Explicit and proximal gradient descent for deformable mirrors actuation update

Théo Jolivet

January 17, 2020

1 Motivation

Problem formulation We want to solve

$$\min_{\mathbf{u}_k} F(\mathbf{u}_k) \quad \text{s.t.} \quad F(\mathbf{u}_k) := f(\mathbf{u}_k, \mathbf{E}_{k-1}) + \lambda g(\mathbf{u}_k) \tag{1}$$

with $\lambda > 0$ a regularization parameter, f the so-called fidelity term measuring the distance between the observed and the desired signals and g a regularization term where one can encode knowledge concerning some prior information on the desired solution if it is available. The minimisation problem (1) is in general non-smooth due to the non-smoothness of the regularising functional. A common choice for g and f is a sparsity-inducing regularization such as the ℓ_1-norm and a quadratic distance term respectively.

In the case of Stroke Minimization (SM) of Electric Field Conjugation (EFC), we have $f(\mathbf{u}_k, \mathbf{E}_{k-1}) = \frac{1}{2}(\mathbf{E}_{k-1}+\mathbf{G}_k\mathbf{u}_k)^\dagger(\mathbf{E}_{k-1}+\mathbf{G}_k\mathbf{u}_k)$ and $g(\mathbf{u}_k) = \frac{1}{2}||\mathbf{u}_k||_2^2$, with $\mathbf{u}_k \in \mathbb{R}^{N_{act}}$ the Deformable Mirrors (DMs) actuation vector, $\mathbf{E}_{k-1} \in \mathbb{R}^{N_{pix}}$ the electric field vector after correction step $k-1$ and $\mathbf{G}_k \in \mathbb{R}^{N_{pix}\times N_{act}}$ the Jacobian matrix. Without loss of generality, we consider the regularization parameter λ fixed.

Why The matrix inversion in SM and EFC is computationnaly expensive, and the method only allows for ℓ_2 regularization on the actuation. We make the case that gradient descent would be an efficient alternative to classical methods for wavefront control on space telescopes. Furthermore, the cost function is *convex*, which implies that gradient descent will converge to a *global minimizer* of F.

2 Explicit Gradient Descent

Problem We want to solve

$$\min_{\mathbf{u}_k} \frac{1}{2}(\mathbf{E}_{k-1} + \mathbf{G}_k\mathbf{u}_k)^\dagger(\mathbf{E}_{k-1} + \mathbf{G}_k\mathbf{u}_k) + \lambda\frac{1}{2}||\mathbf{u}_k||_2^2 \tag{2}$$

Principle We introduce here the idea behind the basic gradient descent which seeks to iteratively find the minimum of a function.

Gradient descent schemes are iterative algorithms used to solve:

$$\min_x F(x)$$

While gradient descent methods can be seen as outdated, many optimization problems can be solved using refined methods, that for instance use second-order information such as the Newton

method, or schemes that involve the previous and the current iterate in order to compute the gradient, such as the proximal algorithm. They belong in a group of optimization schemes called *first-order methods* because they are based on function values and first-order evaluations. They are commonly used for large-scale problems where the evaluation of second-order information would cost too much.

A first-order Taylor expansion around x easily shows that

$$F(y) = F(x) + \langle \nabla F(x), y - x \rangle + o(||y - x||_2)$$

Thus it is clear that the direction of greatest descent is $y - x \propto -\nabla F(x)$. Now let $y \equiv x^{k+1}$ and $x \equiv x^k$. Gradient descent gives us the iteration

$$x^{k+1} = x^k - \tau \nabla F(x^k) \tag{3}$$

Note that such a Taylor expansion only exists at the neighborhood of x, so we can already sense that for this method to converge, we will have to respect some constraints concerning τ.

Algorithm and convergence conditions In order for this algorithm to converge, we need to have ∇F L-Lipschitz with $L > 0$ and $0 < \tau < \dfrac{2}{L}$.

Here, F is an application which holds $\nabla F(\mathbf{u}_k) = \mathbf{G}_k^\dagger(\mathbf{G}_k\mathbf{u}_k + \mathbf{E}_{k-1}) + \lambda \mathbf{u}_k$, and is consequently L-Lipschitz continuous with $L = ||\mathbf{G}_k^\dagger \mathbf{G}_k + \lambda \mathbb{I}_{N_{act}}||_2$.

Proof. Let $F(x) = \dfrac{1}{2}(\mathbf{E}_{k-1} + \mathbf{G}_k\mathbf{u}_k)^\dagger(\mathbf{E}_{k-1} + \mathbf{G}_k\mathbf{u}_k) + \lambda\dfrac{1}{2}||\mathbf{u}_k||^2$

$$\begin{aligned}
& ||\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})||_2 \leq L||\mathbf{x} - \mathbf{y}||_2 \\
\Leftrightarrow \quad & ||\mathbf{G}_k^\dagger(\mathbf{G}_k\mathbf{x} + \mathbf{E}_{k-1}) + \lambda\mathbf{x} - \mathbf{G}_k^\dagger(\mathbf{G}_k\mathbf{y} + \mathbf{E}_{k-1}) - \lambda\mathbf{y}||_2 \leq L||\mathbf{x} - \mathbf{y}||_2 \\
\Leftrightarrow \quad & ||\mathbf{G}_k^\dagger\mathbf{G}_k(\mathbf{x} - \mathbf{y}) + \lambda\mathbb{I}_{N_{act}}(\mathbf{x} - \mathbf{y}) + \mathbf{G}_k^\dagger\mathbf{E}_{k-1} - \mathbf{G}_k^\dagger\mathbf{E}_{k-1}||_2 \leq L||\mathbf{x} - \mathbf{y}||_2 \\
\Leftrightarrow \quad & ||\mathbf{G}_k^\dagger\mathbf{G}_k + \lambda\mathbb{I}_{N_{act}}||_2||(\mathbf{x} - \mathbf{y})||_2 \leq L||\mathbf{x} - \mathbf{y}||_2
\end{aligned}$$

□

So the algorithm will converge as long as $0 < \tau < 2/||\mathbf{G}_k^\dagger\mathbf{G}_k + \lambda\mathbb{I}_{N_{act}}||_2$

We now write the pseudocode of the Gradient Descent algorithm, letting $\mathbf{u}_k \equiv \mathbf{x}$

Algorithm 1 Gradient descent with fixed step

1: Initialize: Choose $\mathbf{x}^0 \in \mathbb{R}^{N_{act}}$, $0 < \tau < \dfrac{2}{L}$, set $i = 0$
2: **while** $i < i_{max}$ **do**
3: $\quad \mathbf{x}^{i+1} = \mathbf{x}^i - \tau\nabla F(\mathbf{x}^i) = \mathbf{x}^i - \tau(\mathbf{G}_k^\dagger(\mathbf{G}_k\mathbf{x}^i + \mathbf{E}_{k-1}) + \lambda\mathbf{x}^i)$
4: $\quad i = i + 1$
5: **end**

3 Composite Gradient Descent

Problem We want to solve

$$\min_{\mathbf{u}_k} \frac{1}{2}(\mathbf{E}_{k-1} + \mathbf{G}_k\mathbf{u}_k)^\dagger(\mathbf{E}_{k-1} + \mathbf{G}_k\mathbf{u}_k) + \lambda||\mathbf{u}_k||_1 \tag{4}$$

Introduction to the proximal operator Let $g : K \to K$ be a closed, proper and lower semi-continuous convex function.

Definition 3.1 (Proximal operator)**.** *We define the proximal operator* $\boldsymbol{prox}_{\tau g}$*:* $K \to K$ *of* g *in correspondence to a parameter* $\tau > 0$ *as*

$$\boldsymbol{prox}_{\tau g}(\mathbf{v}) = \underset{\mathbf{x}}{\operatorname{argmin}}\ g(\mathbf{x}) + \frac{1}{2\tau}||\mathbf{x} - \mathbf{v}||_2^2 \tag{5}$$

An important remark is that on the right hand side we have a strongly convex function, there is therefore a unique minimizer of $g(\cdot) + \frac{1}{2\tau}||\cdot - \mathbf{v}||_2^2$.

For a given element $\mathbf{v}$, the proximal operator returns the (unique) element that compromizes between minimizing g and being close to $\mathbf{v}$. If τ is large, the point will be closer to the minimum of g, while for smaller values of τ, it will be closer to $\mathbf{v}$.

Fundamental property of the proximal operator: Let $\hat{\mathbf{y}} = \boldsymbol{prox}_{\tau g}(\mathbf{v})$. Then, basic differential calculus shows that

$$\partial g(\hat{\mathbf{y}}) + \frac{\hat{\mathbf{y}} - \mathbf{v}}{\tau} \ni 0 \Leftrightarrow \mathbf{v} \in \hat{\mathbf{y}} + \tau\partial g(\hat{\mathbf{y}}) \Leftrightarrow \hat{\mathbf{y}} = \underbrace{(\mathbb{I}_{N_{act}} + \tau\partial g)^{-1}}_{\boldsymbol{prox}_{\tau g}}(\mathbf{v}) \tag{6}$$

The last equivalence follows the fact that the proximal operator of $\mathbf{v}$ is unique.

Proximal gradient descent Several variational problems solving ill-posed inverse problems can be cast in a composite form. The problem here is to minimize $F(\mathbf{u}_k) = f(\mathbf{u}_k, \mathbf{E}_{k-1}) + \lambda g(\mathbf{u}_k)$ with f and g two proper convex functions and f differentiable with an L-Lipschitz gradient and g "simple".

Before introducing the suitable composite optimisation algorithm needed to solve the composite problem, let us remark the following property:

Proposition 3.1. *If* $\mathbf{x}^*$ *is a minimizer of* F*, we have*

$$\mathbf{x}^* = \boldsymbol{prox}_{\tau\lambda g}(\mathbf{x}^* - \tau\nabla f(\mathbf{x}^*))$$

Proof. Let $\mathbf{x}^*$ be a minimizer of F, then

$$\begin{aligned}
& 0 \in \lambda\partial g(\mathbf{x}^*) + \nabla f(\mathbf{x}^*) \\
\Leftrightarrow \quad & 0 \in \tau\lambda\partial g(\mathbf{x}^*) + \tau\nabla f(\mathbf{x}^*) \\
\Leftrightarrow \quad & \mathbf{x}^* - \tau\nabla f(\mathbf{x}^*) \in \mathbf{x}^* + \tau\lambda\partial g(\mathbf{x}^*) \\
\Leftrightarrow \quad & \mathbf{x}^* = (I_d + \tau\lambda\partial g)^{-1}(\mathbf{x}^* - \tau\nabla f(\mathbf{x}^*)) \\
\Leftrightarrow \quad & \mathbf{x}^* = \boldsymbol{prox}_{\tau\lambda g}(\mathbf{x}^* - \tau\nabla f(\mathbf{x}^*))
\end{aligned}$$

□

The passage from the third to the fourth line comes from the fact that the proximal operator is single-valued as seen in Eq. (6). This fixed-point equation suggests the possibility of iterating to converge to a minimizer of F thanks to Banach fixed point theorem.

The proximal operator for the ℓ_1-norm is the so-called "Soft Thresholding", which can be easily computed as $\boldsymbol{prox}_{\tau\lambda|\cdot|_1}(\mathbf{u}_k)_j = \max\,(0, 1 - \frac{\tau\lambda}{|u_{k,j}|})u_{k,j}$, with $u_{k,j}$ the j^{th} component of $\mathbf{u}_k$.

Proof. Let $\lambda g(\mathbf{x}) = \lambda||\mathbf{x}||_1$ and $\boldsymbol{prox}_{\tau\lambda g}(\mathbf{v}) = \underset{\mathbf{x}}{\operatorname{argmin}}\ \lambda||\mathbf{x}||_1 + \frac{1}{2\tau}||\mathbf{x}-\mathbf{v}||_2^2$. This function is separable so we minimize it with respect to every x_j .

If $x_j \geq 0$

$$\begin{aligned}
x_j^* &= \underset{x_j\geq 0}{\operatorname{argmin}}\ \lambda x_j + \frac{1}{2\tau}(x_j - v_j)^2 \\
&= \underset{x_j\geq 0}{\operatorname{argmin}}\ \frac{1}{2\tau}v_j^2 + \frac{1}{2\tau}u_j^2 - u_j(\frac{1}{\tau}v_j - \lambda) \\
&= \underset{x_j\geq 0}{\operatorname{argmin}}\ v_j^2 + u_j^2 - 2u_j(v_j - \tau\lambda) \\
&= \underset{x_j\geq 0}{\operatorname{argmin}}\ (u_j - (v_j - \tau\lambda))^2 \\
&= \max\,(0, v_j - \tau\lambda) \\
&= \max\,(0, 1 - \frac{\tau\lambda}{v_j})v_j
\end{aligned}$$

The same reasoning for $x_j < 0$ leads to $x_j^* = \max\,(0, 1 - \frac{\tau\lambda}{|v_j|})v_j$ □

The following pseudocode describes in a compact form the proximal algorithm, letting $\mathbf{u}_k \equiv \mathbf{x}$

Algorithm 2 Proximal gradient descent with fixed step

1: Initialize: Choose $\mathbf{x}^0 \in \mathbb{R}^{N_{act}},\ 0 < \tau < \frac{2}{L}$, set $i = 0$
2: **while** $i < i_{max}$ **do**
3 : $\quad \mathbf{x}^{i+1/2} = \mathbf{x}^i - \tau\nabla f(\mathbf{x}^i) = \mathbf{x}^i - \tau\mathbf{G}_k^\dagger(\mathbf{G}_k\mathbf{x}^i + \mathbf{E}_{k-1})$
4 : $\quad \mathbf{x}_j^{i+1} = \boldsymbol{prox}_{\tau\lambda g}(\mathbf{x})_j = \max\,(0, 1 - \frac{\tau\lambda}{|x_j^{i+1/2}|})x_j^{i+1/2},\ j = 1, \ldots, N_{act}$
5: $\quad i = i + 1$
6: **end**

Perspectives

- Implement the code for explicit gradient descent, compare convergence time with EFC.
- Implement the code for proximal gradient descent, compare efficiency with ℓ_2 regularization.
- Investigate Huber regularization, compare efficiency with ℓ_2 regularization.