}^2 \right] = 0$$

which can be shown to be equivalent to

$$A^* H_V (Ax - v) = 0$$

Multiplying both sides by H_X^{-1} leads

$$H_X^{-1} A^* H_V (Ax - v) = 0$$

and substituting the adjoint of A leads

$$A^{\dagger}(Ax - v) = 0$$

which is equivalent to the normal equation

$$A^{\dagger}Ax = A^{\dagger}v$$

Normal Equation

The solution set S_{LSQ} is therefore equal to the solution set of the normal equation. We now describe this solution set.

We recall that the kernel of A is given by

$$\mathrm{Ker}(A) \coloneqq \{x \in X \mid A\,x = 0\} \subseteq X$$

If it is not equal to $\{0\}$, any solution to the least-square problem (if it exist) is not unique. This follows from the fact that if $n \in \mathrm{Ker}(A)$ and $x^{\#} \in S_{LSQ}$, then is $x^{\#} + n$ a solution too since

$$\frac{1}{2}\left\|A\,(x^{\#} + n) - v\right\|_{V,2}^{2} = \frac{1}{2}\left\|A\,x^{\#} - v\right\|_{V,2}^{2} = \min_{x \in X}\frac{1}{2}\|A\,x - v\|_{V,2}^{2}$$

We recall that the orthogonal projection of v onto $\overline{V} = \mathrm{Im}(A)$ is uniquely given by

$$\overline{v} = \underset{v' \in \mathrm{Im}(A)}{\mathrm{argmin}}\frac{1}{2}\|v' - v\|_{V,2}^{2}$$

If x verifies $A\,x = \overline{v}$ then x verifies

$$x \in \underset{x' \in X}{\mathrm{argmin}}\frac{1}{2}\|A\,x' - v\|_{V,2}^{2}$$

and thus $x \in S_{LSQ}$. Inversely, if $x \in S_{LSQ}$ it must hold $A\,x = \overline{v}$. The solution set S_{LSQ} is therefore the pre-image of $\overline{v}$ by the map A :

$$S_{LSQ} = \{x \in X \,|A\,x = \overline{v}\,\}$$

In particular, the set S_{LSQ} of minimizers is not empty i.e. the minimum is reached.

Moreover, if $x_1^{\#}$ and $x_2^{\#}$ are two solutions, then holds

$$A\,x_1^{\#} = A\,x_2^{\#} = \overline{v}$$

and therefore

$$A\left(x_1^{\#} - x_2^{\#}\right) = 0$$

which means

$$x_1^{\#} - x_2^{\#} \in \mathrm{Ker}(A)$$

The solution set S_{LSQ} is thus given by

$$S_{LSQ} = x^{\#} + \mathrm{Ker}(A)$$

Where $x^{\#}$ is an arbitrary particular solution. There is however a unique solution that is in $\overline{X}$. Let be $x_1^{\#}$ and $x_2^{\#}$ two points which are both in S_{LSQ} and $\overline{X}$. They are then equal to their own orthogonal projection on $\overline{X}$:

$$x_1^{\#} = \overline{x_1^{\#}} \qquad \text{and} \qquad x_2^{\#} = \overline{x_2^{\#}}$$

As we have seen holds

$$x_1^{\#} = x_2^{\#} + n \quad \text{with} \quad n \in \mathrm{Ker}(A)$$

Taking the orthogonal projection on $\overline{X}$ on both sides leads

$$\overline{x_1^{\#}} = \overline{x_2^{\#}}$$

hence

$$x_1^{\#} = x_2^{\#}$$

We will write $\overline{x^{\#}}$ this unique solution which lies both in S and in $\overline{X}$. This notation is well posed since $\overline{x^{\#}}$ is in fact the orthogonal projection onto $\overline{X}$ of any solution $x^{\#} \in S$. It can be shown that it is the unique vector of S_{LSQ} with smallest 2-norm. This situation is described in figure 2.

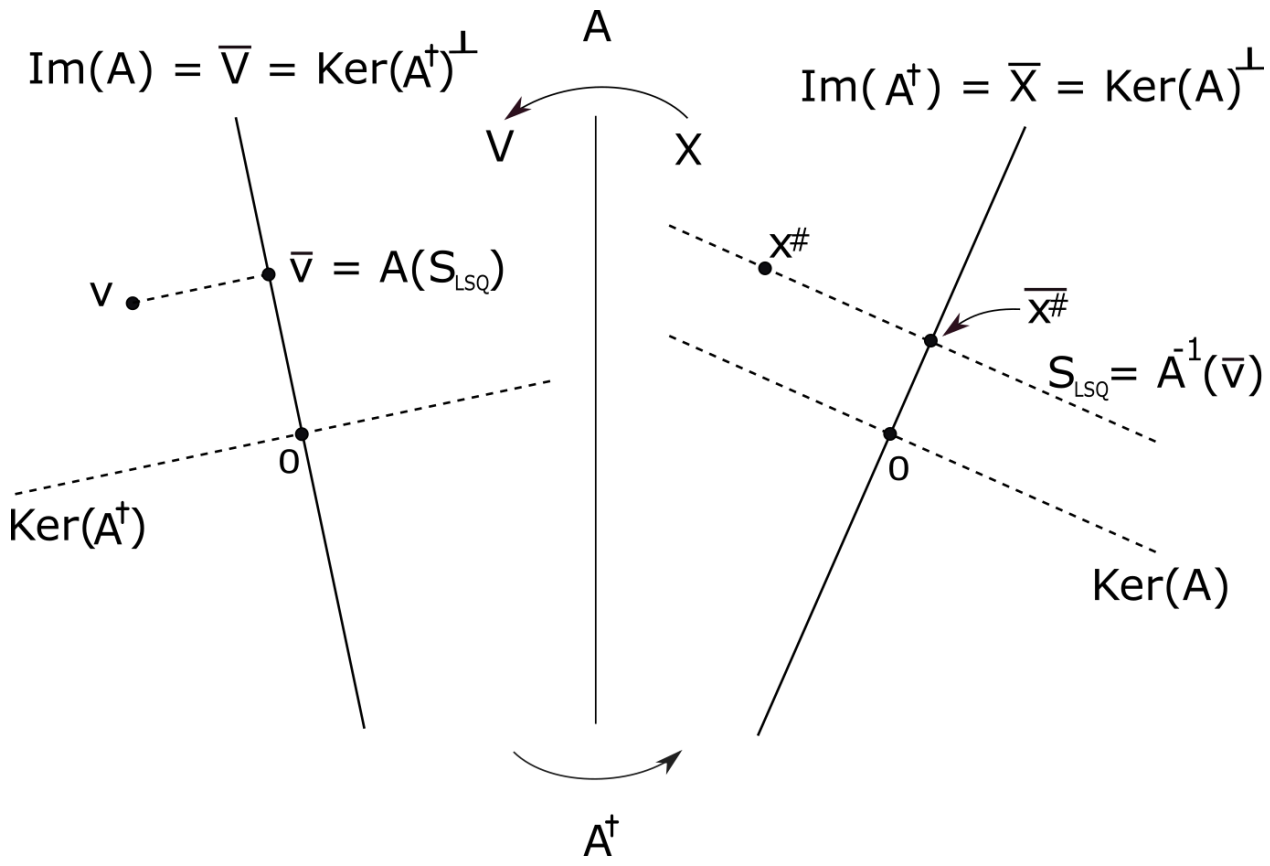

Figure 2: S_{LSQ} is the solution set of the least-square problem. It is equal to $A^{-1}(\bar{v})$, where A^{-1} stands for the set-theoretical inverse of A (i.e. $A^{-1}(\bar{v})$ is the pre-image set of $\bar{v}$ by A). The point $\bar{v}$ is the orthogonal projection of v on $\overline{V}$.

Of note, the solution set of the least-square problem is never empty, while the solution set of the exact equation

$$Ax = v$$ ***Exact Equation***

may be empty. If that equation has a solution, then it is equivalent to the least-square problem and its solution set is equal to S_{LSQ}. This happens if and only if $v \in \mathrm{Im}(A)$.

We will call A^{pinv} the pseudo-inverse (or Moor-Penrose pseudo-inverse) of A:

$$A^{pinv}: V \longrightarrow X$$

$$w \longmapsto A^{pinv} w \,, \forall\, w \in V$$

The existence and uniqueness of A^{pinv} follows from the defining properties of the pseudo-invers. It holds in particular

$$\overline{x^{\#}} = A^{pinv}\, v$$

The solution $\overline{x^{\#}}$ is therefore always related to v by the pseudo inverse of A. We distinguish the following cases:

- In the special case where A has full column-rank (i.e. A is injective in $\mathrm{Im}(A)$, the solution to the least-square problem is unique, all columns of A are linearly independent, the rank of A equals its column-rank) is the unique solution of least-square problem given by

  $$\overline{x^{\#}} = \left(A^{\dagger}A\right)^{-1}A^{\dagger}v$$

  and it holds

  $$A^{pinv} = \left(A^{\dagger}A\right)^{-1}A^{\dagger}$$

- In the special case where A has full raw-rank (i.e. A is surjective on V, there is a solution to the exact equation, all raws of A are linearly independent, the rank of A equals its raw-rank) the solution to the least-square problem with least-2-norm is given by

  $$\overline{x^{\#}} = A^{\dagger}\left(AA^{\dagger}\right)^{-1}v$$

  and it holds

$$A^{pinv} = A^{\dagger}\left(AA^{\dagger}\right)^{-1}$$

- In the special case where A has full rank (i.e. A is bijective on V, there is a unique solution to the exact equation, all raws and columns of A are linearly independent, the rank of A equals its raw-rank and its column-rank) is A invertible and the unique solution to the least-square problem is

$$\overline{x^{\#}} = A^{-1}v$$

and it holds

$$A^{pinv} = A^{-1}$$

- In practice however, we rarely encounter one of these special cases and the pseudo-inverse is given, for example, by the general formulas

$$A^{pinv} = \lim_{\kappa \to 0}\left(A^{\dagger}A + \kappa\, id_X\right)^{-1}A^{\dagger} = \lim_{\kappa \to 0} A^{\dagger}\left(AA^{\dagger} + \kappa\, id_Y\right)^{-1}$$

Informal note*: There is some exceptional cases (we will encounter at least one) where some of the previous formulas can be applied in practice for MRI reconstruction, if A is simple enough. But usually, these expressions are of no practical interest for MRI reconstruction because the linear systems we encounter are usually so big that it is out of question to perform a matrix inversion by an exact method. We will instead approximate the pseudo inverse by a truncated iterative method. In any case, these expressions of the pseudo-inverse remain of theoretical interest and are important to understand some scientific articles.*

Summary:

The set of minimizers of the **squared-norm function**

$$S_{LSQ} = \underset{x \in X}{\operatorname{argmin}} \frac{1}{2} \|A\,x - v\|_{V,2}^2$$

is the solution set of the **least-square problem** and is the pre-image of $\overline{v}$ by the map A. It is a non-empty affine space given by

$$S_{LSQ} = \overline{x^{\#}} + \operatorname{Ker}(A)$$

where $\overline{x^{\#}}$ is the unique solution that belongs to $\overline{X}$, is the least 2-norm vector of S_{LSQ}, and is given by the (Moor-Penrose) pseudo invers.

To the least-square problem, we associate the **normal equation**

$$A^{\dagger}Ax = A^{\dagger}v$$

which solution set is S_{LSQ}, and the **exact equation**

$$Ax = v$$

which solution is S_{LSQ} if and only if $v \in \operatorname{Im}(A)$, and is empty if not.

2.2 The invertible normal equation associated to the least-square problem

Let be the least-square problem

$$x^{\#} \in S_{LSQ} := \underset{x}{\operatorname{argmin}} \frac{1}{2} \|Ax - v\|_{V,2}^2$$

We consider $\overline{v}$ to be the orthogonal projection in of $v \in V$ onto $\overline{V} = \operatorname{Im}(A)$:

$$\overline{v} = \underset{v' \in \overline{V}}{\operatorname{argmin}} \frac{1}{2} \|v - v'\|_{V,2}^2$$

It follows that x is solution of the least-square problem exactly if it is solution of

$$Ax = \overline{v}$$

The solution set of that equation is

$$\overline{x^{\#}} + \operatorname{Ker}(A)$$

Where $\overline{x^{\#}} \in S_{LSQ}$ is the least 2-norm solution.

As defined earlier, the restricted space $\overline{X} \subseteq X$ is the orthogonal complement to $\operatorname{Ker}(A)$:

$$\overline{X} := \operatorname{Ker}(A)^{\perp} \subseteq X$$

And the restricted space $\overline{V} \subseteq V$ is the image of A:

$$\overline{V} := \operatorname{Im}(A)$$

We consider $\overline{A}$, the restriction of A to the sub spaces $\overline{X}$ and $\overline{V}$:

$$\overline{A} : \overline{X} \dashrightarrow \overline{V} \subseteq V$$

$$\overline{x} \longmapsto \overline{A}\,\overline{x}$$

where $\overline{V}$ contains $\overline{v}$, the orthogonal projection v onto $\overline{V}$. We have seen that $\overline{A}$ is invertible from $\overline{X}$ to $\overline{V}$. The invertible exact equation

$$\overline{A}\,\overline{x} = \overline{v}$$

Invertible Exact Equation

has thus a single solution because it is implicitly meaned that $\overline{x}$ is in the definition domain of $\overline{A}$, which is $\overline{X}$. It is straight forward to show that this single solution is $\overline{x^{\#}}$, the lest 2-norm solution of the least-square problem.

We have seen that $\overline{A}^{\dagger}$ is invertible too. The exact invertible equation is therefore fully equivalent to the invertible normal equation

$$\overline{A}^{\dagger}\overline{A}\,\overline{x} = \overline{A}^{\dagger}\overline{v}$$

Invertible Normal Equation

The single solution of that equation is $\overline{x^{\#}}$ and can be written as

$$\overline{x^{\#}} = \overline{A}^{-1}\overline{v}$$

It is also the single solution of the least-square problem which is both in $\overline{X}$ and S_{LSQ}. The solution $\overline{x^{\#}}$ is the orthogonal projection on $\overline{X}$ of any element any $x^{\#} \in S_{LSQ}$.

Because $\overline{A}$ is invertible, the homomorphism $\overline{A}^{\dagger}\overline{A}$ is hermitian and positive-definite, and of course invertible. The invertible normal equation can thus be solved with the conjugate-gradient algorithm (CGD-algorithm) in order to lead the unique solution $\overline{x^{\#}}$ in a finite number of steps $i_{max} \leq \dim(\overline{X}) \leq \dim(X)$. The obtained sequence from the CGD-algorithm can only be written abstractly and cannot be evaluated in practice. The key to solve that issue is to perform the CGD-algorithm with homomorphism $A^{\dagger}A$ instead of $\overline{A}^{\dagger}\overline{A}$ and with data vector v instead of $\overline{v}$. We show in the following section that the result is a solution of the least-square problem of the form

$$x^{\#} = \overline{x^{\#}} + x_{0,\perp}$$

where x_0 is the initial value of the CGD-algorithm and $x_{0,\perp} \in \mathrm{Ker}(A)$ is its orthogonal projection on $Ker(A)$. We also show that the solution $x^{\#}$ obtained by the CGD-method is the orthogonal projection of the initial value x_0 onto the affine solution set $S_{LSQ} = \overline{x^{\#}} + \mathrm{Ker}(A)$.

2.3 Conjugate-gradient-descent method for the least-square problem

In their original article of 1952 [2], Stiefel and Hestenes presented two iterative algorithms (and some variants), both called "conjugate-gradient-descent method" (CGD-method or CGD-algorithm). The authors chose to describe the methods in term of matrices and coordinate vectors but we will use a basis-free description in the following subsection. We will write homomorphism free from any basis and vector free of coordinates.

The first method is to solve the invertible linear equation

$$Qx = q$$

where Q is an homomorphism from a real vector space L to itself (Q is an endomorphism) that is symmetric and positive-definite (and thus invertible), and where $x, q \in L$. The method is described in term of matrices and coordinate vector in $\mathbb{R}^n$ and it is considered that an orthogonal basis on X is used. That means that the norm is the canonical 2-norm $\|\cdot\|_{\mathbb{R}^n,2}$ on $\mathbb{R}^n$ given by

$$\|x\|_{\mathbb{R}^n,2} = \sqrt{\langle x|x\rangle_{\mathbb{R}^n}}$$

where the canonical Euclidean-product $\langle \cdot \mid \cdot \rangle_{\mathbb{R}^n}$ is given by

$$\langle x|x\rangle_{\mathbb{R}^n} = x^T \cdot x$$

and where x is to be read as a coordinate vector lying in $\mathbb{R}^n$. In that context, Q is read as a matrix and its adjoint verifies $Q^\dagger = Q^T$.

However, the method and all demonstrations in that article can be extended in a straight forward way to the case of an arbitrary E-product. We don't need to choose any basis to describe that method. We have simply a vector space L with a 2-norm $\|\cdot\|_{L,2}$ induced by the E-product $\langle\cdot \mid \cdot\rangle_L$. The homomorphism Q can be written free from any basis and the matrix Q^* has to be replaced by the matrix-free homomorphism $Q^\dagger$. All vectors can also written in a basis-free way i.e. without use of coordinates. Moreover, the original description of the CGD-algorithm on a real Euclidean vector spaces is perfectly adapted to our problem since our vector spaces of interest are also real Euclidean vector spaces (and not complex Hermitian). In order to match our naming convention, we will name that method the "CGD-algorithm (for the invertible exact equation)". The first CGD-algorithm presented by Stiefel and Hestenes in their article solves invertible exact equation and reads as follows:

CGD-algorithm (for the invertible exact equation):

INITIALIZE

(a) Choose an $x_0 \in L$

(b) Initialize the residual $\quad r_0 = q - Q\,x_0 \in L$

(c) Initialize the search direction $\quad p_0 = r_0 \in L$

DO FOR i = 0, 1, 2, …

(d) Evaluate the line-search parameter : $\quad a_i = \frac{\|r_i\|_{L,2}^2}{\langle p_i | A\,p_i\rangle_L} \in \mathbb{R}$

(e) Update the approximated solution : $\quad x_{i+1} = x_i + a_i\,p_i \in L$

(f) Update the residual: $\quad r_{i+1} = r_i - a_i\,Q\,p_i \in L$

(g) Update the b-parameter: $\quad b_i = \frac{\|r_{i+1}\|_{L,2}^2}{\|r_i\|_{L,2}^2}$

(h) Update the search direction: $\quad p_{i+1} = r_{i+1} + b_i\,p_i \in L$

UNTIL $\|r_{i+1}\|_{L,2}^2 = 0$

The second method described in the article is a method to solve the special case where the equation to solve is normal, which means

$$Q = B^{\dagger}B \qquad \text{and} \qquad q = B^{\dagger}b$$

for some B invertible from a space L to a linear space R and where $b \in R$ is a vector. It is sated in the article that this second method is fully equivalent to the first method applied to homomorphism Q and vector q, but its advantage is that the original data B, $B^{\dagger}$ and b are used instead of Q and q, which is better suited numerically. This normal equation above becomes equivalent to the invertible normal equation by setting

$$B = \bar{A} \qquad b = \bar{v} \qquad L = \overline{X} \qquad R = \overline{V}$$

The invertible normal equation can thus be solved with the second CGD-algorithm presented in the article. We will therefore call it the "CGD-algorithm (for the invertible normal equation)". It reads as follows:

CGD-algorithm (for invertible normal equation):

INITIALIZE

(i) Choose an $\overline{x_0} \in \overline{X}$

(j) Initialize the residual $\overline{r_0} = \bar{v} - \bar{A}\ \overline{x_0} \in \overline{V}$

(k) Evaluate the search direction $\overline{p_0} = \bar{A}^{\dagger}\ \overline{r_0} \in \overline{X}$

DO FOR i = 0, 1, 2, …

(l) Evaluate the line-search parameter : $\overline{a_\iota} = \frac{\|\bar{A}^{\dagger}\bar{r}_\iota\|^2_{\overline{X},2}}{\|\bar{A}\ \overline{p_\iota}\|^2_{\overline{V},2}} \in \mathbb{R}$

(m) Update the approximated solution : $\overline{x_{\iota+1}} = \bar{x}_\iota + \bar{a}_\iota\, \overline{p_\iota} \in \overline{X}$

(n) Update the residual: $\overline{r_{\iota+1}} = \bar{r}_\iota - \overline{a_\iota}\ \bar{A}\, \overline{p_\iota} \in \overline{V}$

(o) Update the b-parameter: $\bar{b}_\iota = \frac{\|\bar{A}^{\dagger}\overline{r_{\iota+1}}\|^2_{\overline{X},2}}{\|\bar{A}^{\dagger}\bar{r}_\iota\|^2_{\overline{X},2}}$

(p) Update the search direction: $\overline{p_{\iota+1}} = \bar{A}^{\dagger}\overline{r_{\iota+1}} + \ \bar{b}_\iota\, \overline{p_\iota} \in \overline{V}$

UNTIL $\|\overline{r_{\iota+1}}\|^2_{\overline{V},2} = 0$

By virtue of the CGD-algorithm properties, the above iterations generate sequence

$$\overline{x_0}, \overline{x_1}, \overline{x_2}, \ldots \overline{x_{\iota_{max}}} \qquad \text{with} \qquad \overline{x_{\iota_{max}}} = \overline{x^{\#}} \qquad \text{and} \qquad i_{max} \leq \dim(\overline{X}) \leq \dim(X)$$

We now rewrite this algorithm by replacing all unknown quantities by quantities we can evaluate in practice. We obtain thus a new algorithm that generates a different sequence. For some reason that will become clear in a few lines, we will call it the “CGD-algorithm (for least-square problem)”.

CGD-algorithm (for least-square problem):

INITIALIZE

(a) Choose an $x_0 \in X$

(b) Evaluate the residual $r_0 = v - A\,x_0 \ \in V$

(c) Evaluate the search direction $p_0 = A^\dagger\, r_0 \in X$

DO

(d) Evaluate the line-search parameter : $a_i = \frac{\|A^\dagger r_i\|_{X,2}^2}{\|A\,p_i\|_{V,2}^2} \in \mathbb{R}$

(e) Update the approximated solution : $x_{i+1} = x_i + a_i\,p_i \in X$

(f) Update the residual: $r_{i+1} = \ r_i - a_i\ Ap_i \ \in V$

(g) Update the b-parameter: $b_i = \frac{\|A^\dagger r_{i+1}\|_{X,2}^2}{\|A^\dagger r_i\|_{X,2}^2}$

(h) Update the search direction: $p_{i+1} = A^\dagger r_{i+1} + \ b_i\,p_i \ \in X$

UNTIL $\left\|A^\dagger r_{i+1}\right\|_{X,2}^2 = 0$

Given any $x_0 \in X$, we decompose it as

$$x_0 = x_{0,\perp} + \overline{x_0}$$

and we decompose v as

$$v = v_\perp + \overline{v}$$

We will now show that the sequence defined by the CGD-algorithm (for least-square problem) is as well defined as the sequence defined by the CGD-algorithm (for invertible normal equation) by showing the following theorem:

Theorem:

For all $i = 1, \dots, i_{max}$ holds

$$x_i = \overline{x_\iota} + \ x_{0,\perp}$$
$$r_i = \ \overline{r_\iota} + \ v_\perp$$
$$p_i = \overline{p_\iota}$$
$$a_i = \ \overline{a_\iota}$$
$$b_i = \ \overline{b_\iota}$$

Proof by induction.

We make in this proof an extensive use the "Lemma of the invertible restriction". We begin with $i = 0$.

Step (a) simply consist of an initial choice for x_0 and

$$x_0 = \overline{x_0} + x_{0,\perp}$$

is true by definition.

In step (b), be evaluate

$$r_0 = v - A\, x_0 \in V$$

We note that

$$r_0 = v - A\, x_0 = \overline{v} + v_\perp - A\left(\overline{x_0} + x_{0,\perp}\right) = \overline{v} + v_\perp - \overline{A}\,\overline{x_0} = \overline{r_0} + v_\perp$$

In step (c), we evaluate

$$p_0 = A^\dagger\, r_0 \in X$$

We note that

$$p_0 = A^\dagger\, r_0 = A^\dagger\,(\overline{r_0} + v_\perp) = \overline{A}^\dagger \overline{r_0} = \overline{p_0}$$

We have thus shown for $i = 0$ that

$$x_i = \overline{x_i} + x_{0,\perp} \qquad \text{and} \qquad r_i = \overline{r_i} + v_\perp \qquad \text{and} \qquad p_i = \overline{p_i}$$

We now assume by induction that this is true for a given i.

For step (d) we evaluate

$$a_i = \frac{\left\|A^\dagger\, r_i\right\|_{X,2}^2}{\|A\, p_i\|_{V,2}^2} \in \mathbb{R}$$

We note that

$$\left\|A^\dagger\, r_i\right\|_{X,2}^2 = \left\|A^\dagger\,(\overline{r_i} + v_\perp)\right\|_{X,2}^2 = \left\|\overline{A}^\dagger\, \overline{r_i}\right\|_{X,2}^2 = \left\|\overline{A}^\dagger\, \overline{r_i}\right\|_{\overline{X},2}^2$$

and similarly

$$\|A\,p_i\|_{V,2}^2 = \|A\,\overline{p_i}\|_{V,2}^2 = \|\overline{A}\;\overline{p_i}\|_{V,2}^2 = \|\overline{A}\;\overline{p_i}\|_{\overline{V},2}^2$$

It follows

$$a_i = \overline{a_i}$$

For step (e) we evaluate

$$x_{i+1} = x_i + a_i\,p_i \in X$$

We note that

$$x_{i+1} = x_i + a_i\,p_i = \overline{x_i} + x_{0,\perp} + \overline{a_i}\,\overline{p_i} = \overline{x_{i+1}} + x_{0,\perp}$$

In step (f) we evaluate

$$r_{i+1} = r_i - a_i\;Ap_i \in V$$

We note that

$$r_{i+1} = r_i - a_i\;Ap_i = \overline{r_i} + v_\perp - \overline{a_i}\,\overline{A}\,\overline{p_i} = \overline{r_{i+1}} + v_\perp$$

In step (g) we evaluate

$$b_i = \frac{\left\|A^\dagger r_{i+1}\right\|_{X,2}^2}{\|A^\dagger r_i\|_{X,2}^2}$$

We note that

$$b_i = \frac{\left\|A^\dagger r_{i+1}\right\|_{X,2}^2}{\|A^\dagger r_i\|_{X,2}^2} = \frac{\left\|\bar{A}^\dagger \overline{r_{i+1}}\right\|_{X,2}^2}{\|\bar{A}^\dagger \overline{r_i}\|_{X,2}^2} = \frac{\left\|\bar{A}^\dagger \overline{r_{i+1}}\right\|_{\bar{X},2}^2}{\|\bar{A}^\dagger \overline{r_i}\|_{\bar{X},2}^2} = \overline{b_i}$$

In step (h) we evaluate

$$p_{i+1} = A^\dagger r_{i+1} + b_i\,p_i \in V$$

We note that

$$p_{i+1} = A^\dagger r_{i+1} + b_i\,p_i = \overline{A}^\dagger \overline{r_{i+1}} + \overline{b_i}\,\overline{p_i} = \overline{p_{i+1}}$$

■

Corollary:

Let be A any homomorphism from X to V and let be $v \in V$. Then, the CGD-algorithm (for least square problems) converges to a solution of the least-square problem

$$x^{\#} \in S_{LSQ} = \underset{x \in X}{\operatorname{argmin}} \frac{1}{2} \|Ax - v\|_{V,2}^2$$

in a number of steps $i_{max} \leq \dim(X)$ and it holds

$$x^{\#} = x_{i_{max}} = \overline{x^{\#}} + x_{0,\perp}$$

Corollary:

By setting the initial value $x_0 = 0$, the CGD-algorithm (for least square problems) leads to the solution

$$x^{\#} = \overline{x^{\#}} = A^{pinv} v$$

and realizes therefore the (Moore-Penrose) pseudo-inverse of v.

We note that for the CGD-algorithm (for least square problems), the residual

$$r_i = \overline{r_i} + v_{\perp}$$

is possibly never 0 because of the contribution of $v_{\perp}$. The stopping condition is instead $\left\|A^{\dagger} r_{i+1}\right\|_{X,2}^2 = 0$ which is equivalent to $\|\overline{r_{i+1}}\|_{V,2}^2 = 0$.

We now show that the solution

$$x^{\#} = x_{i_{max}} = \overline{x^{\#}} + x_{0,\perp}$$

is the orthogonal projection of the initial value x_0 onto the solution set $S_{LSQ} = \overline{x^{\#}} + \operatorname{Ker}(A)$, as described in figure 3.

The orthogonal projection of x_0 onto S is then given by

$$\Pi_S(x_0) = \underset{x' \in S}{\operatorname{argmin}} \frac{1}{2}\|x' - x_0\|_{X,2}^2$$

$$= \underset{x' \in S}{\operatorname{argmin}} \frac{1}{2}\left\|x' - \overline{x_0} - x_{0,\perp}\right\|_{X,2}^2$$

$$= \overline{x^{\#}} + \underset{x'_\perp \in Ker(A)}{\operatorname{argmin}} \frac{1}{2}\left\|\overline{x^{\#}} + x'_\perp - \overline{x_0} - x_{0,\perp}\right\|_{X,2}^2$$

$$= \overline{x^{\#}} + \underset{x'_\perp \in Ker(A)}{\operatorname{argmin}} \frac{1}{2}\left\|\overline{x^{\#}} - \overline{x_0}\right\|_{X,2}^2 + \frac{1}{2}\left\|x'_\perp - x_{0,\perp}\right\|_{X,2}^2$$

$$= \overline{x^{\#}} + x_{0,\perp} = x^{\#}$$

We have shown:

Lemma of the orthogonal projection by CGD-algorithm

Given a point $x_0 \in X$, the CGD-algorithm (for least square problems) realizes the orthogonal projection of x_0 on the affine space S_{LSQ} by choosing x_0 as initial value:

$$x^{\#} = \Pi_S(x_0)$$

We highlight finally the link between the search direction p_i and the gradient of the squared-norm function

$$x \longmapsto \frac{1}{2}\|Ax - v\|_{V,2}^2$$

It can be verified that the residuals all verify for all $i = 0, \ldots, i_{max}$

$$r_i = v - Ax_i$$

The search direction p_{i+1} verifies

$$p_{i+1} = A^{\dagger} r_{i+1} + b_i\, p_i$$

The first term is therefore

$$A^{\dagger}r_{i+1} = A^{\dagger}(v - Ax_{i+1}) = -A^{\dagger}(Ax_{i+1} - v) = -grad_x(\frac{1}{2}\|Ax - v\|^2_{\bar{V},2})|_{x=x_{i+1}}$$

and is thus the negative gradient of squared-norm function evaluated in x_{i+1}. The second term is a correction that implies the $A^{\dagger}A$-orthogonality of all search directions

$$\langle A^{\dagger}Ap_i | p_j \rangle_X = 0 \qquad \text{for} \qquad i \neq j$$

hence the name “conjugate-gradient-descent”.

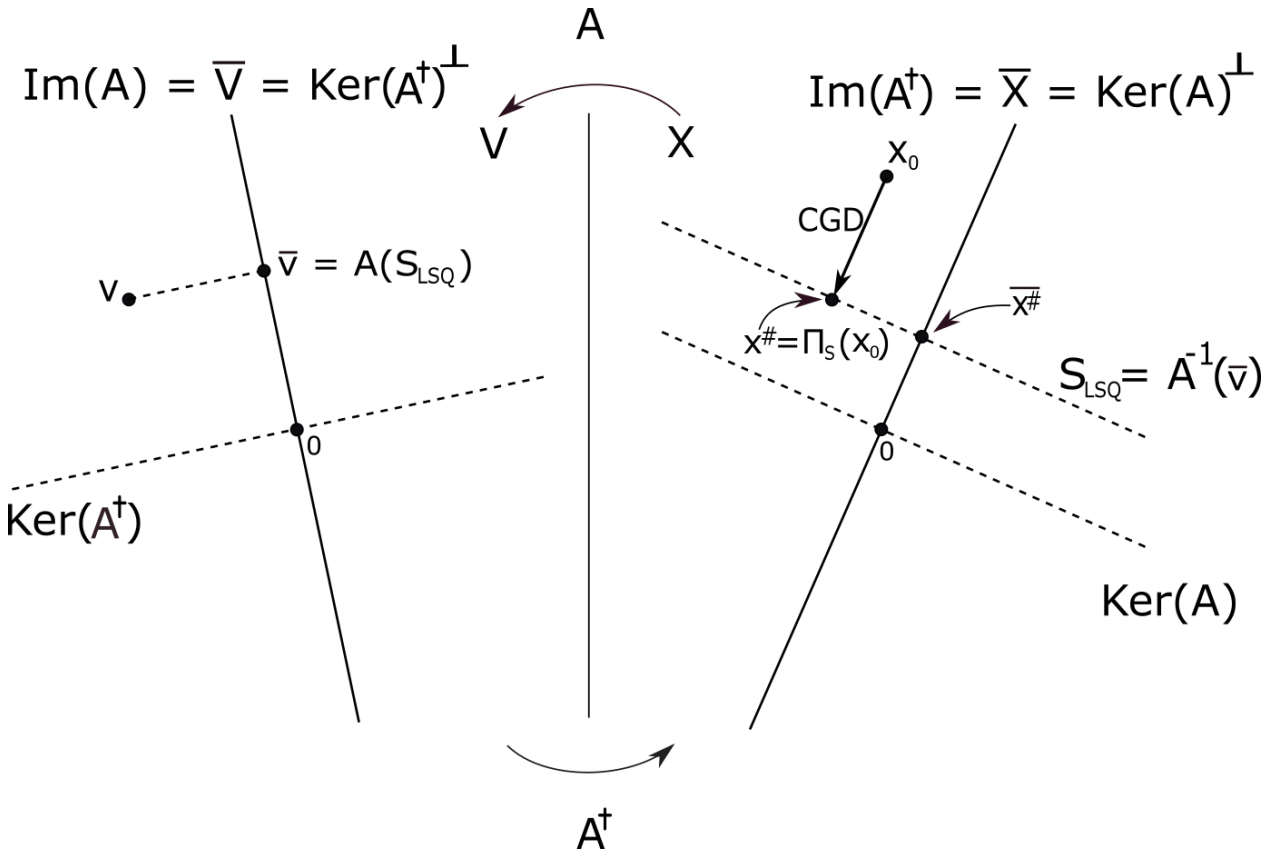

Figure 3: The CGD-algorithm (for least-square problems) takes x_0 as initial value and performs the orthogonal projection of x_0 on the solution set S_{LSQ} of the least-square problem leading to a minimizer $x^{\#}$ of the squared-norm-function. Note that $\overline{x^{\#}}$ is the least-2-norm vector of S. Choosing $x_0 = 0$ leads to the solution $\overline{x^{\#}}$ and the CGD-algorithm realizes in that case the evaluation of the More-Penrose pseudo-inverse in v.

***Informal Note**: In practice, in MRI reconstructions, many implementations of the CGD-algorithm (for least square problems) are actually inexact because the line-search α-parameter is not computed exactly but with inexact methods instead. The author can only hypostatize about the reason for that. But it may be that there is some errors in those implementations that propagate in the computation of the α-parameter and therefore gives a wrong value. A heuristic and inexact evaluation of the α-parameter must then be done in order to restore the descent property of the algorithm. If you allow a point of sarcasm, the persons implementing such wrong algorithms also claim that the CGD-algorithm is simple to implement and can be written in 4 lines. It seems therefore that in the real world, even very simple things must be considered very carefully.*

3. The generalized-LASSO problem

3.1 The 1-norm and the Soft-thresholding

If $z \in \mathbb{C}^n$ we define the 1-norm of z as

$$\|z\|_{\mathbb{C}^n,1} \coloneqq \sum_{k=1}^{n} |rz_k| + |iz_k|$$

For $\mathcal{R}z \in \mathbb{R}^{2n}$ we define the 1-norm of $\mathcal{R}z$ as

$$\|\mathcal{R}z\|_{\mathbb{R}^{2n},1} \coloneqq \sum_{k=1}^{2n} |(\mathcal{R}z)_k| = \sum_{k=1}^{n} |rz_k| + |iz_k| = \|z\|_{\mathbb{C}^n,1}$$

We note that we don't make use of the conventional 1-norm on $\mathbb{C}^n$ given by

$$\sum_{k=1}^{n} |rz_k + j\, iz_k|$$

The definition of the 1-norm makes use of specific coordinates and in thus intrinsically linked to the choice of a vector basis.

For the need of our theory, we define the component dependent weighted 1-norm on a vector space $Z \simeq \mathbb{C}^n$ by

$$\|z\|_{Z,1} \coloneqq \|H_Z\, z\|_{\mathbb{C}^n,1} = \|H_Z\, rz\|_{\mathbb{R}^n,1} + \|H_Z\, iz\|_{\mathbb{R}^n,1}$$

where H_Z is a diagonal matrix of positive weights $\Delta Z_1, \dots, \Delta Z_N$ given by

$$H_Z = \begin{bmatrix} \Delta Z_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Delta Z_n \end{bmatrix}$$

It is therefore a hermitian matrix and is naturally associated to the E-product on Z given by

$$\langle z_1 | z_2 \rangle_Z = real\{z_1^* \cdot H_Z \cdot z_2\}$$

In the ADMM-algorithm presented later, appears the proximal operator associated to the 1-norm. In general, for any real-valued and proper, closed, convex function $f(\cdot)$ defined on a real Hilbert space Z (in particular a real finite dimensional vector space), the proximal operator associated to $f(\cdot)$ is defined by

$$prox_f : Z \longrightarrow Z$$

$$z \longrightarrow prox_f(z) := \underset{w \in Z}{\operatorname{argmin}} f(w) + \frac{1}{2}\|w - z\|_{Z,2}^2$$

For the special case $f(\cdot) = s \cdot \|\cdot\|_{\mathbb{R}^n,1}$ i.e. the standard 1-norm on the real vector space $\mathbb{R}^n$ multiplied by a positive parameter $s > 0$, the corresponding proximal operator

$$prox_{s\cdot\|\cdot\|_{\mathbb{R}^n,1}}(z) = \underset{w \in \mathbb{R}^n}{\operatorname{argmin}} \; s \cdot \|w\|_{\mathbb{R}^n,1} + \frac{1}{2}\|w - z\|_{\mathbb{R}^n,2}^2$$

has a closed form called "**soft-thresholding**" and given component wise by

$$\left(prox_{s\cdot\|\cdot\|_{\mathbb{R}^n,1}}(z)\right)_k = \begin{Bmatrix} z_k - s & : & z_k > s \\ 0 & : & |z_k| \leq s \\ z_k + s & : & z_k < -s \end{Bmatrix} for\ k = 1, \dots, n$$

Soft-Thresholding

We note that in that special case, the 1-norm is the standard 1-norm on $\mathbb{R}^n$ given by

$$\|z\|_{\mathbb{R}^n,1} = \sum_{k=1}^{n} |z_k|$$

and the 2-norm is the standard 2-norm on $\mathbb{R}^n$ given by

$$\|z\|_{\mathbb{R}^n,2} = \sqrt{\sum_{k=1}^{n} z_k^2}$$

In our case however, the ADMM algorithm requires the proximal operator to be define on $Z \simeq \mathbb{C}^n$ which is a complex-valued vector space. Moreover, its 1-norm is given by

$$\|z\|_{Z,1} = \|H_Z\, z\|_{\mathbb{C}^n,1} = \|H_Z\, rz\|_{\mathbb{R}^n,1} + \|H_Z\, iz\|_{\mathbb{R}^n,1}$$

while its 2-norm is given by

$$\|z\|_{Z,2} = \sqrt{real\{z^* \cdot H_Z \cdot z\}}$$

and the associated E-product is

$$\langle z_1 | z_2 \rangle_Z \;=\; real\{z_1^* \cdot H_Z \cdot z_2\}$$

The corresponding proximal operator evaluated on vector $z \in Z \simeq \mathbb{C}^n$ is then

$$\underset{w \in Z}{\operatorname{argmin}}\, s \cdot \|w\|_{Z,1} + \frac{1}{2}\|z - w\|_{Z,2}^2$$

$$= \underset{(rw + j\, iw) \in Z}{\operatorname{argmin}}\; s \cdot \|H_Z\, rw\|_{\mathbb{R}^n,1} + \; s \cdot \|H_Z\, iw\|_{\mathbb{R}^n,1} + \|rz - rw\|_{Z,2}^2 + \; \|iz - iw\|_{Z,2}^2$$

$$= \underset{rw \in \mathbb{R}^n}{\operatorname{argmin}}\, s \cdot \|H_Z\, rw\|_{\mathbb{R}^n,1} + \|rz - rw\|_{Z,2}^2 + j\; \underset{iw \in \mathbb{R}^n}{\operatorname{argmin}}\, s \cdot \|H_Z\, iw\|_{\mathbb{R}^n,1} + \|iz - iw\|_{Z,2}^2$$

This optimization problem reduces for each component independently to the 1-dimensional problem

$$\underset{w \in \mathbb{R}}{\operatorname{argmin}}\; s \cdot |\Delta Z_k\, w| + \Delta Z_k (rz_k - w)^2 = \; \underset{w \in \mathbb{R}}{\operatorname{argmin}}\; s \cdot |w| + (rz_k - w)^2 = \; prox_{s \cdot \|\cdot\|_{\mathbb{R},1}}(rz_k)$$

for the real part and similarly

$$\underset{w \in \mathbb{R}}{\operatorname{argmin}}\; s \cdot |\Delta Z_k\, w| + \Delta Z_k (iz_k - w)^2 = \; \underset{w \in \mathbb{R}}{\operatorname{argmin}}\; s \cdot |w| + (iz_k - w)^2 = \; prox_{s \cdot \|\cdot\|_{\mathbb{R},1}}(iz_k)$$

for the imaginary part.

It follows

$$\underset{w \in Z}{\operatorname{argmin}}\, s \cdot \|w\|_{Z,1} + \frac{1}{2}\|z - w\|_{Z,2}^2 = prox_{s \cdot \|\cdot\|_{\mathbb{R}^n,1}}(rz) + \; j\; prox_{s \cdot \|\cdot\|_{\mathbb{R}^n,1}}(iz)$$

It follows that the proximal operator of $s \cdot \|\cdot\|_{Z,1}$ defined on space Z with 2-norm $\|\cdot\|_{Z,2}$ is given by

$$prox_{s \cdot \|\cdot\|_{Z,1}}(z) = \; prox_{s \cdot \|\cdot\|_{\mathbb{R}^n,1}}(rz) + \; j\; prox_{s \cdot \|\cdot\|_{\mathbb{R}^n,1}}(iz)$$

It is simply the complexification of the soft-thresholding applied on each component rz and iz independently. We note in particular that this expression is independent of the matrix H_Z. Please take note that we will assume the following in the reste of the text.

Every time we encounter a 1-norm $\|\cdot\|_{Z,1}$ on a $2n$-dimensional vector space $Z \simeq \mathbb{C}^n$, it will be assumed that this 1-one norm is of the form

$$\|z\|_{Z,1} = \|H_Z\, z\|_{\mathbb{C}^n,1}$$

with real positive-definite diagonal $n \times n$ matrix

$$H_Z = \begin{bmatrix} \Delta Z_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Delta Z_n \end{bmatrix}$$

which is simultaneously the hermitian matrix of the E-product $\langle \cdot \mid \cdot \rangle_Z$ on Z. In that case, the proximal operator associated to the 1-norm is given by

$$prox_{s\cdot\|\cdot\|_{Z,1}}(z) = prox_{s\cdot\|\cdot\|_{\mathbb{R}^n,1}}(rz) + j\, prox_{s\cdot\|\cdot\|_{\mathbb{R}^n,1}}(iz)$$

3.2 The constrained 1-norm problem and the generalized-LASSO problem

We define "quadratically-constrained 1-norm minimization problem" (Q-constrained 1-norm problem) as the optimization problem

$$x^{\#} \in S_1(\delta_Y) \coloneqq \underset{x \in X}{\operatorname{argmin}} \left\{ \|\phi\, x\|_{Z,1}\; s.t.\; \|M\, x - y\|_{Y,2}^2 \leq {\delta_Y}^2 \right\}$$

Q-constrained 1-norm problem

where the real value δ_Y is a maximal tolerance (or residual), the vector space X is endowed with the E-product $\langle \cdot \mid \cdot \rangle_X$ and $\|\cdot\|_{X,2}$ is the induced 2-norm, vector space Y is endowed with the E-product $\langle \cdot \mid \cdot \rangle_Y$ and $\|\cdot\|_{Y,2}$ its induced 2-norm, vector space Z is endowed with the E-product $\langle \cdot \mid \cdot \rangle_Z$ and induced 2-norm $\|\cdot\|_{Z,2}$ and also with the 1-norm $\|\cdot\|_{Z,1}$ by a choice of a basis on Z, and where ϕ is a linear map from X to Z and M is a linear map from X to Y. We restrict ourselves on the case where the matrix H_Z of the E-product on Z is also the matrix of the 1-norm on Z. Note that the solution set S_1 is parametrized by the positive constant δ_Y.

We decompose spaces X and Y according to the orthogonal decomposition associated to M:

$$X = \overline{X} \oplus \overline{X}^{\perp} \qquad \text{and} \qquad Y = \overline{Y} \oplus \overline{Y}^{\perp}$$

with

$$\overline{X} = \mathrm{Ker}(M)^{\perp} = \mathrm{Im}(M^{\dagger}) \qquad \text{and} \qquad \overline{Y} = \mathrm{Ker}(M^{\dagger})^{\perp} = \mathrm{Im}(M)$$

The bar symbol $\bar{\cdot}$ stands for the orthogonal projection on $\overline{X}$ or $\overline{Y}$ depending on the variable under the bar. Therefore is $\bar{y}$ the projection of y on $\overline{Y}$ i.e. the closest point of $\mathrm{Im}(M)$ to y. The orthogonal decomposition of y is

$$y = \bar{y} + y_{\perp}$$

where $\bar{y} \in \overline{Y}$ and $y_{\perp} \in \overline{Y}^{\perp} = \mathrm{Im}(M)^{\perp}$ are unique.

We define the closed ball of radius δ_Y centered in y as the closed set

$$\overline{B_{\delta_Y}(y)} \coloneqq \{y' \in Y \,|\, \|y' - y\|_{Y,2} \leq \delta_Y\} \subset Y$$

We define the constraint function $\gamma(\cdot)$ by

$$\gamma(x) \coloneqq \|M\,x - y\|_{Y,2}^2 - {\delta_Y}^2$$

The constraint of the Q-constraint 1-norm problem can thus be written as

$$\gamma(x) \leq 0$$

The set of vectors that satisfy this constraint will be called the "feasible set" and will be denoted by $\Gamma(\delta_Y)$. A vector $x \in X$ satisfies that constraint exactly if

$$Mx \in \overline{B_{\delta_Y}(y)}$$

It follows that the feasible-set is the pre-image by M of the set $\mathrm{Im}(M) \cap \overline{B_{\delta_Y}(y)}$:

$$\Gamma(\delta_Y) = \{x \in X \,|\, Mx \in \mathrm{Im}(M) \cap \overline{B_{\delta_Y}(y)}\} = \{x \in X \,|\, Mx \in \overline{B_{\delta_Y}(y)}\}$$

This set is non-empty exactly if

$${\delta_Y}^2 \geq \|y_{\perp}\|_{Y,2}^2 = \|\bar{y} - y\|_{Y,2}^2$$

is satisfied. This gives a lower bound for δ_Y that that ensure that the feasible set is non-empty.

Since both $\mathrm{Im}(M)$ and $\overline{B_{\delta_Y}(y)}$ are closed, $\mathrm{Im}(M) \cap \overline{B_{\delta_Y}(y)}$ is closed too. And since M is continuous, the feasible set is closed as well.

In order to guaranty that $\Gamma(\delta_Y)$ satisfies the constraint qualification of Slater, it must have non-empty relative interior ([1] page 226). A sufficient condition for that is

$$\delta_Y{}^2 > \|y_\perp\|_{Y,2}^2$$

and we will assume that it is the case. The situation is described in figure 4.

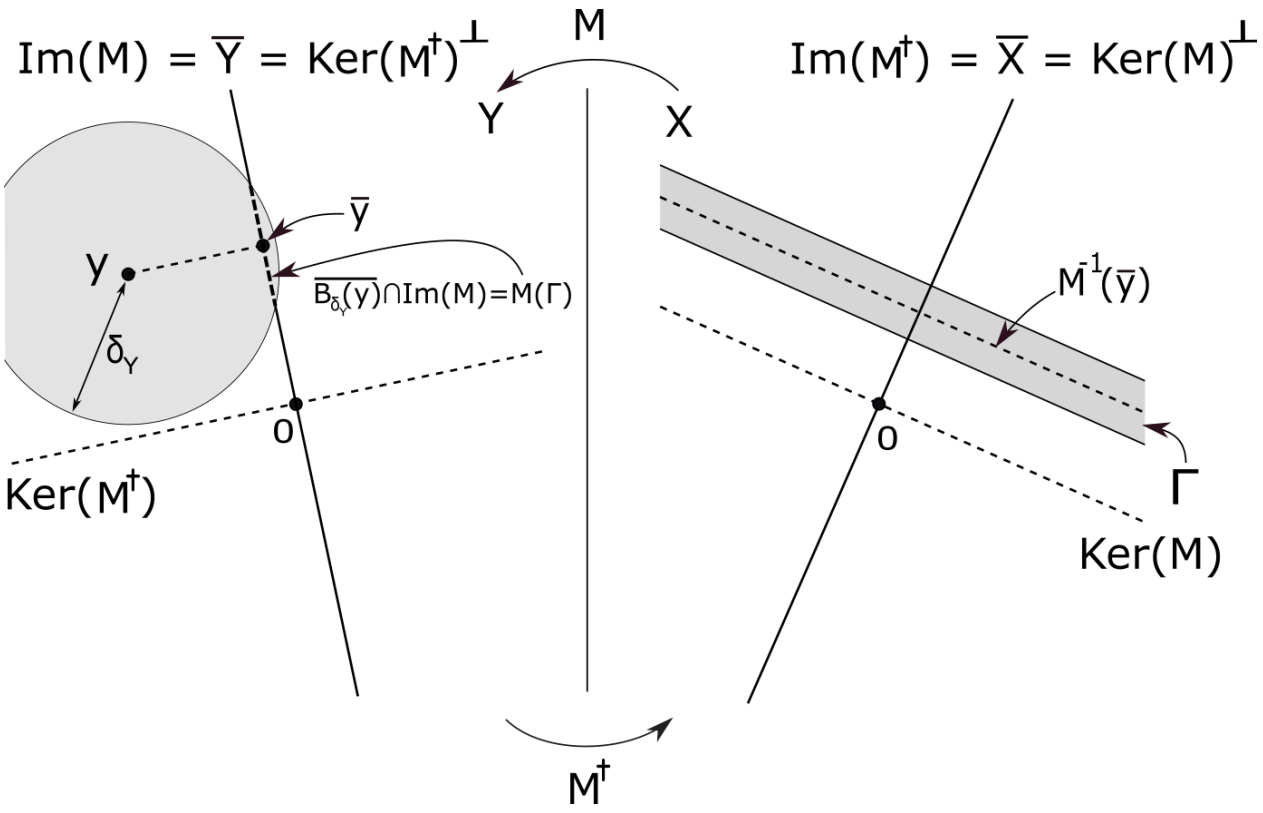

Figure 4: The feasible set is Γ (written $\Gamma(\delta_Y)$ in the text). The important fact about that figure is that if $\delta_Y < \|y_\perp\|_{Y,2}$, then is the feasible set of the quadratically-constrained 1-norm minimization empty. In that case exist no equivalent generalized-LASSO problem.

We can now reformulate the Q-constrained 1-norm problem as

$$x^{\#} \in S_1(\delta_Y) \coloneqq \operatorname{argmin}\{\|\phi\, x\|_{Z,1}\; s.t.\;\; x \in \Gamma(\delta_Y)\}$$

The feasible set $\Gamma(\delta_Y)$ is non-empty, closed and convex with non-empty relative interior. It verifies thus the constraint qualification of Slater. The objective function of the problem is continuous and convex. Our Q-constrained 1-norm problem verifies thus strong (Lagrange) duality.

To the Q-constrained 1-norm problem, we associate the generalized-LASSO problem

$$x^{\#} \in S_{LASSO}(\lambda) \coloneqq \underset{x\,\in\,X}{\operatorname{argmin}}\; \|\phi\, x\|_{X,1} + \frac{1}{\lambda}\, \|M\, x - y\|_{Y,2}^2$$

where λ is a positive constant. This can be rewritten in the standard form

$$x^{\#} \in S_{LASSO}(\lambda) = \underset{x \in X}{\operatorname{argmin}} \frac{1}{2}\|M\, x - y\|_{Y,2}^2 + \frac{\lambda}{2}\|\phi\, x\|_{Z,1}$$

Generalized-LASSO Problem

Note that the generalized-LASSO problem is unconstrained, in contrast to the Q-constrained 1-norm problem. This problem can be solved with the ADMM-algorithm (for generalized-LASSO problem).

It is usually assumed in the MRI literature that, as a result of the theory of Lagrange duality, there exist for any positive $\delta_Y > \|y_\perp\|_{Y,2}^2$ a positive value $\lambda(\delta_Y)$ so that any obtained solution for the generalized-LASSO problem is also a solution of the Q-constrained 1-norm problem, or in other words:

$$S_{LASSO}(\lambda(\delta_Y)) \subseteq S_1(\delta_Y)$$

We allow here a point of criticism concerning that assumption. The best result the author could find to justify such a clam is proposition 3.2 in [3]. But that proposition is only valid for the LASSO problem, which is equivalent to our generalized-LASSO only if ϕ is invertible. However is ϕ usually not invertible in our MRI applications. Moreover, this proposition 3.2 describes explicitly values λ and δ as dependent of the solution $x^{\#}$ of both problems, while the ideal situation we would like to be true for MRI reconstruction is that λ only depends on δ_Y. The help of mathematician would be welcome in order to clarify this situation.

In any case, even if the previously discussed assumption is true, choosing the appropriate value of λ as a function of δ_Y remains a non-trivial problem and there is no generic method for that as far as the author knows. To the time when the present text is written, the usual practice for performing so called "compressed-sensing" MRI reconstructions is to ignore the Q-constraint 1-norm problem and to solve the generalized-LASSO problem instead with an empirically chosen value for λ. We will follow that established procedure in the present version of the text. But it is a hope of the author that this uncomfortable situation will improve in the future.

As we will see in the section about the ADDMM-algorithm, each generalized-LASSO problem is associated to a 2-terms lest-square sub-problem given by

$$x^{\#} \in S_{LSQ} \coloneqq \underset{x \in X}{\operatorname{argmin}} \frac{1}{2}\|M\,x - y\|_{Y,2}^2 + \frac{\rho}{2}\|\phi x - z\|_{Z,2}^2$$

2-terms least-square sub-problem associated to the generalized-LASSO problem

where ρ is a positive constant. This problem can be solved with the CGD-algorithm (for 2-terms least-square problems).

The solution set of S_{LASSO} (i.e. the set of minimizer) for the generalized-LASSO problem is never empty (see [4] for example). But it contains several different solutions in general. For example, if $L \coloneqq Ker(M) \cap Ker(\phi)$ is not$\{0\}$, then for $0 \neq n \in L$ and $x^{\#}$ being a solution, is $x^{\#} + n$ is a different solution since

$$\frac{1}{2}\left\|M\,(x^{\#} + n) - y\right\|_{Y,2}^2 + \frac{\lambda}{2}\left\|\phi\,(x^{\#} + n)\right\|_{Z,1} = \frac{1}{2}\left\|M\,x^{\#} - y\right\|_{Y,2}^2 + \frac{\lambda}{2}\left\|\phi\,x^{\#}\right\|_{Z,1}$$
$$= \min_{x \in X} \frac{1}{2}\|M\,x - y\|_{Y,2}^2 + \frac{\lambda}{2}\|\phi\,x\|_{Z,1}$$

We give in appendix A another example of non-uniqueness even in the case where $L \coloneqq Ker(M) \cap Ker(\phi)$ is$\{0\}$.

> ***Informal note****: As observed by the author, solving the generalized-LASSO problem with the ADMM-algorithm (presented later) converges (seemingly) to different solutions when giving different initial images as input to the algorithm. This is an experimental indication that the optimization problem we solve for such reconstructions do not have a unique solution.*

3.3 The generalized-LASSO problem with multiple 1-norm-terms

Let be the following generalized-LASSO problem with multiple 1-norm-terms:

$$x^{\#} \in S_{LASSO} := \underset{x \in X}{\operatorname{argmin}} \frac{1}{2}\|M\,x - y\|_{Y,2}^2 + \frac{\lambda_1}{2}\|\phi_1\,x\|_{Z_1,1} + \cdots + \frac{\lambda_R}{2}\|\phi_R\,x\|_{Z_R,1}$$

Generalized-LASSO problem with multiple 1-norm terms

We are going to rewrite it as a generalized-LASSO problem with single 1-norm-term.

First of all, we define

$$\lambda := \frac{1}{R}\sum_{i=1}^{R}\lambda_i \qquad \text{and} \qquad l_i := \frac{\lambda_i}{\lambda} \qquad \text{and} \qquad l_i\phi_i := l_i \cdot \phi_i$$

and we rewrite the problem as

$$x^{\#} \in S_{LASSO} := \underset{x \in X}{\operatorname{argmin}} \frac{1}{2}\|M\,x - y\|_{Y,2}^2 + \frac{\lambda}{2}\|l_1\phi_1\,x\|_{Z_1,1} + \cdots + \frac{\lambda}{2}\|l_R\phi_R\,x\|_{Z_R,1}$$

We define furthermore the space Z by

$$Z := Z_1 \times \ldots \times Z_R$$

We assume here that $Z_i \simeq \mathbb{C}^{n_i}$ and we define $n := \sum_{i=1}^{R} n_i$. For $z_i \in Z_i \simeq \mathbb{C}^{n_i}, i \in \{1, \ldots, R\}$ we define

$$z := \begin{bmatrix} z_1 \\ \vdots \\ z_R \end{bmatrix} \in Z$$

We define the matrix H_Z by

$$H_z := \begin{bmatrix} H_{Z_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & H_{Z_R} \end{bmatrix}$$

We define the 1-norm on Z as

$$\|z\|_{Z,1} := \sum_{i=1}^{R}\|z_i\|_{Z_i,1} = \sum_{i=1}^{R}\left\|H_{Z_i}\,z_i\right\|_{\mathbb{C}^{n_i},1} = \|H_Z\,z\|_{\mathbb{C}^n,1}$$

and the E-product for any $a, b \in Z$ as

$$\langle a|b\rangle \coloneqq real\left\{\begin{bmatrix} a_1 \\ \vdots \\ a_R \end{bmatrix}^* \begin{bmatrix} H_{Z_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & H_{Z_R} \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_R \end{bmatrix}\right\} = real\{a^* H_Z b\} \; = \; \sum_{i=1}^{R} \langle a_i|b_i\rangle_{Z_i}$$

so that the 2-norm on Z is given by

$$\|z\|_{Z,2}^2 = \sum_{i=1}^{R} \|z_i\|_{Z_i,2}^2$$

We define the map ϕ as

$$x \longmapsto \phi x \coloneqq \begin{bmatrix} l_1\phi_1 \\ \vdots \\ l_R\phi_R \end{bmatrix} x = \begin{bmatrix} l_1\phi_1 \, x \\ \vdots \\ l_R\phi_R x \end{bmatrix}$$

Our problem with multiple 1-norm-terms can thus be rewritten as the generalized-LASSO problem

$$x^{\#} \in S \coloneqq \underset{x\in X}{\operatorname{argmin}} \frac{1}{2}\|M\,x - y\|_{Y,2}^2 + \frac{\lambda}{2}\|\phi\, x\|_{Z,1}$$

and can be solved with the ADMM-algorithm presented later.

The associated least-square sub-problem is

$$x^{\#} \in S \coloneqq \underset{x\in X}{\operatorname{argmin}} \frac{1}{2}\|M\,x - y\|_{Y,2}^2 + \frac{\rho}{2}\|\phi x - z\|_{Z,2}^2$$

By expending the 2-norm we obtain the following least-square problem with multiple terms:

$$x^{\#} \in S_{LSQ} \coloneqq \underset{x\in X}{\operatorname{argmin}} \frac{1}{2}\|M\,x - y\|_{Y,2}^2 + \frac{\rho}{2}\|l_1\phi_1\,x - z_1\|_{Z_1,2}^2 + \cdots + \frac{\rho}{2}\|l_R\phi_R\,x - z_R\|_{Z_R,2}^2$$

Multiple-terms least-square sub-problem associated to the generalized-LASSO problem with multiple 1-norm terms

which can be solved with the CGD-algorithm (for least-square problem with multiple terms).

Part II: Algorithms

4. The conjugate-gradient-algorithm

4.1 The CGD-algorithm for least-square problems

We recall the CGD-algorithm for least-square problems already presented in subsection 2.3.

CGD-algorithm (for least-square problems):

INITIALIZE

(a) Choose an $x_0 \in X$

(b) Evaluate the residual $r_0 = v - A\,x_0 \in V$

(c) Evaluate the search direction $p_0 = A^\dagger\, r_0 \in X$

DO

(d) Evaluate the line-search parameter : $a_i = \frac{\|A^\dagger r_i\|_{X,2}^2}{\|A\,p_i\|_{V,2}^2} \in \mathbb{R}$

(e) Update the approximated solution : $x_{i+1} = x_i + a_i\,p_i \in X$

(f) Update the residual: $r_{i+1} = r_i - a_i\,Ap_i \in V$

(g) Update the b-parameter: $b_i = \frac{\|A^\dagger r_{i+1}\|_{X,2}^2}{\|A^\dagger r_i\|_{X,2}^2}$

(h) Update the search direction: $p_{i+1} = A^\dagger r_{i+1} + b_i\,p_i \in X$

UNTIL $\|A^\dagger r_{i+1}\|_{X,2}^2$ ***is 0***

The CGD-algorithm (for least-square problems) as given above can hardly be translated directly into a programing language code. We define here a way to name the variables with alpha-numeric symbols and rewrite the algorithm in pseudo-code using these new variable names. In the following algorithm, all variable names are non-dissociable symbols, meaning that Ap, for example, do not mean "A time p". Instead, Ap designate a single variable. All products will be written with a dote "$\cdot$". The product "A times p", for example, will be written $A \cdot p$. We will typically write expression such that $Ap := A \cdot p$. The index i will be replaced by the postfix "**_curr**" (for "current") and the index $i+1$ will be replaced by the postfix "**_next**". The squared-norm of a variable var will be written "**sqn_**var". The adjoint of an homomorphism A will be written "**dag**A" (for "A dagger" or "dagger A"). The residuals, written with r in the algorithms above, will be written "**res**". The algorithm can now be written with alpha-numeric variable names on the left-hand-side and mathematical operation on the right-hand-side. Note the

explicit intervention the hermitian matrices H_X and H_V. The matrix H_X will be written HX and the matrix H_V will be written HV.

Pseudo-code for CGD-algorithm (for least-square problems)

INITIALIZE

Choose an initial guess x_0

Choose a machine-epsilon eps

Choose a maximal number of iteration $nIter\_max$

$$\text{nIter} = 0$$

$$\text{res\_next} = v - A \cdot x_0$$

$$\text{dagA\_res\_next} = A^{\dagger} \cdot \text{res\_next}$$

$$\text{p\_next} = \text{dagA\_res\_next}$$

$$\text{sqn\_dagA\_res\_next} = real\{\text{dagA\_res\_next}^{*} \cdot HX \cdot \text{dagA\_res\_next}\}$$

DO

$$\text{nIter} = \text{nIter} + 1$$

$$\text{res\_curr} = \text{res\_next}$$

$$\text{sqn\_dagA\_res\_curr} = \text{sqn\_dagA\_res\_next}$$

$$\text{p\_curr} = \text{p\_next}$$

IF $sqn\_dagA\_res\_curr < eps$ ***BREAK***

$$\text{Ap\_curr} = A \cdot \text{p\_curr}$$

$$\text{sqn\_Ap\_curr} = real\{\text{Ap\_curr}^{*} \cdot HV \cdot \text{Ap\_curr}\}$$

$$a = \frac{\text{sqn\_dagA\_res\_curr}}{\text{sqn\_Ap\_curr}}$$

$$\text{x\_next} = \text{x\_curr} + a \cdot \text{p\_curr}$$

IF $nIter \geq nIter\_max$ ***BREAK***

$$\text{res\_next} = \text{res\_curr} - a \cdot \text{Ap\_curr}$$

$$\text{dagA\_res\_next} = A^{\dagger} \cdot \text{res\_next}$$

$$\text{sqn\_dagA\_res\_next} = real\{\text{dagA\_res\_next}^{*} \cdot HX \cdot \text{dagA\_res\_next}\}$$

$$b = \frac{\text{sqn\_dagA\_res\_next}}{\text{sqn\_dagA\_res\_curr}}$$

$$\text{p\_next} = \text{dagA\_res\_next} + b \cdot \text{p\_curr}$$

END DO

4.2 Preconditioning

Let be the E-product $\langle \cdot \mid \cdot \rangle_X$ with associated H-product $(\cdot \mid \cdot)_X$ on a vector space $X \simeq \mathbb{C}^n$. Let be $S_X = \mathcal{R}H_X$ the matrix of that E-product in the real representation and H_X the matrix of the associated H-product. It holds then

$$\langle x_1 | x_2 \rangle_X = \mathcal{R}x_1{}^T S_X \, \mathcal{R}x_2 = \text{real}(x_1^* \, H_X \, x_2) \quad \forall x_1, x_2 \in X$$

Let be the E-product $\langle \cdot \mid \cdot \rangle_V$ with associated H-product $(\cdot \mid \cdot)_V$ on a vector space $V \simeq \mathbb{C}^m$. Let be $S_V = \mathcal{R}H_V$ the matrix of that E-product in the real representation and H_V the matrix of the associated H-product. It holds then

$$\langle v_1 | v_2 \rangle_V = \mathcal{R}v_1{}^T S_V \, \mathcal{R}v_2 = \text{real}(v_1^* \, H_V \, v_2) \quad \forall v_1, v_2 \in V$$

Since H_X resp. H_V are hermitian and positive-definite, there exist unique hermitian positive-definite square-roots (or principal square root) matrices W_X resp. W_V such that

$$H_X = W_X^* \, W_X \qquad \text{resp.} \qquad H_V = W_V^* \, W_V$$

or in the real representation

$$S_X = \mathcal{R}W_X^T \, \mathcal{R}W_X \qquad \text{resp.} \qquad S_V = \mathcal{R}W_V^T \, \mathcal{R}W_V$$

For $v_1, v_2 \in V$ it holds thus

$$\langle v_1 | v_2 \rangle_V = \text{real}(v_1^* H_V v_2) = \text{real}(v_1^* W_V^* \, W_V v_2) = \text{real}((W_V v_1)^* \, W_V v_2) = \langle W_V v_1 | W_V v_2 \rangle_{\mathbb{C}^m}$$

Similarily on X holds

$$\langle x_1 | x_2 \rangle_X = \langle W_X x_1 | W_X x_2 \rangle_{\mathbb{C}^n}$$

Let be A the matrix of a homomorphism from X to , $v \in V$ and $x \in X$. It holds then

$$\|A \, x - v\|_{V,2}^2 = \|W_V A \, x - W_V v\|_{\mathbb{C}^m,2}^2$$

We define the substitute variable $\hat{x}$ by

$$\hat{x} := W_X \, x \qquad \text{which means} \quad x = W_X^{-1} \hat{x}$$

It holds then

$$\langle x_1 | x_2 \rangle_X = \langle W_X x_1 | W_X x_2 \rangle_{\mathbb{C}^n} = \langle \hat{x}_1 | \hat{x}_2 \rangle_{\mathbb{C}^n} = \mathcal{R}\hat{x}_1{}^T \, \mathcal{R}\hat{x}_2$$

and

$$\|A\,x - v\|_{V,2}^2 = \|W_V A\,x - W_V v\|_{\mathbb{C}^m,2}^2 = \left\|W_V A\,W_X^{-1}\hat{x} - W_V v\right\|_{\mathbb{C}^m,2}^2$$

We do the substitution

$$\hat{A} \coloneqq W_V A\,W_X^{-1} \qquad \text{and} \qquad \hat{v} \coloneqq W_V v$$

It follows

$$S = \underset{x\in X}{\operatorname{argmin}}\|A\,x - v\|_{V,2}^2 = W_X^{-1}\underset{\hat{x}\in\mathbb{C}^n}{\operatorname{argmin}}\left\|\hat{A}\,\hat{x} - \hat{v}\right\|_{\mathbb{C}^m,2}^2$$

where the E-product for the variable $\hat{x} \in \mathbb{C}^n$ is the canonical E-product.

This is a reformulation of the lest-square problem on vector space $\mathbb{C}^n$ and $\mathbb{C}^m$ with canonical E-products and the set of substitutions

$$\hat{x} \coloneqq W_X\,x \qquad \hat{A} \coloneqq W_V A\,W_X^{-1} \qquad \hat{v} \coloneqq W_V v$$

is an example of preconditioning.

The solving of the least square problem can thus be achieved with the CGD-algorithm with canonical E-products by preconditioning (i.e. by performing the above substitutions). For solving least-square problems with the CGD-algorithm, preconditioning is an alternative and fully equivalent method to the use non-canonical E-products.

***Informal note**: We mention this method because it is popular in engineering and many MRI-reconstructions make use that. We will however stick to the use of non-canonical E-products in this book. But we want to stress the fact that it is only a matter of convention.*

4.3 The CGD-algorithm for least-square problems with 2 terms

The ADMM-algorithm presented later includes, as a sub-problem, the following two-terms least-square problem:

$$x^{\#} \in S_{LSQ} \coloneqq \underset{x}{\operatorname{argmin}}\frac{1}{2}\|M\,x - y\|_{Y,2}^2 + \frac{\rho}{2}\|\phi x - z\|_{Z,2}^2$$

We have

$$\|M\,x - y\|_{Y,2}^2 = real\{(M\,x - y)^*\,H_Y\,(M\,x - y)\}$$

and

$$\rho\|Dx - z\|_{Z,2}^2 = real\{(\phi x - z)^*\,\rho H_Z\,(\phi x - z)\}$$

In order to rewrite the objective function as a single square term, we define the homomorphism A by the concatenation

$$A \coloneqq \begin{bmatrix} M \\ \phi \end{bmatrix}$$

and the vector v by the concatenation

$$v \coloneqq \begin{bmatrix} y \\ z \end{bmatrix}$$

Since $Mx \in Y$ and $\phi x \in Z$ it follows that $Ax \in Y \times Z$. We define therefore $V \coloneqq Y \times Z$. It follows that any $a \in V$ can be written as the pair

$$a = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \qquad \text{with} \qquad a_1 \in Y \text{ and } \qquad a_2 \in Z.$$

In particular is

$$v = \begin{bmatrix} y \\ z \end{bmatrix} \in V$$

On V we define the E-product $\langle \cdot \mid \cdot \rangle_V$ as

$$\langle a|b\rangle_V = real\left\{\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}^* \cdot \begin{bmatrix} H_Y & 0 \\ 0 & \rho H_Z \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}\right\} = real\{a^* \cdot\, H_V \cdot b\} = \langle a_1|b_1\rangle_Y + \rho\langle a_2|b_2\rangle_Z$$

With

$$H_V = \begin{bmatrix} H_Y & 0 \\ 0 & \rho H_Z \end{bmatrix}$$

It follows immediately that the induced 2-norm on V is given by

$$\|a\|_{V,2}^2 = \left\|\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}\right\|_{V,2}^2 = \|a_1\|_{Y,2}^2 + \rho\|a_2\|_{Z,2}^2$$

We now observe that

$$\frac{1}{2}\|Ax - v\|_{V,2}^2 = \frac{1}{2}\left\|\begin{bmatrix} Mx - y \\ \phi\, x - z \end{bmatrix}\right\|_{V,2}^2 = \frac{1}{2}\|Mx - y\|_{Y,2}^2 + \frac{\rho}{2}\|\phi\, x - z\|_{Z,2}^2$$

Our 2-terms least-square optimization problem can thus be rewritten with a single square objective as

$$x^{\#} \in S_{LSQ} \coloneqq \underset{x}{\operatorname{argmin}} \frac{1}{2}\|Ax - v\|_{V,2}^2$$

Given an initial value x_0, a solution $x^{\#}$ can be obtained by mean of the CGD-algorithm (for least-square problem) given in subsection 4.1.

For the evaluation of $A^{\dagger}$ we note that

$$A^{\dagger} = H_X^{-1} A^* H_V = H_X^{-1}[M^* \quad \phi^*] \begin{bmatrix} H_Y & 0 \\ 0 & \rho H_Z \end{bmatrix} = H_X^{-1}[M^* H_Y \quad \rho\phi^* H_Z] = [M^{\dagger} \quad \rho\phi^{\dagger}]$$

We observe the subtle fact that a factor ρ is present in front of $\phi^{\dagger}$.

For implementation purposes, we also propose the corresponding pseudo-code. The matrices H_X and H_Y will be written HX and HY respectively. The matrix $\rho \cdot H_Z$ will be written ρH_Z.

CGD-algorithm (for least-square problems with 2 terms):

INITIALIZE

(a) Choose an $x_0 \in X$

(b) Evaluate the residual

$$res\_y_0 = y - M\, x_0 \ \in Y$$

$$res\_z_0 = z - \phi\, x_0 \ \in Z$$

(c) Evaluate the search direction

$$p_0 = M^\dagger\, res\_y_0 + \ \rho\, \phi^\dagger\, res\_z_0 \in X$$

DO

(d) Evaluate the line-search parameter :

$$A^\dagger r_i = \ M^\dagger\, res\_y_i + \ \rho\, \phi^\dagger\, res\_z_i$$

$$A\, p_i = \begin{bmatrix} M\, p_i \\ \phi\, p_i \end{bmatrix} \in V = Y \times Z$$

$$a_i = \frac{\left\| A^\dagger r_i \right\|_{X,2}^2}{\| A\, p_i \|_{V,2}^2} \in \mathbb{R}$$

(e) Update the approximated solution :

$$x_{i+1} = x_i + a_i\, p_i \in X$$

(f) Update the residual:

$$res\_y_{i+1} = \ res\_y_i - a_i\ Mp_i \ \in Y$$

$$res\_z_{i+1} = \ res\_z_i - a_i\ \phi p_i \ \in Z$$

(g) Update the b-parameter:

$$A^\dagger r_{i+1} = \ M^\dagger\, res\_y_{i+1} + \ \rho\, \phi^\dagger\, res\_z_{i+1}$$

$$b_i = \frac{\left\| A^\dagger r_{i+1} \right\|_{X,2}^2}{\| A^\dagger r_i \|_{X,2}^2}$$

(h) Update the search direction:

$$p_{i+1} = A^\dagger r_{i+1} + \ b_i\, p_i \ \in X$$

UNTIL $\left\| A^\dagger r_{i+1} \right\|_{X,2}^2$ ***is 0***

Pseudo-code for CGD-algorithm (for least-square problems with 2 terms)

INITIALIZE

Choose an initial guess x_0

Choose a machine-epsilon eps

Choose a maximal number of iteration $nIter\_max$

$$\text{nIter} = 0$$

$$\text{res\_y\_next} = y - M \cdot x_0$$

$$\text{res\_z\_next} = z - \phi \cdot x_0$$

$$\text{dagM\_res\_y\_next} = M^{\dagger} \cdot \text{res\_y\_next}$$

$$\text{dagF\_res\_z\_next} = \rho \cdot \phi^{\dagger} \cdot \text{res\_z\_next}$$

$$\text{dagA\_res\_next} = \text{dagM\_res\_y\_next} + \text{dagF\_res\_z\_next}$$

$$\text{p\_next} = \text{dagA\_res\_next}$$

$$\text{sqn\_dagA\_res\_next} = real\{\text{dagA\_res\_next}^{*} \cdot HX \cdot \text{dagA\_res\_next}\}$$

DO

$$\text{nIter} = \text{nIter} + 1$$

$$\text{res\_y\_curr} = \text{res\_y\_next}$$

$$\text{res\_z\_curr} = \text{res\_z\_next}$$

$$\text{sqn\_dagA\_res\_curr} = \text{sqn\_dagA\_res\_next}$$

$$\text{p\_curr} = \text{p\_next}$$

IF $sqn\_dagA\_res\_curr < eps$ ***BREAK***

$$\text{Mp\_curr} = M \cdot \text{p\_curr}$$

$$\text{Fp\_curr} = \phi \cdot \text{p\_curr}$$

$$\text{sqn\_Mp\_curr} = real\{\text{Mp\_curr}^{*} \cdot HY \cdot \text{Mp\_curr}\}$$

$$\text{sqn\_Fp\_curr} = real\{\text{Fp\_curr}^{*} \cdot \rho HZ \cdot \text{Fp\_curr}\}$$

$$\text{sqn\_Ap\_curr} = \text{sqn\_Mp\_curr} + \text{sqn\_Fp\_curr}$$

$$a = \frac{\text{sqn\_dagA\_res\_curr}}{\text{sqn\_Ap\_curr}}$$

$$\text{x\_next} = \text{x\_curr} + a \cdot \text{p\_curr}$$

IF $nIter \geq nIter\_max$ ***BREAK***

$$\text{res\_y\_next} = \text{res\_y\_curr} - a \cdot \text{Mp\_curr}$$
$$\text{res\_z\_next} = \text{res\_z\_curr} - a \cdot \text{Fp\_curr}$$

$$\text{dagM\_res\_y\_next} = M^{\dagger} \cdot \text{res\_y\_next}$$
$$\text{dagF\_res\_z\_next} = \rho \cdot \phi^{\dagger} \cdot \text{res\_z\_next}$$

$$\text{dagA\_res\_next} = \text{dagM\_res\_y\_next} + \text{dagF\_res\_z\_next}$$
$$\text{sqn\_dagA\_res\_next} = real\{\text{dagA\_res\_next}^{*} \cdot HX \cdot \text{dagA\_res\_next}\}$$

$$b = \frac{\text{sqn\_dagA\_res\_next}}{\text{sqn\_dagA\_res\_curr}}$$

$$\text{p\_next} = \text{dagA\_res\_next} + b \cdot \text{p\_curr}$$

END DO

4.4 The CGD-algorithm for least-square problems with multiple terms

In order to be able to solve the generalized-LASSO problem with multiple 1-norm-terms, we need to solve the following least-square sub problem

$$x^{\#} \in S_{LSQ} \coloneqq \underset{x \in X}{\operatorname{argmin}} \frac{1}{2} \|M\, x - y\|_{Y,2}^2 + \frac{\rho}{2} \|\phi x - z\|_{Z,2}^2$$

where ϕ resp. z are of the form

$$\phi = \begin{bmatrix} l_1 \phi_1 \\ \vdots \\ l_R \phi_R \end{bmatrix} \qquad \text{resp.} \qquad z = \begin{bmatrix} z_1 \\ \vdots \\ z_R \end{bmatrix}$$

Here are

$$l_i \phi_i : X \longrightarrow Z_i$$

homomorphisms for $i \in \{1, \dots, R\}$ where each Z_i is a vector space with its own E-product $\langle \cdot \mid \cdot \rangle_{Z_i}$ and the induced 2-norm $\|\cdot\|_{Z_i,2}$. We note that the previous least-square problem is therefore equal to the following least-square problem with multiple terms:

$$x^{\#} \in S_{LSQ} \coloneqq \underset{x \in X}{\operatorname{argmin}} \frac{1}{2} \|M\, x - y\|_{Y,2}^2 + \frac{\rho}{2} \|l\phi_1 - z_1\|_{Z_1,2}^2 + \cdots + \frac{\rho}{2} \|l\phi_R - z_R\|_{Z_R,2}^2$$

but this has only an informative role. For our needs, we only have to consider the 2-term least-square problem above. The CGD-algorithm that solves this problem is the CGD-algorithm (for least-square problems with 2-terms) written above. For the sake of clarity and in order to help its implementation, we will write now that algorithm again with all components of ϕ and z explicitly, and we will call it the "CGD-algorithm (for least-square problems with multiple-terms)". For implementation purposes, we also propose the corresponding pseudo-code.

CGD-algorithm (for least-square problems with multiple-terms):

INITIALIZE

(a) Choose an $x_0 \in X$

(b) Evaluate the residuals

$$res\_y_0 = y - M\, x_0 \ \in Y$$
$$res\_z_{1,0} = z_1 - l_1\phi_1\, x_0 \ \in Z_1$$
$$\vdots$$
$$res\_z_{R,0} = z_R - l_R\phi_R\, x_0 \ \in Z_R$$

(c) Evaluate the search direction

$$p_0 = M^\dagger\, res\_y_0 + \ \rho\, l_1\phi_1^\dagger\, res\_z_{1,0} + \cdots + \ \rho\, l_R\phi_R^\dagger\, res\_z_{R,0} \in X$$

DO

(d) Evaluate the line-search parameter :

$$A^\dagger r_i = \ M^\dagger\, res\_y_i + \ \rho\, l_1\phi_1^\dagger\, res\_z_{1,i} + \cdots + \ \rho\, l_R\phi_R^\dagger\, res\_z_{R,i} \in X$$
$$A\, p_i = \begin{bmatrix} M\, p_i \\ l_1\phi_1\, p_i \\ \vdots \\ l_R\phi_R\, p_i \end{bmatrix} \in V = Y \times Z_1 \times \ldots \times Z_R$$
$$a_i = \frac{\left\|A^\dagger r_i\right\|_{X,2}^2}{\|A\, p_i\|_{V,2}^2} \in \mathbb{R}$$

(e) Update the approximated solution :

$$x_{i+1} = x_i + a_i\, p_i \in X$$

(f) Update the residual:

$$res\_y_{i+1} = \ res\_y_i - a_i\ Mp_i \ \in Y$$
$$res\_z_{1,i+1} = \ res\_z_{1,i} - a_i\ l_1\phi_1 p_i \ \in Z_1$$
$$\vdots$$
$$res\_z_{R,i+1} = \ res\_z_{R,i} - a_i\ l_R\phi_R p_i \ \in Z_R$$

(g) Update the b-parameter:

$$A^\dagger r_{i+1} = \ M^\dagger\, res\_y_i + \ \rho\, l_1\phi_1^\dagger\, res\_z_{1,i+1} + \cdots + \ \rho\, l_R\phi_R^\dagger\, res\_z_{R,i+1}$$
$$b_i = \frac{\left\|A^\dagger r_{i+1}\right\|_{X,2}^2}{\left\|A^\dagger r_i\right\|_{X,2}^2}$$

(h) Update the search direction:

$$p_{i+1} = A^\dagger r_{i+1} + \ b_i\, p_i \ \in X$$

UNTIL $\left\|A^\dagger \boldsymbol{r_{i+1}}\right\|_{\boldsymbol{X,2}}^{\boldsymbol{2}}$ ***is 0***

Pseudo-code for CGD-algorithm (for least-square problems with multiple terms)

INITIALIZE

Choose an initial guess x_0

Choose a machine-epsilon eps

Choose a maximal number of iteration $nIter\_max$

$$\text{nIter} = 0$$

$$\text{res\_y\_next} = y - M \cdot x_0$$

$$\text{res\_z1\_next} = z1 - l_1\phi_1 \cdot x_0$$

$$\text{res\_z2\_next} = z2 - l_2\phi_2 \cdot x_0$$

$$\text{res\_z3\_next} = z3 - l_3\phi_3 \cdot x_0$$

$$\vdots$$

$$\text{dagM\_res\_y\_next} = M^{\dagger} \cdot \text{res\_y\_next}$$

$$\text{dagF1\_res\_z1\_next} = \rho \cdot l_1\phi_1^{\dagger} \cdot \text{res\_z1\_next}$$

$$\text{dagF2\_res\_z2\_next} = \rho \cdot l_2\phi_2^{\dagger} \cdot \text{res\_z2\_next}$$

$$\text{dagF3\_res\_z3\_next} = \rho \cdot l_3\phi_3^{\dagger} \cdot \text{res\_z3\_next}$$

$$\vdots$$

$$\text{dagA\_res\_next} = \text{dagM\_res\_y\_next} + \text{dagF1\_res\_z1\_next} + \text{dagF2\_res\_z2\_next} \ldots$$

$$\text{p\_next} = \text{dagA\_res\_next}$$

$$\text{sqn\_dagA\_res\_next} = real\{\text{dagA\_res\_next}^{*} \cdot HX \cdot \text{dagA\_res\_next}\}$$

DO

$$\text{nIter} = \text{nIter} + 1$$

$$\text{res\_y\_curr} = \text{res\_y\_next}$$

$$\text{res\_z1\_curr} = \text{res\_z1\_next}$$

$$\text{res\_z2\_curr} = \text{res\_z2\_next}$$

$$\text{res\_z3\_curr} = \text{res\_z3\_next}$$

$$\vdots$$

$$\text{sqn\_dagA\_res\_curr} = \text{sqn\_dagA\_res\_next}$$

$$\text{p\_curr} = \text{p\_next}$$

IF $sqn\_dagA\_res\_curr < eps$ ***BREAK***

$$\text{Mp\_curr} = M \cdot \text{p\_curr}$$

$$\text{F1p\_curr} = l_1\phi_1 \cdot \text{p\_curr}$$

$$\text{F2p\_curr} = l_2\phi_2 \cdot \text{p\_curr}$$

$$\text{F3p\_curr} = l_3\phi_3 \cdot \text{p\_curr}$$

$$\vdots$$

$$\text{sqn\_Mp\_curr} = real\{\text{Mp\_curr}^* \cdot HY \cdot \text{Mp\_curr}\}$$
$$\text{sqn\_F1p\_curr} = real\{\text{F1p\_curr}^* \cdot \rho HZ1 \cdot \text{F1p\_curr}\}$$
$$\text{sqn\_F2p\_curr} = real\{\text{F2p\_curr}^* \cdot \rho HZ2 \cdot \text{F2p\_curr}\}$$
$$\text{sqn\_F3p\_curr} = real\{\text{F3p\_curr}^* \cdot \rho HZ3 \cdot \text{F3p\_curr}\}$$
$$\vdots$$
$$\text{sqn\_Ap\_curr} = \text{sqn\_Mp\_curr} + \text{sqn\_F1p\_curr} + \text{sqn\_F2p\_curr} + \text{sqn\_F3p\_curr} + \ldots$$
$$a = \frac{\text{sqn\_dagA\_res\_curr}}{\text{sqn\_Ap\_curr}}$$
$$\text{x\_next} = \text{x\_curr} + a \cdot \text{p\_curr}$$

IF $nIter \geq nIter\_max$ ***BREAK***

$$\text{res\_y\_next} = \text{res\_y\_curr} - a \cdot \text{Mp\_curr}$$
$$\text{res\_z1\_next} = \text{res\_z1\_curr} - a \cdot \text{F1p\_curr}$$
$$\text{res\_z2\_next} = \text{res\_z2\_curr} - a \cdot \text{F2p\_curr}$$
$$\text{res\_z3\_next} = \text{res\_z3\_curr} - a \cdot \text{F3p\_curr}$$
$$\vdots$$
$$\text{dagM\_res\_y\_next} = M^{\dagger} \cdot \text{res\_y\_next}$$
$$\text{dagF1\_res\_z1\_next} = \rho \cdot l_1 \phi_1^{\dagger} \cdot \text{res\_z1\_next}$$
$$\text{dagF2\_res\_z2\_next} = \rho \cdot l_2 \phi_2^{\dagger} \cdot \text{res\_z2\_next}$$
$$\text{dagF3\_res\_z3\_next} = \rho \cdot l_3 \phi_3^{\dagger} \cdot \text{res\_z3\_next}$$
$$\vdots$$
$$\text{dagA\_res\_next} = \text{dagM\_res\_y\_next} + \text{dagF1\_res\_z1\_next} + \text{dagF2\_res\_z2\_next} + \ldots$$
$$\text{sqn\_dagA\_res\_next} = real\{\text{dagA\_res\_next}^* \cdot HX \cdot \text{dagA\_res\_next}\}$$
$$b = \frac{\text{sqn\_dagA\_res\_next}}{\text{sqn\_dagA\_res\_curr}}$$
$$\text{p\_next} = \text{dagA\_res\_next} + b \cdot \text{p\_curr}$$

END DO

5 The ADMM-algorithm for the generalized-LASSO problem

An optimization problem that will frequently be encountered in MRI reconstructions is the so called generalized-LASSO problem

$$x^{\#} \in S_{LASSO} := \underset{x \in X}{\operatorname{argmin}} \frac{1}{2}\|M\,x - y\|_{Y,2}^2 + \frac{\lambda}{2}\|\phi\,x\|_{Z,1}$$

where S_{LASSO} is the set of minimizers of the objective, X, Y and Z are real Euclidean vector spaces of finite dimension, the 2-norm $\|\cdot\|_{X,2}$ is induced by the E-product $\langle\cdot \mid \cdot\rangle_X$ on X, the 2-norm $\|\cdot\|_{Y,2}$ is induced by the E-product $\langle\cdot \mid \cdot\rangle_Y$ on Y, the 1-norm $\|\cdot\|_{Z,1}$ is induced by a choice of basis on Z and the 2-norm $\|\cdot\|_{Z,2}$ is induced by the E-product $\langle\cdot \mid \cdot\rangle_Z$, $y \in Y$ is a parameter, $\lambda > 0$ is a positive number and M resp. ϕ are homomorphisms given by

$$M : X \longrightarrow Y \qquad \text{resp.} \qquad \phi : X \longrightarrow Z$$

The following ADMM-algorithm finds a solution of the generalized-LASSO problem and was adapted from [5] as follows:

INITIALIZE parameters and variables

a) Chose a real positive constant ρ.
b) Initialize the variables x_{curr} , z_{curr} and u_{curr}

DO

c) $x_{next} \in \underset{x \in X}{\operatorname{argmin}} \frac{1}{2}\|M\,x - y\|_{Y,2}^2 + \frac{\rho}{2}\|\phi x - (z_{curr} - u_{curr})\|_{Z,2}^2$
d) $z_{next} = prox_{\lambda/\varrho \cdot \|\cdot\|_{Z,1}}(\phi\, x_{next} + u_{curr})$
e) $u_{next} = \phi\, x_{next} + u_{curr} - z_{next}$
f) Update $(x_{curr}, z_{curr}, u_{curr}) \longleftarrow (x_{next}, z_{next}, u_{next})$

UNTIL some stopping-criterion is satisfied

We note the generalized-LASSO problem above is a general formulation that includes the generalized-LASSSO problem with multiple 1-norm terms. If the problem includes only one 1-norm term, the least-square sub-problem can be solved with the CGD-algorithm (for least-square problems with 2-terms). If the problem includes multiples 1-norm terms, the least-square sub-problem can be solved with the CGD-algorithm (for least-square problems with multiples-terms).

We also note the symbol "$\in$" in sub-problem (c). In the original algorithm presented in [5] stands an "$=$" symbol instead, suggesting that sub problem (c) should have a single solution. In practice however, the solution of sub-problem (c) is usually not unique. Instead, we take x_{curr} and project it orthogonally on the solution set of sub-problem (c) with the CGD-algorithm in order to obtain x_{next}. The author ignore if this process then still leads to an algorithm that converges to a solution of the generalized-LASSO problem. It is probably a question that needs to be answered.

Part III: Discretization

6. Sampling of functions and approximation of norms with their associated Euclidean-products

In this chapter are done the first approximations. We bring a particular care to the definitions of the norms on vector-spaces, in order to ensure that these norms are induced, or approximated, by a sampling-independent norm. Doing so, we ensure that a resampling do not alter the norm of vectors, at least approximately.

In this first version of the text, part III contains only one chapter. The discretization of space and time with linear maps from space-time to other domains will be exposed in the next chapters coming in future versions.

6.1 Sampling

Let be $f(\cdot)$ a complex valued functions given by:

$$f(\cdot) : \mathbb{R}^d \longrightarrow \mathbb{C}$$

$$\overrightarrow{q} \longmapsto f(\overrightarrow{q})$$

where d is typically equal to 1, 2, or 3. The variable $\vec{q}$ will be called the position. Let be n a positive integer and let be $\vec{q}_1, \dots, \vec{q}_n$ a list of positions. We define

$$f_k \coloneqq f(\vec{q}_k), \qquad k = 1, \dots, n$$

The vector

$$f \coloneqq \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}$$

will be called the sampling of function $f(\cdot)$ according to the list of positions (or on the positions) $\vec{q}_1, \dots, \vec{q}_n$.

6.2 The 2-norm of a vector related to the L2-norm of a function

As it often appears in practice, let be $y \in Y \simeq \mathbb{C}^m$ the sampling vector of a complex-valued squared-integrable function $y(\cdot)$ defined on $\mathbb{R}^d$ and sampled on the positions $\vec{k}_1, \dots, \vec{k}_m \in \mathbb{R}^d$:

$$y_p = y(\vec{k}_p) \qquad \text{for} \qquad p = 1, \dots, m$$

The L^2-norm of the function $y(\cdot)$ is then given by

$$\|y(\cdot)\|_{L^2}^2 = \int_{\mathbb{R}^d} dk^d \ \left|y(\vec{k})\right|^2$$

The finite-element approximation of the integral on the sample points $\overrightarrow{k_1}, \dots, \overrightarrow{k_m}$ is

$$\int_{\mathbb{R}^d} dk^d \ \left|y(\vec{k})\right|^2 \approx \sum_{p=1}^{m} \Delta K_p \left|y(\vec{k}_p)\right|^2 = \sum_{p=1}^{m} \Delta K_p \left|y_p\right|^2 = \sum_{p=1}^{m} \Delta K_p \ ry_p{}^2 + \Delta K_p \ iy_p{}^2$$

where the real numbers $\Delta K_1, \dots, \Delta K_m$ are the volume elements containing each a point of the list $\overrightarrow{k_1}, \dots, \overrightarrow{k_m}$. Typically, ΔK_p can be chosen to be the Voronoi-region of point $\overrightarrow{k_p}$. By choosing rH to be the diagonal-matrix with diagonal-elements $\Delta K_1, \dots, \Delta K_m$ and iH to be the zero-matrix i.e.

$$H_Y = \begin{bmatrix} \Delta K_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & \Delta K_n \end{bmatrix} = rH_Y$$

we obtain

$$\sum_{p=1}^{m} \Delta K_p \ ry_p{}^2 + \Delta K_p \ iy_p{}^2 = ry^T \cdot H \cdot ry \ + \ iy^T \cdot H \cdot iy \ = \ \langle y|y\rangle_Y = \mathrm{real}(y|y)_Y = \ \|y\|_{Y,2}^2 \approx \|y(\cdot)\|_{L^2}^2$$

In that case the matrix S is given by

$$S_Y = \begin{bmatrix} rH_Y & 0 \\ 0 & rH_Y \end{bmatrix}$$

And we have shown that

$$\|y\|_{Y,2} \approx \|y(\cdot)\|_{L^2}$$

We have defined a H-product on the space Y by defining the matrix H_Y. We have therefore also defined the associated E-product. These products have a natural signification because their associated 2-norm is an approximation of the L^2-norm of function defined on Y.

We will also make use of similar but different definitions of 2-norms as follows. Let be a vector $x \in X \simeq \mathbb{C}^n$ be the sampling of a complex-valued squared-integrable function $x(\cdot)$ defined on $\mathbb{R}^d$ and sampled on the positions $\vec{r}_1, \dots, \vec{r}_n \in \mathbb{R}^d$:

$$x_p = x(\vec{r}_p) \qquad \text{for} \qquad p = 1, \dots, n$$

We restrict to the case where the positions sample a uniform (Cartesian) grid.

> ***Informal note****: As you probably guess, the positions $\vec{r}_1, \dots, \vec{r}_n \in \mathbb{R}^3$ are typically the voxel-center positions of an MRI-image and x represent typically the transverse magnetization.*

Let be the L^2-norm of the function $x(\cdot)$ given by

$$\|x(\cdot)\|_{L^2}^2 \coloneqq \int_{\mathbb{R}^d} dr^d \ \mu(\vec{r}) \ |x(\vec{r})|^2$$

where the positive-valued function $\mu(\cdot)$ is sufficiently well-behaved so that it defines a measure on $\mathbb{R}^d$. The finite-element approximation of the previous integral on the sample points $\overrightarrow{r_1}, \dots, \overrightarrow{r_n}$ is

$$\int_{\mathbb{R}^d} dr^d \ \mu(\vec{r})|x(\vec{r})|^2 \approx \sum_{p=1}^{n} \Delta_{VOL}\, \mu(\vec{r}_p) \left|x(\vec{r}_p)\right|^2 = \sum_{p=1}^{n} \Delta R_p \left|x_p\right|^2 = \sum_{p=1}^{n} \Delta R_p \ rx_p{}^2 + \Delta R_p \ ix_p{}^2$$

where the real positive numbers $\Delta R_1, \dots, \Delta R_n$ are defined by

$$\Delta R_p \coloneqq \Delta_{VOL} \cdot \mu(\vec{r}_p)$$

and where Δ_{VOL} is the volume of a cell of the uniform (Cartesian) grid sampled by the positions $\vec{r}_1, \dots, \vec{r}_n$.

By choosing rH_X to be the diagonal-matrix with diagonal-elements $\Delta R_1, \dots, \Delta R_n$ and iH_X to be the zero-matrix i.e.

$$H_X = \begin{bmatrix} \Delta R_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & \Delta R_n \end{bmatrix} = rH_X$$

we obtain

$$\sum_{p=1}^{n} \Delta R_p \ rx_p{}^2 + \Delta R_p \ ix_p{}^2 = rx^T \cdot H_X \cdot rx \ + \ ix^T \cdot H_X \cdot ix \ = \ \langle x|x\rangle_X = real(x|x)_X$$

$$= \|x\|_{X,2}^2 \approx \|x(\cdot)\|_{L^2}^2$$

In that case the matrix S is given by

$$S_X = \begin{bmatrix} rH_X & 0 \\ 0 & rH_X \end{bmatrix}$$

and we have shown that

$$\|x\|_{X,2} \approx \|x(\cdot)\|_{L^2}$$

This defines a spatially weighted 2-norm. The weighting can be done uniform by choosing $\mu(\cdot) = 1$, which means

$$\Delta X_p := \Delta_{VOL}$$

for all p.

6.3 The 1-Norm of a vector related to the L1-norm of a function

Please refer to sub-section 3.1 for the definition of the 1-norm on a complex- and real- finite dimensional vector space.

Let be $z(\cdot)$ a complex valued function given as

$$z(\cdot): \ \mathbb{R}^d \longrightarrow \mathbb{C}$$

$$\overrightarrow{r} \longmapsto z(\overrightarrow{r})$$

and let be $z \in Z \simeq \mathbb{C}^n$ the sampling of $z(\cdot)$ on the position $\overrightarrow{r_1}, \dots, \overrightarrow{r_n} \in \mathbb{R}^d$:

$$z_p = z(\overrightarrow{r_p}) \qquad \text{for} \qquad p = 1, \dots, n$$

Let be the real and imaginary part of $z(\cdot)$ given by the functions

$$rz(\cdot): \ \mathbb{R}^d \longrightarrow \mathbb{R}$$

and

$$iz(\cdot): \ \mathbb{R}^d \longrightarrow \mathbb{R}$$

It holds thus

$$rz_p = rz(\overrightarrow{r_p}) \quad \text{and} \quad iz_p = iz(\overrightarrow{r_p}) \quad \text{for} \quad p = 1, \dots, n$$

We assume furthermore that $rz(\cdot)$ and $iz(\cdot)$ are absolutely-integrable meaning that the integrals

$$\int_{\mathbb{R}^d} dr^d\, \mu(\vec{r})|rz(\vec{r})| \qquad \text{and} \qquad \int_{\mathbb{R}^d} dr^d\, \mu(\vec{r})\, |iz(\vec{r})|$$

both have finite values and where $\mu(\vec{r})$ is a sufficiently well-behaved positive-valued function so that it defines a measure on $\mathbb{R}^d$. We assume that Δ_{VOL} is the volume of the Voronoi-regions of each of the points $\overrightarrow{r_1}, \dots, \overrightarrow{r_n} \in \mathbb{R}^d$. This is typically the case when $\overrightarrow{r_1}, \dots, \overrightarrow{r_n}$ are located on a uniform (Cartesian) grid (such as the pixel-centers of a picture), in which case Δ_{VOL} is the volume of each cell of the grid. The finite-element approximation of the previous integrals are then

$$\sum_{p=1}^{N} \Delta Z_p \cdot \left|rz_p\right| \qquad \text{and} \qquad \sum_{p=1}^{N} \Delta Z_p \cdot \left|iz_p\right|$$

where

$$\Delta Z_p = \mu\left(\vec{r}_p\right) \Delta_{VOL}$$

We define the sampling-independent L^1-norm $\|\cdot\|_{L^1}$ of the function

$$\vec{r} \mapsto \begin{bmatrix} rz(\vec{r}) \\ iz(\vec{r}) \end{bmatrix}$$

as

$$\left\| \begin{bmatrix} rz(\cdot) \\ iz(\cdot) \end{bmatrix} \right\|_{L^1} \coloneqq \int_{\mathbb{R}^d} dr^d\, \mu(\vec{r})\, (|rz(\vec{r})| + |iz(\vec{r})|\,) \;=\; \|rz(\cdot)\|_{L^1} + \|iz(\cdot)\|_{L^1}$$

[6.3.1]

This is the most natural $L1$-norm defined for vector-valued function having each component absolutely-integrable since any pair of norms $\|\cdot\|_U$ (defined on a space U) and $\|\cdot\|_W$ (defined on a space W) naturally induce the norm $\|\cdot\|_{U\times W}$ (defined on the space $U \times W$) defined by

$$\left\| \begin{bmatrix} u \\ w \end{bmatrix} \right\|_{U\times W} \coloneqq \|u\|_U + \|w\|_W \;\forall \begin{bmatrix} u \\ w \end{bmatrix} \in U \times W$$

Since the value of this norm depends only of the function, it is sampling-independent.

This definition of the $L1$-norm of vector valued function is particularly suited for our needs since the finite-element approximation of [6.3.1] is given by

$$\left\| \begin{bmatrix} rz(\cdot) \\ iz(\cdot) \end{bmatrix} \right\|_{L^1} = \int_{\mathbb{R}^d} dr^d \, \mu(\vec{r}) \, (|rz(\vec{r})| + |iz(\vec{r})| \,) \;\approx\; \sum_{p=1}^{n} \Delta Z_p \, |rz_p| + \Delta Z_p \, |iz_p|$$

$$= \; \|z\|_{Z,1} = \; \|H_Z \, z\|_{\mathbb{C}^n,1}$$

where H_Z is the diagonal matrix

$$H_Z = \begin{bmatrix} \Delta Z_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Delta Z_n \end{bmatrix}$$

We have thus

$$\left\| \begin{bmatrix} rz(\cdot) \\ iz(\cdot) \end{bmatrix} \right\|_{L^1} \approx \; \|z\|_{Z,1} = \; \|H_Z \, z\|_{\mathbb{C}^n,1}$$

This shows that the 1-norm $\|\cdot\|_{Z,1}$ on the vector space Z is an approximation of a sampling-independent norm.

We never use the 1-norm $\|\cdot\|_{L^1}$ in practice. The reason why we define that norm is because it provides a number close to $\|z\|_{Z,1}$ independently of the sampling. If we assume another sampling

$$\overrightarrow{r_1'}, \ldots, \overrightarrow{r_{n'}'} \in \mathbb{R}^d$$

on another Cartesian grid with cell volume $\Delta Z'$, we obtain a different sampling $z' \in Z' \simeq \mathbb{C}^{n'}$ of the same function $z(\cdot)$. But for the associated 1-norm $\|\cdot\|_{Z',1}$ still holds

$$\left\| \begin{bmatrix} rz(\cdot) \\ iz(\cdot) \end{bmatrix} \right\|_{L^1} \approx \; \|z'\|_{Z',1}$$

and therefore

$$\|z\|_{Z,1} \approx \; \|z'\|_{Z',1}$$

The definition of the sampling independent $L1$-norm $\|\cdot\|_{L^1}$ allows thus to induce a weighted 1-norm on Z that is approximately independent of the chosen Cartesian sampling.

We note moreover that the diagonal matrix H_Z, which defines the 1-norm, naturally induce an E-product $\langle \cdot \mid \cdot \rangle_Z$ on Z. It is important that the matrices defining the E-product on Z and the one norm on Z co-inside i.e. are both equal to H_Z in order to guaranty that the proximal operator of the 1-norm is equal to

the soft-thresholding described in section 3.1 and remain independent of H_Z. If they are note, the soft-thresholding becomes component-dependent and we don't treat that case.

In the rest of the text, we will always assume the following.

Every time we encounter a 1-norm $\|\cdot\|_{Z,1}$ on a $2n$-dimensional vector space Z, it will be assumed that this 1-one norm is of the form

$$\|z\|_{Z,1} = \|H_Z\, z\|_{\mathbb{C}^n,1}$$

with real positive-definite diagonal $n \times n$ matrix

$$H_Z = \begin{bmatrix} \Delta Z_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Delta Z_n \end{bmatrix}$$

which is simultaneously the hermitian matrix of the E-product $\langle \cdot \mid \cdot \rangle_Z$ on Z. In that case, the proximal operator associated to the 1-norm is given by

$$prox_{s\cdot\|\cdot\|_{Z,1}}(z) = prox_{s\cdot\|\cdot\|_{\mathbb{R}^n,1}}(rz) + j\, prox_{s\cdot\|\cdot\|_{\mathbb{R}^n,1}}(iz)$$

Appendices

A. Counter-example to uniqueness of the solution of the generalized-LASSO problem

We consider $X = Y = Z = \mathbb{C}^2$ and we consider the homomorphism

$$M : X \longrightarrow Y = X$$

$$x \longmapsto Mx$$

given by the matrix

$$M = \sqrt{\frac{2}{5}} \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \qquad \text{or alternatively} \qquad \mathcal{R}M = \sqrt{\frac{2}{5}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 2 & 2 \end{bmatrix}$$

We chose $\phi = id_X$ so that $\mathrm{Ker}(\phi) = \{0\}$ and thus $L \coloneqq Ker(M) \cap Ker(\phi) = \{0\}$. We choose the canonical E-product on X. We chose $\lambda = 2$ and $y \in \mathbb{C}^2$ by

$$y = \sqrt{\frac{2}{5}} \begin{bmatrix} \zeta \\ 2\,\zeta \end{bmatrix}$$

where $\zeta \in \mathbb{R}$ so that iy is 0. Moreover we write x as

$$x = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbb{C}^2$$

with $\alpha, \beta \in \mathbb{C}$. We define then the objective function of our generalized-LASSO problem as

$$f(x) \coloneqq \frac{1}{2}\|M\,x - y\|_{X,2}^2 + \frac{\lambda}{2}\|\phi\,x\|_{Z,1} = \frac{1}{2}\|M\,x - y\|_{\mathbb{C}^2,2}^2 + \frac{2}{2}\|x\|_{\mathbb{C}^2,1}$$

$$= \frac{1}{2}\left\| \sqrt{\frac{2}{5}} \begin{bmatrix} \alpha + \beta - \zeta \\ 2\alpha + 2\beta - 2\zeta \end{bmatrix} \right\|_{\mathbb{C}^2,2}^2 + \left\| \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right\|_{\mathbb{C}^2,1}$$

$$= \frac{1}{5}\,|\alpha + \beta - \zeta|^2 + \frac{4}{5}\,|\alpha + \beta - \zeta|^2 + |\alpha| + |\beta|$$

$$= |\alpha + \beta - \zeta|^2 + |\alpha| + |\beta|$$

We note now that

$$f(x) = |\alpha + \beta - \zeta|^2 + |\alpha| + |\beta| = |r\alpha + r\beta - \zeta|^2 + |i\alpha + i\beta|^2 + |\alpha| + |\beta|$$

$$\geq |r\alpha + r\beta - \zeta|^2 + |r\alpha| + |r\beta| = f(rx)$$

The set of minimizer of $f(\cdot)$ is thus purely real. Hence we can restrict α and β to real values. We define the substitute variable

$$u(\alpha, \beta) \coloneqq \alpha + \beta \qquad \text{so that} \qquad \alpha = u - \beta$$

Then is

$$f(rx) = |u - \zeta|^2 + |u - \beta| + |\beta|$$

Since the objective function of the generalized-LASSO problem always reaches its infimum, there exists some minimizer

$$x^{\#} = rx^{\#} = \begin{bmatrix} \alpha^{\#} \\ \beta^{\#} \end{bmatrix} \in \mathbb{R}^2$$

So that

$$\inf_{x \in \mathbb{C}^2} f(x) = \left|u^{\#} - \zeta\right|^2 + \left|u^{\#} - \beta^{\#}\right| + \left|\beta^{\#}\right|$$

where $u^{\#} = \alpha^{\#} + \beta^{\#}$. We can assume without limit of generality that $u^{\#} > 0$ because we can always chose ζ large enough so that the value of $u^{\#}$ is forced to be larger than 0. It holds moreover

$$\beta^{\#} = \operatorname*{argmin}_{\beta \in \mathbb{R}} \left|u^{\#} - \beta\right| + |\beta|$$

We observe that

$$\left|u^{\#}\right| = \left|u^{\#} - \beta + \beta\right| \leq \left|u^{\#} - \beta\right| + |\beta|$$

By choosing $\beta \in [0, u^{\#}]$ it holds then

$$\left|u^{\#} - \beta\right| + |\beta| = u^{\#} - \beta + \beta = u^{\#} = \left|u^{\#}\right| = \min_{\beta' \in \mathbb{R}} \left|u^{\#} - \beta'\right| + |\beta'|$$

As a consequence, $\beta^{\#}$ can take any value in the interval $[0, u^{\#}]$ and the minimizer is not unique.

■

References

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. Cambridge University Press, 2004. doi: 10.1017/CBO9780511804441.

[2] M. R. Hestenes and E. Stiefel, 'Methods of conjugate gradients for solving linear systems', *J. RES. NATL. BUR. STAN.*, vol. 49, no. 6, p. 409, Dec. 1952, doi: 10.6028/jres.049.044.

[3] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*. in Applied and numerical harmonic analysis. New York: Birkhäuser, 2013.

[4] G. Giacchi, B. Milani, and B. Franchieschiello, 'On the determination of optimal tuning parameters for a space-variant LASSO problem using geometric and convex analysis techniques', 2023, doi: 10.48550/ARXIV.2301.09083.

[5] S. Boyd, 'Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers', *FNT in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010, doi: 10.1561/2200000016.