

A GENERIC ONLINE ACCELERATION SCHEME FOR OPTIMIZATION ALGORITHMS VIA RELAXATION AND INERTIA

F. IUTZELER AND J. M. HENDRICKX *

Abstract. We propose generic acceleration schemes for a wide class of optimization and iterative schemes based on relaxation and inertia. In particular, we introduce methods that automatically tunes the acceleration coefficients online, and establish their convergence. This is made possible by considering the class of fixed-points iterations over averaged operators which encompass gradient methods, ADMM, primal dual algorithms, an so on.

Key words. Applied Optimization Methods, Relaxation, Inertia, Acceleration.

1. Introduction. A large class of optimization algorithms can be cast as fixed-point iterations in the sense that they consist in applying the same operation successively in order to converge to a fixed point of this operation. For the gradient algorithm on a differentiable function f , the operation consists in applying the identity minus the gradient of f , and the fixed point reached nulls the gradient of f . The convergence of such fixed-points iterations can be proven by finding a suitable contraction property, for which the *monotone operators* provide an attractive framework. They also provide an elegant framework to derive splitting algorithms such as the Alternating Direction Method of Multipliers (ADMM) [23], or, more recently, primal-dual algorithms [10], and randomized or distributed optimization algorithms [17, 38, 33, 18, 7].

In order to accelerate the convergence of fixed point algorithms and in particular optimization methods, there exists a variety of modifications based on the construction of the next iterate by combining the output of the operation with former outputs or iterates. We focus here on the two main modification schemes: *relaxation* and *inertia*.

■ *Relaxation* combines the output of the operation with the former iterate as

$$x_{k+1} = \eta T(x_k) + (1 - \eta)x_k$$

where η is some positive parameter. This modification notably appears in Richardson’s method for solving linear systems [31], and in Krasnoselskiĭ–Mann monotone operators convergence theorem. For the gradient algorithm, relaxation amounts to modifying the step-size. For ADMM, the benefits of relaxation are often reduced to the phrase “experiments [...] suggest that over-relaxation with $\eta \in [1.5, 1.8]$ can improve convergence.” (see [11] and [8, Chap. 3.4.3]) except in specific cases [13].

■ *Inertia* on the other side is performed by combining the output of the operation with the former output. An *inertial* iteration for operator T writes

$$\begin{cases} x_{k+1} = T(y_k) \\ y_{k+1} = x_k + \gamma(x_k - x_{k-1}) \end{cases} \Leftrightarrow x_{k+1} = T(x_k + \gamma(x_k - x_{k-1}))$$

where γ is some positive parameter. This modification was made immensely popular by Nesterov’s accelerated gradient algorithm [27]. More recently extensions of this method to proximal gradient (FISTA [6]) and ADMM (Fast ADMM [15]) were proposed and quite popular themselves.

*F.I. is with LJK, Université Grenoble Alpes, Grenoble, France. J.H. is with ICTEAM, Université Catholique de Louvain, Louvain-la-Neuve, Belgium. This project was conducted while F.I. was a post-doctoral researcher at UCL and is supported by the Belgian Network DYSCO, funded by the Belgian government and the Concerted Research Action (ARC) of the French Community of Belgium.

However, despite the popularity of these methods, proving the convergence of the iterates sequence (x_k) is still an issue in many situations (see *e.g.* [9, 3] for the case of FISTA) and additional restart mechanisms may have to be implemented to improve the convergence properties [29]. Finally, the key problem when using these methods is *tuning efficiently* their parameters. Indeed, “good”, if not optimal, parameters depend on a variety of elements including the algorithm itself (an optimal parameter for the gradient may make the ADMM divergent for instance) or function parameters in the case of optimization (often through the strong convexity constant which may be hard to estimate [22] or maladjusted to local analysis [35]).

Contributions. In this paper, our aim is to propose *online acceleration methods* for a general class of fixed point algorithms that encompasses the aforementioned optimization methods. The idea of generic acceleration using inertia was investigated in the sub-linear case in [21] by sequentially solving well-chosen strongly convex approximations of the original problem or in [22, 14] which are based on line-search.

Our approach is to be based on the monotone operators framework and more precisely on the *averaging contraction property*, verified by a large class of algorithms such as (proximal) gradient algorithms and, very interestingly, ADMM and recent primal-dual algorithms for which only seldom results exist concerning the choice of relaxation or inertial parameters. More precisely, we begin by considering the particular case of affine operators ($T(x) = Rx + d$ where R is a matrix and d a vector) and study how these modifications translate for the spectrum of the linear part and thus for the convergence rate. This spectral characterization makes possible the derivation of optimal parameters (in the linear case) and gives us useful guidelines for the general case. Our online acceleration methods are based on approximating the base algorithm by an affine operator at each iteration and choosing the acceleration parameter as the optimal one for the linear approximation. Finally, we illustrate the performance of our online acceleration methods for the proximal gradient algorithm, ADMM¹, and a primal-dual algorithm on popular lasso and logistic regression problems.

The paper is organized as follows. In Section 2, we introduce the averaged operators framework and related useful lemmas. In Section 3, we formulate Relaxation, Inertia, and Alternated Inertia as modifications on the fixed-point iterations on averaged operators; we provide a coherent set of results concerning convergence (in the general case) and linear rate in the case of affine operators. In Section 4, based on the previous analysis, we derive and prove the convergence of our online acceleration methods. These algorithms are based on the general operator framework and thus fit a large variety of optimization algorithms. Finally, Section 5 is devoted to numerical illustrations.

2. Fixed-point Algorithms.

2.1. Averaged Operators. Let T be a mapping² on \mathbb{R}^N . T is said *monotone* if $\forall x, y \in \mathbb{R}^N$, $\langle x - y; T(x) - T(y) \rangle \geq 0$. For $\alpha \in]0, 1[$, T is said α -*averaged* iff

$$\forall x, y \in \mathbb{R}^N, \quad \|T(x) - T(y)\|^2 + \frac{1 - \alpha}{\alpha} \|(I - T)(x) - (I - T)(y)\|^2 \leq \|x - y\|^2$$

and T is said to be *Firmly Non-Expansive (FNE)* if it is $1/2$ -averaged. The set of the fixed points of T will be denoted by $\text{fix}T = \{\bar{x} : \bar{x} = T\bar{x}\}$.

¹For ADMM, this led us to develop a new *Inertial ADMM*, different from Fast ADMM [15], build on the monotone operator formulation (see [11] and references therein).

²For the sake of clarity, we only discuss single-valued mappings in finite dimensional spaces; further results on monotone operators theory can be found in [5].

For instance, if f is a convex function, then its subgradient ∂f is monotonous. $J = (I + \partial f)^{-1}$ is FNE. Furthermore, if its gradient ∇f is L -Lipschitz continuous, $G = I - (1/L)\nabla f$ is also FNE. Both the fixed points of J and G coincide with the points where 0 belong to ∂f i.e. the minimizers of f . Similar derivations can be performed for a large class of algorithms such as the proximal gradient, the ADMM, etc.

LEMMA 1 (Krasnoselskiĭ–Mann algorithm). [5, Prop. 5.15] *Let $\alpha \in]0, 1[$. Let T be an α -averaged operator such that $\text{fix}T \neq \emptyset$. Then, the sequence $(x_k)_{k>0}$ generated by $x^0 \in \mathbb{R}^N$ and the iterations*

$$x_{k+1} = T(x_k)$$

converges to a point in $\text{fix}T$.

REMARK 1. *The iterations produced by averaged operators give Fejér monotonous iterates sequences $(x_k)_{k>0}$: for any fixed point \bar{x} and iteration k ,*

$$\|x_{k+1} - \bar{x}\| \leq \|x_k - \bar{x}\|;$$

we will investigate this attractive property for the modifications considered.

2.2. Linear Convergence of Affine Operators. We now give a precise characterization of the spectral signification of the averaging property for an affine operator. This will be useful to investigate the relaxation and inertia and will lead to our online algorithms for the general class of averaged operators.

Results of the literature include analyses of matrices with subdominant eigenvalues and applications to alternating projections and Douglas-Rachford splitting [4] or spectral analysis in the case of the FISTA algorithm [35]. The novelties in our characterization include i) a proof that the algebraic and geometric multiplicities of eigenvalue 1 coincide, which allows the definition of a proper projection onto the fixed points space (see Apx. A); ii) the characterization of the practical linear convergence rate based on the greatest eigenvalue in magnitude, 1 excluded (Theo. 2); and iii) the derivation of the position of the eigenvalues under the averaging property (Lemma 3). For the sake of clarity, all the proof details are reported in Apx. A.

T is an *affine operator* denoted by $T = R \cdot + d$ if it can be written

$$T(x) = Rx + d$$

where R is an $N \times N$ real matrix and d is a size- N real vector.

Let us define the eigenspace of R linked to eigenvalue 1: $\mathcal{N} \triangleq \{x \in \mathbb{R}^N : Rx = x\}$. Importantly, as shown in Apx. A, the averaging property implies that one can define a projection $\Pi_{\mathcal{N}}$ onto \mathcal{N} ; and thus the complementary projection $\overline{\Pi_{\mathcal{N}}}$.

THEOREM 2. *Let $\alpha \in]0, 1[$. Let $T = R \cdot + d$ be an α -averaged operator and suppose that $\text{fix}T \neq \emptyset$. Then, the sequence $(x_k)_{k>0}$ generated by $x^0 \in \mathbb{R}^N$ and*

$$x_{k+1} = T(x_k)$$

converges linearly to a point in $\text{fix}T$ at a rate

$$\nu \triangleq \max\{|\lambda_i| : \lambda_i \neq 1 \text{ is an eigenvalue of } R\} < 1$$

in the sense that $\exists \bar{x} \in \text{fix}T, \limsup_k \frac{\log \|\overline{\Pi_{\mathcal{N}}}(x_k - \bar{x})\|}{k} \leq \log \nu$.

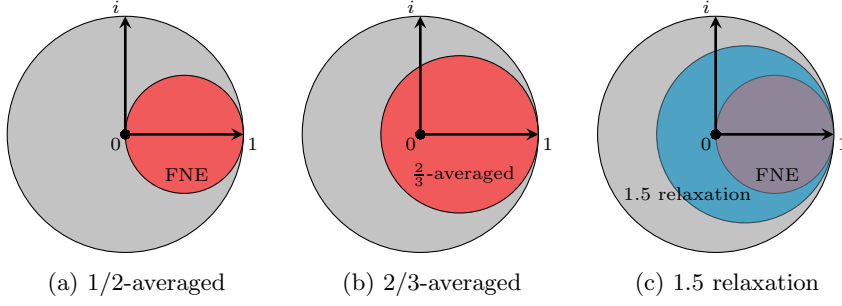


Fig. 1: Eigenvalues disks of some α -averaged linear operators

The proof is reported in Apx. A. In the sequel, we will call any eigenvalue λ such that $|\lambda| = \nu$ a *dominant* eigenvalue and we will approximate it online as $v_k = \|x_{k+1} - x_k\| / \|x_k - x_{k-1}\|$.

REMARK 2. *This definition of the convergence rate differs from [4] (notably Example 2.11) as taking the log enables to retrieve directly the principal eigenvalue and not some $\nu + \varepsilon$ or $k^n \nu^k$; this choice was made in order to match practical rates and justifies our next analysis.*

LEMMA 3. *Let $\alpha \in]0, 1[$ and $\mathsf{T} \triangleq R \cdot + d$ be an α -averaged affine operator. Then, every eigenvalue λ_i of R satisfies $|\lambda_i - (1 - \alpha)| \leq \alpha$. Furthermore, $|\lambda_i| \leq 1$ with equality iff $\lambda_i = 1$, so $\nu < 1$.*

This lemma shows, if T is α -averaged, the eigenvalues of R are contained in a disk of center $1 - \alpha$ and radius α as illustrated by Fig. 1-a,b.

EXAMPLE 1 (Gradient algorithm on a Quadratic Function). *For a differentiable convex function f with an L -Lipschitz gradient ∇f , the standard gradient algorithm writes*

$$(1) \quad x_{k+1} = x_k - \frac{1}{L} \nabla f(x_k)$$

and the related operator is $\mathsf{T} = \text{Id} - \frac{1}{L} \nabla f$.

For this illustration, we take quadratic $f(x) = \frac{1}{2} \|Ax - b\|^2$; thus, f is L -smooth with $L = \lambda_{\max}(A^T A)$ and μ -strongly convex with $\mu = \lambda_{\min}(A^T A)$. The iterations are affine and the spectrum of the linear part of T is comprised in the interval $[0, 1 - \mu/L]$ so we obtain the well-known rate $\nu = 1 - \mu/L$.

3. Relaxation and Inertia. In this section, we describe *Relaxation*, *Inertia*, and *Alternated Inertia* as modifications on the classical fixed-point iterations presented above. Notably, we give convergence results for the iterates, and exhibit the differences in monotonicity between inertia and relaxation. In addition, we derive optimal parameters and rates in the case of affine operators with real eigenvalues.

3.1. Relaxation.

3.1.1. Convergence. For a positive sequence (η_k) , the *relaxed* iterations follow

$$(2) \quad x_{k+1} = \eta_k \mathsf{T}(x_k) + (1 - \eta_k) x_k = \mathsf{T}(x_k) + (\eta_k - 1)(\mathsf{T}(x_k) - x_k).$$

As mentioned in the introduction, this modification is present since Richardson's iterations and Krasnosel'skiĭ–Mann algorithm, and over-relaxation ($\eta > 1$) is still investigated to improve convergence speed. The following result directly comes from Krasnoselskiĭ–Mann theorem.

LEMMA 4. *Let $\alpha \in]0, 1[$ and let the sequence (η_k) verify $0 < \eta \leq \eta_k \leq \bar{\eta} < 1/\alpha$ for all $k > 0$. Let T be an α -averaged operator such that $\text{fix } T \neq \emptyset$. Then, the sequence $(x_k)_{k>0}$ generated by $x^0 \in \mathbb{R}^N$ and the iterations*

$$x_{k+1} = \eta_k T(x_k) + (1 - \eta_k)x_k$$

converges to a point in $\text{fix } T$.

The proof is based on the fact that if $\alpha \in]0, 1[$ and $\eta \in]0, 1/\alpha[$, then T_η is $\eta\alpha$ -averaged [5, Prop. 4.28]. The convergence thus directly follows from Lemma 1 and the produced iterates are *monotonous* in the light of Remark 1.

3.1.2. Optimal parameters for real eigenvalues. Let $T = R \cdot + d$ be an α -averaged linear operator. Suppose that R has real eigenvalues $\lambda_i \in [1 - 2\alpha, \lambda] \cup \{1\}$. The eigenvalues of $R_\eta = \eta R + (1 - \eta)I$ have the form $\mu_i = \eta\lambda_i + (1 - \eta)$. The effect of over-relaxation (for $\eta > 1$) is thus the combination of an inflation and a translation as seen in Figure 1-c.

- i) When $\eta > 0$ is small enough, the dominant eigenvalue of R_η is $\eta\lambda + (1 - \eta) > 0$; so that the convergence rate ν will decrease when η increases.
- ii) When $\eta < 1/\alpha$ is big enough, the dominant eigenvalue of R_η will be $\eta(1 - 2\alpha) + (1 - \eta) = 1 - 2\alpha\eta < 0$; so that the convergence rate ν will increase when η increases. Finally, The optimal parameter η^* , which minimizes the rate, corresponds to the case where the dominant eigenvalues in the two cases are the opposite one of each other:

$$\eta^* = \frac{2}{2\alpha + 1 - \lambda} \text{ and optimal rate } \nu^* = \frac{2\alpha - 1 + \lambda}{2\alpha + 1 - \lambda}.$$

In the field of iterative methods for solving linear systems, relaxation has received a lot of attention and the optimal relaxation boils down to Richardson/Chebyshev iterations (see [32, Example 4.1] and Fig. 2a for an illustration).

APPLICATION IN THE SETUP OF EX. 1: *The relaxed iterations write*

$$x_{k+1} = x_k - \frac{\eta_{k+1}}{L} \nabla f(x_k)$$

and thus relaxation simply consists in adjusting the step size for the gradient algorithm and we have the following optimal relaxation parameter

$$\eta^* = \frac{2}{1 + \mu/L} \text{ and rate } \nu^* = \frac{1 - \mu/L}{1 + \mu/L}$$

leading to an optimal stepsize of $2/(\mu + L)$ which matches the asymptotic optimal stepsize (see e.g. [36, Sec. 4.1.2]).

3.2. Inertia.

3.2.1. Convergence. Stemming from popular inertial methods [30, 27, 28], acceleration techniques based on the use of the memory of the previous outputs are very popular both from a theoretical and a practical point of view (see [37] and references

therein for an overview of these methods). Formally, with T an operator, the core of these methods consist in performing the following iterations.

$$(3) \quad \begin{cases} x_{k+1} = \mathsf{T}(y_k) \\ y_{k+1} = x_{k+1} + \gamma_k(x_{k+1} - x_k) \end{cases} \Leftrightarrow x_{k+1} = \mathsf{T}(x_k + \gamma_k(x_k - x_{k-1}))$$

A careful choice of the sequence $(\gamma_k)_{k>0}$ is known to accelerate the theoretical functional convergence rate from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/k^2)$ for a large class of algorithms (see [37, 9, 20] for details) and is very popular in practice.

However, contrary to the relaxation, this modification of the algorithm deeply changes the algorithm behavior as the error between the iterates and some fixed point is *not monotonously decreasing* anymore which can cause stability or domain problems for the iterates. The next lemma provides a general set of conditions for iterates convergence encompassing several results of the literature [2, 1, 25, 24, 21].

LEMMA 5. *Let $\alpha \in]0, 1[$. Let T be an α -averaged operator such that $\text{fix}\mathsf{T} \neq \emptyset$. Assume one the following:*

- i) $\exists \gamma, 0 \leq \gamma_k \leq \bar{\gamma} < 1$ and $\sum_{k=1}^{\infty} \gamma_k \|x_k - x_{k-1}\|^2 < \infty$.
- ii) $\exists \bar{\gamma} < 1$, $(\gamma_k)_{k>0}$ is non-decreasing sequence in $[0, \bar{\gamma})$ such that $\forall k > 1$

$$1 - \gamma_{k-1} - (1 - \gamma_k)\gamma_k - \frac{\alpha}{1 - \alpha}\gamma_k(1 + \gamma_k) \geq \underline{m} > 0.$$

iii) as a particular case of ii), when $\gamma_k = \gamma$ for all $\forall k > 1$, $(1 - \gamma)^2 > \frac{\alpha}{1 - \alpha}\gamma(1 + \gamma)$. Then, the sequence $(x_k)_{k>0}$ generated by $x_{-1} = x_0 \in \mathbb{R}^N$ and the iterations

$$x_{k+1} = \mathsf{T}(x_k + \gamma_k(x_k - x_{k-1}))$$

converges to a point in $\text{fix}\mathsf{T}$.

3.2.2. Optimal parameters for real eigenvalues. Let us define T^γ , the operator generating (x_{k+1}, x_k) from (x_k, x_{k-1}) where $x_{k+1} = \mathsf{T}(x_k + \gamma(x_k - x_{k-1}))$. When $\mathsf{T} = R \cdot + d$, we have

$$\mathsf{T}^\gamma \left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \right) = \begin{bmatrix} (1 + \gamma)Rz_1 - \gamma Rz_2 + d \\ z_1 \end{bmatrix} = \underbrace{\begin{bmatrix} (1 + \gamma)R & -\gamma R \\ I & 0 \end{bmatrix}}_{R^\gamma} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \underbrace{\begin{bmatrix} d \\ 0 \end{bmatrix}}_{\tilde{d}}.$$

As for relaxation, the eigenvalues of R^γ can be derived from those of from R . However, one eigenvalue λ_i of R leads to two eigenvalues for R^γ ; they are the roots of

$$p_i(\mu) = \mu^2 - (1 + \gamma)\lambda_i\mu + \gamma\lambda_i.$$

The main results are:

- i) for *negatives eigenvalues* $\lambda_i < 0$, the magnitude of μ_i is

$$\frac{(1 + \gamma)|\lambda_i| + \sqrt{(1 + \gamma)^2\lambda_i^2 + 4\gamma|\lambda_i|}}{2} \geq (1 + \gamma)|\lambda_i|$$

thus inertia has a *negative effect* on the negative side of the spectrum. For the sake of clarity, we will focus on the non-negative eigenvalue case in the following, corresponding to $\alpha \in (0, 1/2]$ for the averaging property.

- ii) for *non-negative eigenvalues* $\lambda_i \in [0, \lambda] \cup \{1\}$, optimal parameter and rate are

$$\gamma^* = \frac{(1 - \sqrt{1 - \lambda})^2}{\lambda} \text{ and } \nu^* = 1 - \sqrt{1 - \lambda}.$$

Notably, we have $\nu^* \geq \lambda/2$ which means the rate with inertia cannot be better than half the original rate.

APPLICATION IN THE SETUP OF EX. 1: *The inertial iteration of T writes*

$$\begin{cases} y_k &= x_k + \gamma_k(x_k - x_{k-1}) \\ x_{k+1} &= y_k - \frac{1}{L}\nabla f(y_k) \end{cases}$$

we have the following optimal inertia parameter

$$\gamma^* = \frac{1 - \sqrt{\mu/L}}{1 + \sqrt{\mu/L}} \text{ and rate } \nu^* = 1 - \sqrt{\mu/L}.$$

Once again, the obtained parameter and rate matches practical and theoretical optimal situations as summarized in [29].

3.3. Alternated Inertia.

3.3.1. Convergence. In order to improve the convergence properties of the iterates of Eq. (3), it was suggested in [26] to apply inertia every other iteration. This variant is a lot less popular than vanilla inertia. However, its rather good convergence properties and performances, along with its remarkable closeness with relaxation make it worthy of careful attention. The iterations of alternated inertia are:

$$(4) \quad \begin{cases} x_{k+1} = \mathsf{T}(x_k) & \text{if } k \text{ is even} \\ x_{k+1} = \mathsf{T}(x_k + \gamma_k(x_k - x_{k-1})) & \text{if } k \text{ is odd} \end{cases}$$

Interestingly, using inertia every other iteration can make the error *monotonously* decreasing again which will reveal to be interesting numerically.

LEMMA 6. *Let $\alpha \in]0, 1[$. Let T be an α -averaged operator such that $\text{fix}\mathsf{T} \neq \emptyset$. Assume that the sequence (γ_k) verifies $0 \leq \gamma_k \leq \frac{1-\alpha}{\alpha}$ for all $k > 0$. Then, the sequence $(x_k)_{k>0}$ generated by $x_0 \in \mathbb{R}^N$ and the iterations*

$$\begin{cases} x_{k+1} = \mathsf{T}(x_k) & \text{if } k \text{ is even} \\ x_{k+1} = \mathsf{T}(x_k + \gamma_k(x_k - x_{k-1})) & \text{if } k \text{ is odd} \end{cases}$$

converges to a point in $\text{fix}\mathsf{T}$.

The proof, which generalizes [26] to α -averaged operators, can be found in Apx. B.

3.3.2. Optimal parameters for real eigenvalues. Let us define $\mathsf{T}^{\cdot\gamma}$ the operator generating x_{k+2} from x_k for k even. When $\mathsf{T} = R \cdot + d$, one has

$$x_{k+2} = \mathsf{T}(\mathsf{T}(x_k) + \gamma(\mathsf{T}(x_k) - x_k)) = \underbrace{[(1 + \gamma)R^2 - \gamma R]}_{R^{\cdot\gamma}} x_k + (1 + \gamma)Rd + d$$

so that the eigenvalues of $R^{\cdot\gamma}$ are $\mu_i = (1 + \gamma)\lambda_i^2 - \gamma\lambda_i$ with λ_i an eigenvalue of R .

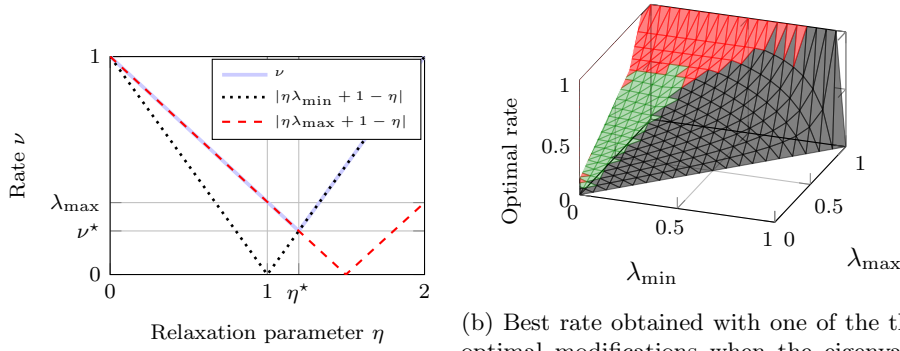
The main results are:

i) for *negatives eigenvalues* $\lambda_i < 0$, the magnitude of μ_i is

$$(1 + \gamma)|\lambda_i|^2 + \gamma|\lambda_i|$$

thus alternated inertia has a negative effect on negative eigenvalues.

ii) for *non-negative eigenvalues* $\lambda_i \in [0, \lambda] \cup \{1\}$, the largest μ_i in magnitude is linked to the original dominant eigenvalue λ and intermediate eigenvalues. Taking the worst



(a) Convergence rate versus the relaxation parameter η . (b) Best rate obtained with one of the three optimal modifications when the eigenvalue interval $[\lambda_{\min}, \lambda_{\max}]$ varies. If the best rate parameter along with tradeoff enabling to de-is attained by relaxation, it is displayed in black; by inertia, in red; by alt. inertia, in green. (for illustration purposes, we assumed $1 \geq \lambda_{\max} \geq \lambda_{\min} \geq 0$)

Fig. 2: Effect of studied modification on linear iterations rate

case scenario over the unknown (and hard to estimate) intermediate eigenvalues, the optimal parameter and rate are

$$\gamma^* = \frac{2(\lambda)^2 + (\sqrt{2} - 1)\lambda}{2\lambda(1 - \lambda) + \frac{1}{2}} \text{ and } \nu^* = \frac{(\gamma^*)^2}{4(1 + \gamma^*)}.$$

APPLICATION IN THE SETUP OF EX. 1: *The alternated inertial iteration of \mathbb{T} writes*

$$\begin{cases} y_k &= x_k + \gamma_k(x_k - x_{k-1}) \\ x_{k+1} &= y_k - \frac{1}{L}\nabla f(y_k) \\ x_{k+2} &= x_{k+1} - \frac{1}{L}\nabla f(x_{k+1}) \end{cases} \Leftrightarrow \begin{cases} x_{k+1} &= x_k - \frac{1}{L}\nabla f(x_k) \\ x_{k+2} &= x_{k+1} - \frac{1+\gamma_{k+2}}{L}\nabla f(x_{k+1}) \end{cases}$$

This formulation properly illustrates the equivalence between alternated inertia and alternated relaxation. We have the following optimal inertia parameter

$$\gamma^* = \frac{2(\mu/L)^2 - (3 + \sqrt{2})\mu/L + 1 + \sqrt{2}}{-2(\mu/L)^2 + 2\mu/L + \frac{1}{2}} \text{ and rate } \nu^* = \frac{(\gamma^*)^2}{4(1 + \gamma^*)}.$$

3.3.3. Comparison of the optimal rates. In general, comparison between relaxation and inertia on linear iterations depends on the interval $[\lambda_{\min}, \lambda_{\max}]$ in which the eigenvalues of the original matrix R live. Fig. 2a provides a graphical illustration of the effect of relaxation on the linear rate. Fig. 2b displays a 3D plot of the optimal rate obtained by numerical simulations (the lower the better) when λ_{\min} and λ_{\max} vary between 0 and 1 along with the modification scheme attaining it.

One can notice that the optimal speed with inertia (alternated or classical) is always faster than with relaxation when $\lambda_{\min} = 0$. For instance, it is the case for the gradient algorithm setup of Ex. 1; the equality case being when $\mu/L = 1$. Between alternated and classical inertia, the alternated version is faster for well enough conditioned problems, more precisely when $1 \geq \mu/L \geq 4/(9 + 4\sqrt{2}) \approx 0.273$; surprisingly making alternated inertia more performing than both inertia and relaxation for some

problems. Also, the optimal parameter for inertia γ^* is greater than theoretical limit $1/3$ as soon as $\mu/L \leq 1/4$. Similarly, for alternated inertia, optimal γ^* is greater than 1 when $\mu/L \leq (3 - \sqrt{2})/4 \approx 0.396$.

Unfortunately, while L can often be known or upper bounded, μ is in general unknown so that the optimal parameters cannot be computed hence the need for automatically tuned schemes as developed further in this paper.

4. Online Acceleration of Linear Rates using Relaxation and Inertia.

In this Section, we provide practical acceleration algorithms for fixed point iterations of general averaged operators using relaxation and inertia.

These methods, that automatically tune relaxation/inertia parameters, are based on affine approximation with real eigenvalues, as investigated in the previous section. This may appear limiting at first but i) in practice, linear approximation of averaged operators often have dominant eigenvalues close to the real line (real eigenvalues are linked to the *cyclic monotonicity property* which appears when considering (sub)-gradients, see [34] and [5, Theo. 22.14]; ii) similar reasoning have been used in recent proofs of inertial algorithms [12]; iii) we prove the iterates convergence in the general averaged operator case (not just affine let alone with real eigenvalues) and iv) our method works very well in practice as demonstrated in Section 5.

For all three modifications, we will iterate in the same steps:

From some acceleration parameter δ ,

- i) Apply the accelerated operator T_δ on the current iterates and estimate its current rate by computing $v_k = \|T_\delta(x_{k-1}) - T_\delta(x_{k-2})\|/\|x_{k-1} - x_{k-2}\|$ as in Sec. 2.2;
- ii) From δ and v_k , construct an approximation of the *virtual* dominant eigenvalue λ of original operator T using the results of the previous section (*virtual* as T is a general non-linear averaged operator, λ is thus linked to an affine approximation of T);
- iii) From λ , update δ as the optimal acceleration parameters previously derived.

4.1. ORM: Online Relaxation Method. Building on the derivations of Sec. 3.1.2, we wish to estimate η^* without having access to the spectrum of R .

- i) To do so, we estimate the current convergence rate as³

$$v_k = (\eta_{k-1}\|x_k - x_{k-1}\|)/(\eta_k\|x_{k-1} - x_{k-2}\|).$$

- ii) Using this v_k , the current relaxation η_k , and the expression for ν^* , we can compute an estimate for virtual dominant eigenvalue λ : $\lambda_k = (v_k + \eta_k - 1)/\eta_k$.

- iii) Using λ_k and optimal η^* , we take our next relaxation parameter as

$$\eta_{k+1} = \frac{2}{2\alpha + 1 - \lambda_k} = \frac{2\eta_k}{2\alpha\eta_k + 1 - v_k}.$$

This gives the intuition for our *Online Relaxation Method (ORM)*.

Online Relaxation Method (ORM) for α-averaged operator T :
--

<u>Initialization:</u> $\varepsilon \in]0, 2 \min(\alpha; 1 - \alpha)]$, $x^0, x^1 = Tx^0$, $\eta^0 = \eta^1 = 1$.
--

³Note the extra factor η_{k-1}/η_k compared to Sec. 2.2. In the specific case of relaxation, this modified definition enables to estimate the convergence of T_{η_k} by applying it only once. Monotone operators theory ensures us that $v_k \in [0, 1]$, and enables the convergence proof.

At each iteration $k \geq 1$:

$$\begin{aligned}\eta_{k+1} &= \frac{(2 - \varepsilon)\eta_k}{2\alpha\eta_k + 1 - \frac{\eta_{k-1}\|x_k - x_{k-1}\|}{\eta_k\|x_{k-1} - x_{k-2}\|}} + \frac{\varepsilon}{4\alpha} \\ x_{k+1} &= \eta_{k+1}\mathsf{T}x_k + (1 - \eta_{k+1})x_k\end{aligned}$$

The following result provides convergence guarantees for this method in the general framework of averaged operators.

THEOREM 7. *Let $\alpha \in]0, 1[$. Let T be an α -averaged operator such that $\text{fix}\mathsf{T} \neq \emptyset$. Then, the sequence $(x_k)_{k \geq 0}$ generated by the Online Relaxation Method converges to a point in $\text{fix}\mathsf{T}$.*

Proof. In order to use Lemma 4 to prove the convergence, let us prove by induction that for all $k \geq 1$, $\eta_k \in [\frac{\varepsilon}{4\alpha}, \frac{1}{\alpha} - \frac{\varepsilon}{4\alpha}]$. It is obviously true for $\eta^1 = 1$. Let us assume that $\eta_k \in [\frac{\varepsilon}{4\alpha}, \frac{1}{\alpha} - \frac{\varepsilon}{4\alpha}]$.

First, as T is α -averaged, it writes $\mathsf{T} = \alpha\mathsf{R} + (1 - \alpha)\mathsf{I}$ with R a non-expansive operator and I the identity. T_{η_k} then writes $\mathsf{T}_{\eta_k} = \alpha\eta_k\mathsf{R} + (1 - \alpha\eta_k)\mathsf{I}$, thus

$$\begin{aligned}\|x_k - x_{k-1}\| &= \alpha\eta_k\|\mathsf{R}(x_{k-1}) - x_{k-1}\| \\ &= \alpha\eta_k\|\mathsf{R}(x_{k-1}) - \mathsf{R}(x_{k-2}) + (1 - \alpha\eta_{k-1})(\mathsf{R}(x_{k-2}) - x_{k-2})\| \\ &\leq \alpha\eta_k\|x_{k-1} - x_{k-2}\| + \alpha\eta_k(1 - \alpha\eta_{k-1})\|\mathsf{R}(x_{k-2}) - x_{k-2}\| \\ &= \alpha\eta_k\|x_{k-1} - x_{k-2}\| + \alpha\eta_k\frac{1 - \alpha\eta_{k-1}}{\alpha\eta_{k-1}}\|x_{k-1} - x_{k-2}\| = \frac{\eta_k}{\eta_{k-1}}\|x_{k-1} - x_{k-2}\|\end{aligned}$$

Thus, we have $v_k \leq 1$ which makes $\eta_{k+1} \geq \frac{\varepsilon}{4\alpha}$. Now,

$$\eta_{k+1} \leq \frac{(2 - \varepsilon)\eta_k}{2\alpha\eta_k} + \frac{\varepsilon}{4\alpha} = \frac{1}{\alpha} - \frac{\varepsilon}{2\alpha} + \frac{\varepsilon}{4\alpha} = \frac{1}{\alpha} - \frac{\varepsilon}{4\alpha}$$

thus we have that $\eta_{k+1} \in [\frac{\varepsilon}{4\alpha}, \frac{1}{\alpha} - \frac{\varepsilon}{4\alpha}]$. This means the generated sequence $(\eta_k)_{k \geq 0}$ lies in $[\frac{\varepsilon}{4\alpha}, \frac{1}{\alpha} - \frac{\varepsilon}{4\alpha}]$ and thus verifies the conditions of Lemma 4 for convergence. \square

Interestingly, one can notice that when the basis algorithm converges sub-linearly, the paramter chosen by ORM becomes close to $2/L$. For the gradient algorithm, this would amount to having a stepsize that becomes close to $2/L$ as the number of iterations grow which is coherent with the optimality results in [36, Sec. 4.1.1]⁴.

4.2. OIM: Online Inertia Method. An online inertia method can be proposed based on the same principles as ORM building on Sec. 3.2.2. In the same vein, we approximate the operation $\mathsf{T}^{\gamma_{2k}}$ by an affine operator with non-negative eigenvalues. i) We estimate the current convergence rate related to operator $\mathsf{T}^{\gamma_{2k}}$ (by applying twice the same inertia twice) as

$$v_{2k} = \sqrt{\frac{\|x_{2k+2} - x_{2k+1}\|^2 + \|x_{2k+1} - x_{2k}\|^2}{\|x_{2k+1} - x_{2k}\|^2 + \|x_{2k} - x_{2k-1}\|^2}}.$$

ii) Using v_{2k} , current inertia γ_{2k} , we estimate λ : $\lambda_{2k} = ((v_{2k})^2)/(\gamma_{2k}v_{2k} - \gamma_{2k} + v_{2k})$. iii) Using λ_{2k} and the formula for optimal γ^* , we take our next relaxation parameter as $\gamma_{2k+2} = (1 - \sqrt{1 - \lambda_{2k}})^2/\lambda_{2k}$.

These steps are at the core of our *Online Inertia Method (OIM)*. However, to the difference of ORM but similarly to other inertia-based accelerations [15], a restart

⁴The optimal stepsize when doing K iterations goes to $2/L$, staying strictly below, when $K \rightarrow \infty$.

mechanism has to be introduced to make sure the algorithm converges. Indeed, this scheme, which is rather aggressive, often overpasses the theoretical limits of the convergence results. Thus, in order to maintain convergence, the algorithm must either i) sufficiently decrease the error $\|x_k - y_k\|$; or ii) set inertial parameter γ_k to 0 so that classical convergence results apply.

Online Inertia Method (OIM) for α-averaged operator T:	
<u>Initialization:</u> $x_1, x_2 = \mathsf{T}(x_1), y_2 = x_1, \gamma_2 = 0, \varepsilon > 0$.	
For each $k \geq 1$:	
$\begin{cases} y_{2k+1} = x_{2k} + \gamma_{2k}(x_{2k} - x_{2k-1}) \\ x_{2k+1} = \mathsf{T}(y_{2k+1}) \\ y_{2k+2} = x_{2k+1} + \gamma_{2k}(x_{2k+1} - x_{2k}) \\ x_{2k+2} = \mathsf{T}(y_{2k+2}) \end{cases}$	
$c_{2k} = \max \left(\frac{\ x_{2k+2} - y_{2k+2}\ }{\ x_{2k+1} - y_{2k+1}\ }, \frac{\ x_{2k+1} - y_{2k+1}\ }{\ x_{2k} - y_{2k}\ } \right)$	
if $c_{2k} \leq 1 - \varepsilon$ [Acceleration]	
$v_{2k} = \sqrt{\frac{\ x_{2k+2} - x_{2k+1}\ ^2 + \ x_{2k+1} - x_{2k}\ ^2}{\ x_{2k+1} - x_{2k}\ ^2 + \ x_{2k} - x_{2k-1}\ ^2}}$	
$\lambda_{2k} = \min \left(\frac{(v_{2k})^2}{\gamma_{2k} v_{2k} - \gamma_{2k} + v_{2k}}; 1 - \varepsilon \right)$	
$\gamma_{2k+2} = \max \left(0; \frac{(1 - \sqrt{1 - \lambda_{2k}})^2}{\lambda_{2k}} \right)$	
elseif $\gamma_{2k} > 0$ [Restart]	
$\gamma_{2k+2} = 0$	
$(x_{2k+1}, x_{2k+2}, y_{2k+2}) = (x_{2k-1}, x_{2k}, y_{2k})$	
elseif $\gamma_{2k} = 0$ [No Acceleration]	
$\gamma_{2k+2} = 0$	

THEOREM 8. *Let $\alpha \in]0, 1[$. Let T be an α -averaged operator such that $\text{fix}\mathsf{T} \neq \emptyset$. Then, the sequence $(y_k)_{k>0}$ generated by the Online Inertia Method converges in the sense that $\|\mathsf{T}(y_k) - y_k\| \rightarrow 0$. Furthermore, if $\text{fix}\mathsf{T}$ is reduced to a single point x^* , $x_k \rightarrow x^*$.*

Proof. The proof follows the same reasoning as [15, Theo. 3]. At each iteration, one of the following situation happens:

- i) the last iteration was beneficial: $c_k \leq 1 - \varepsilon$ so that $\|x_k - y_k\| \leq (1 - \varepsilon)\|x_{k-1} - y_{k-1}\|$ and $\|x_{k-1} - y_{k-1}\| \leq (1 - \varepsilon)\|x_{k-2} - y_{k-2}\|$;
- ii) a restart is made so that the iterates x_k and x_{k-1} by their previous values $\|x_k - y_k\| = \|x_{k-2} - y_{k-2}\|$ and $\|x_{k-1} - y_{k-1}\| = \|x_{k-3} - y_{k-3}\|$;
- iii) there is no acceleration and non expansiveness gives $\|x_k - y_k\| \leq \|x_{k-1} - y_{k-1}\| \leq \|x_{k-2} - y_{k-2}\|$.

To conclude the proof, one has to notice that for all $k > 0$, $\|x_k - y_k\| \leq \|x_{k-1} - y_{k-1}\|$ and $\|x_k - y_k\| \leq (1 - \varepsilon)\|x_{k-1} - y_{k-1}\|$ if i) happens. Now, if there is a finite number of beneficial iterations (when i) happens), then after the last one, the algorithm goes back to the unaccelerated iterations and convergence is ensured by Lemma 1. If there is a infinite number of beneficial iterations, introducing variable ι_k as $\iota_k = 1$ if iteration k is beneficial and 0 elsewhere; we have

$$\sum_{k=1}^{\infty} \iota_k \|\mathsf{T}(y_k) - y_k\| \leq \|\mathsf{T}(x^0) - x^0\| \sum_{k=1}^{\infty} \prod_{\ell=1}^k (1 - \varepsilon)^{\iota_\ell} \leq \frac{\|\mathsf{T}(x^0) - x^0\|}{\varepsilon} < \infty$$

and thus $\|\mathsf{T}(y_k) - y_k\| \rightarrow 0$. This means that the accumulation points of (y_k) are in $\text{fix}\mathsf{T}$. In addition, if it is reduced to a single point, then (y_k) converges to it and as $\|x_k - y_k\| \rightarrow 0$, so does (x_k) . \square

When the convergence is sublinear, the restart condition based on a constant ε may be too harsh. Following the convergence proof, one can easily deduce that ε can be taken as a sequence (ε^ℓ) provided that $1/\varepsilon^\ell = o(\ell)$ where ℓ is the number of accelerations. For instance, a typical setting is to keep track of the number of accelerations ℓ and take $\varepsilon^\ell = \varepsilon_0/\sqrt{\ell}$. Note that in the sublinear case, the OIM makes the acceleration parameter go to 1 as in Nesterov's optimal method [27].

4.3. OAIM: Online Alternated Inertia Method. Using the same reasoning as for OIM, we are able to obtain a similar algorithm.

Online Alternated Inertia Method (OAIM) for an α-averaged operator T:	
<u>Initialization:</u> $x_3, x_4 = \mathsf{T}(x_3), y_4 = x_3, \gamma_4 = 0, \varepsilon > 0$.	
For each $k \geq 1$:	
$\begin{cases} y_{4k+1} = x_{4k} + \gamma_{4k}(x_{4k} - x_{4k-1}) \\ x_{4k+1} = \mathsf{T}(y_{4k+1}) \\ y_{4k+2} = x_{4k+1} \\ x_{4k+2} = \mathsf{T}(y_{4k+2}) \end{cases}$	$\begin{cases} y_{4k+3} = x_{4k+2} + \gamma_{4k}(x_{4k+2} - x_{4k+1}) \\ x_{4k+3} = \mathsf{T}(y_{4k+3}) \\ y_{4k+4} = x_{4k+3} \\ x_{4k+4} = \mathsf{T}(y_{4k+4}) \end{cases}$
$c_{4k} = \max\left(\frac{\ x_{4k+4} - x_{4k+3}\ }{\ x_{4k+2} - x_{4k+1}\ }, \frac{\ x_{4k+2} - x_{4k+1}\ }{\ x_{4k} - x_{4k-1}\ }\right)$	
<div style="display: flex; justify-content: space-between;"> if $c_{4k} \leq 1 - \varepsilon$ [Acceleration] </div>	
$v_{4k} = \frac{\ x_{4k+4} - x_{4k+2}\ }{\ x_{4k+2} - x_{4k}\ }$	
$\lambda_{4k} = \min\left(\frac{\gamma_{4k} + \sqrt{(\gamma_{4k})^2 + 4\gamma_{4k}v_{4k} + 4v_{4k}}}{2(\gamma_{4k} + 1)}; 1 - \varepsilon\right)$	
$\gamma_{4k+4} = \frac{2(\lambda_{4k})^2 + (\sqrt{2}-1)\lambda_{4k}}{2\lambda_{4k}(1-\lambda_{4k}) + \frac{1}{2}}$	
<div style="display: flex; justify-content: space-between;"> elseif $\gamma_{4k} > 0$ [Restart] </div>	
$\gamma_{4k+4} = 0$	
$(x_{4k+3}, x_{4k+4}) = (x_{4k-1}, x_{4k})$	
<div style="display: flex; justify-content: space-between;"> elseif $\gamma_{4k} = 0$ [No Acceleration] </div>	
$\gamma_{4k+4} = 0$	

THEOREM 9. *Let $\alpha \in]0, 1[$. Let T be an α -averaged operator such that $\text{fix}\mathsf{T} \neq \emptyset$. Then, the sequence $(x_k)_{k \geq 0}$ generated by the Online Alternated Inertia Method converges in the sense that $\|\mathsf{T}(x^{2k}) - x^{2k}\| \rightarrow 0$. Furthermore, if $\text{fix}\mathsf{T}$ is reduced to a single point x^* , $x_k \rightarrow x^*$.*

Proof. The proof follow the same steps as the proof of Theo. 8. \square

5. Relaxation and Inertia of Optimization algorithms. We now particularize the operator T to different values corresponding to popular algorithms of the literature. We illustrate the interest of the modifications studied and, most importantly, we demonstrate the acceleration provided by our online methods over three popular algorithms: the proximal gradient algorithm, the ADMM, and a Primal-Dual algorithm by Condat [10].

For each of these algorithms, we will proceed in the same fashion:

- 1) We discuss how relaxation and inertia translate for these algorithms along with a review on existing accelerated versions;
- 2) We provide numerical illustrations over the three following functions chosen for

their differences in terms of smoothness and strong convexity:

a) lasso:

$$\min_{x \in \mathbb{R}^n} F_a(x) = \underbrace{\frac{1}{2} \|Ax - b\|_2^2}_{f_a(x)} + \underbrace{\lambda \|x\|_1}_{g_a(x)}$$

where A has $m = 100$ examples and $n = 300$ observations taken from the normal distribution with zero mean and unit variance, the columns of A are then scaled to have unit norm. b is generated by i) drawing a sparse vector $p \in \mathbb{R}^n$ with 90 non-zeros entries taken from the normal distribution with zero mean and unit variance; ii) then creating b as $b = Ap + e$ where e is a small white noise taken from the normal distribution with zero mean and standard deviation $\sigma = 0.001$. λ is chosen so that the optimal solution has sought sparsity. Lipschitz constant of ∇f_a is taken equal to true $L = \|A^T A\|_2$.

b) ℓ_1 -regularized logistic regression:

$$\min_{x \in \mathbb{R}^n} F_b(x) = \underbrace{\frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-y_i \langle a_i, x \rangle))}_{f_b(x)} + \underbrace{\lambda \|x\|_1}_{g_b(x)}$$

where the couples class/feature vector $(y_i, a_i) \in \{-1, 1\} \times \mathbb{R}^n$ are taken from the `ionosphere` binary classification dataset⁵ which has $m = 351$ observations and $n = 34$ features. Each feature was normalized to have zero mean and unit variance, the resulting size- n observation vectors are denoted by $(a_i)_{i=1, \dots, m}$ and the observed classes $-1, +1$ are denoted by $(y_i)_{i=1, \dots, m}$. Lipschitz constant of ∇f_b is upper bounded by $L' = \max_i \|a_i\|_2^2$. λ was taken equal to 0.1.

c) ℓ_2 -regularized logistic regression:

$$\min_{x \in \mathbb{R}^n} F_c(x) = \underbrace{\frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-y_i \langle a_i, x \rangle))}_{f_c(x)} + \underbrace{\frac{\lambda}{2} \|x\|_2^2}_{g_c(x)}$$

where the couples class/feature vector $(y_i, a_i) \in \{-1, 1\} \times \mathbb{R}^n$ are taken from the same dataset, and λ was taken equal to 0.01.

For these three functions we computed approximated optimal values by external solvers. For the online algorithms, the convergence-ensuring ε is set to 10^{-4}

5.1. Proximal Gradient Algorithm.

Proximal Gradient algorithm for $\min_x f(x) + g(x)$, f L -smooth. _____

$$x_{k+1} = \operatorname{argmin}_w \left\{ g(w) + \frac{L}{2} \left\| w - x_k + \frac{1}{L} \nabla f(x_k) \right\|^2 \right\}$$

5.1.1. Accelerations. It is straightforward to see that an iteration of the algorithm writes as fixed point iteration $x_{k+1} = T_{pg}(x_k)$ and monotone operator theory tells us that T_{pg} is 2/3-averaged [5, Chap. 27.3]. The application of both relaxation and inertia on top of this algorithm is thus easy.

⁵<https://archive.ics.uci.edu/ml/datasets/Ionosphere>

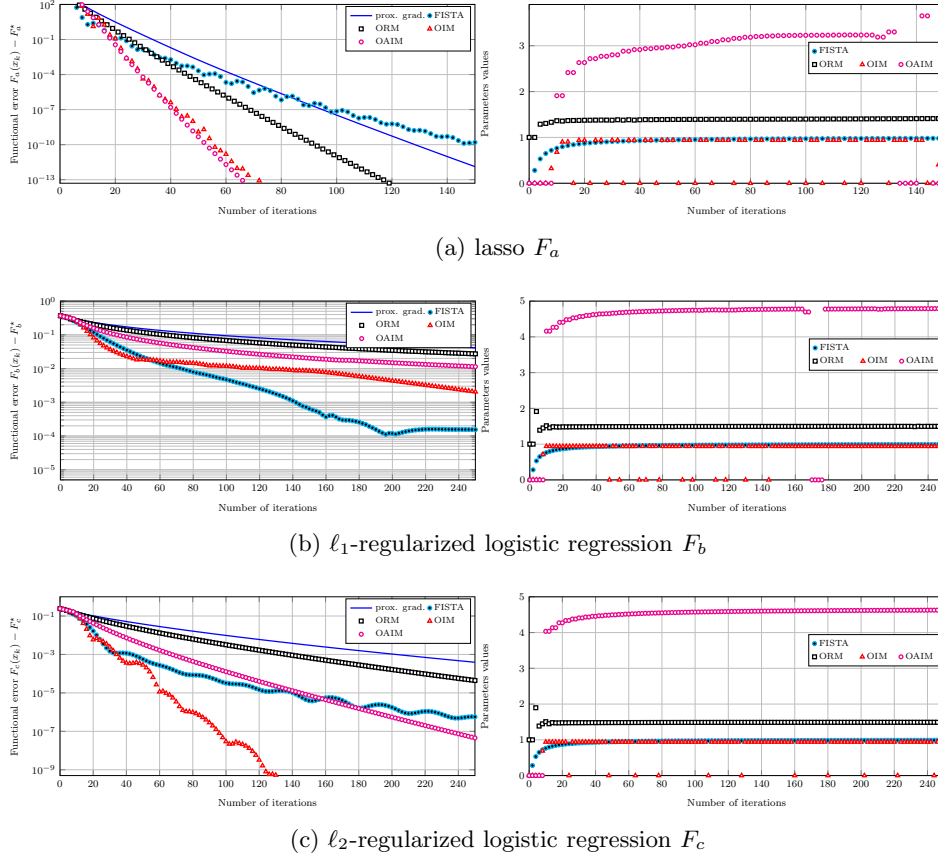


Fig. 3: Proximal Gradient

In fact, the proximal gradient algorithm possesses a very popular inertial version with the popular FISTA method (proximal gradient + Nesterov's acceleration) [6].

5.1.2. Numerical Illustrations. In Fig. 3, we plot the functional error and the parameters for i) classical proximal gradient algorithm; ii) FISTA; iii) our three online methods (we implemented OIM and OAIM as if α was $1/2$, in coherence with the choice for FISTA). We observe that all proposed algorithms show good behaviors, the less favorable case being b as neither functions exhibit strong convexity. Inertia-based methods perform very well: OIM outperforms FISTA except in case b and OAIM performs quite well. We notice significant behavioral differences between inertial methods (OIM, FISTA) which show bounces in the error descent, contrary to OAIM which is much more stable with almost no use of restart, and ORM which is provably monotonous.

5.2. Alternating Direction Method of Multipliers. Consider the following optimization problem:

$$(5) \quad \min_{x \in \mathbb{R}^N} f(x) + g(Mx)$$

with f, g two convex lower semi-continuous functions and M a linear operator. The Alternating Direction Method of Multipliers (ADMM) addresses this problem by performing the following iterations with free parameter $\rho > 0$.

ADMM

$$\begin{aligned} u_{k+1} &= \underset{w}{\operatorname{argmin}} \left\{ f(w) + \frac{\rho}{2} \left\| Mw - v_k + \frac{\lambda_k}{\rho} \right\|^2 \right\} \\ v_{k+1} &= \underset{w}{\operatorname{argmin}} \left\{ g(w) + \frac{\rho}{2} \left\| Mu_{k+1} - w + \frac{\lambda_k}{\rho} \right\|^2 \right\} \\ \lambda_{k+1} &= \lambda_k + \rho(Mu_{k+1} - v_{k+1}) \end{aligned}$$

5.2.1. Accelerations. From an operator point of view, the iterations of ADMM can be seen as updates on the meta-variable $x_k = \lambda_k + \rho v_k = \lambda_{k-1} + \rho Mu_k$ of an $1/2$ -averaged operator T_{admm} (see [11] and references therein for details). This meta-variable is central as it affects the way relaxation and inertia translates for this algorithm.

Relaxation While it is fairly evident to see that the relaxed version of the operation writes $x_{k+1} = \eta \mathsf{T}_{admm}(x_k) + (1 - \eta)x_k$, it is slightly more complex to derive the effect of relaxation on the algorithm variables (u_k, v_k, λ_k) . Indeed, these variables are computed by a *representation* of the meta-variable that is non-linear. Let us call J_v the operation giving v_k from x_k , then

$$(6) \quad v_{k+1} = \mathsf{J}_v(x_{k+1}) = \mathsf{J}_v(\eta \mathsf{T}_{admm}(x_k) + (1 - \eta)x_k) \neq \eta \mathsf{J}_v(\mathsf{T}_{admm}(x_k)) + (1 - \eta)\mathsf{J}_v(x_k).$$

This means that, in general⁶ relaxation *cannot* be added directly *on top* of ADMM in the sense performing the standard ADMM update then adding a step of the form $v_{k+1} \leftarrow \eta v_{k+1} + (1 - \eta)v_k$ and $\lambda_{k+1} \leftarrow \eta \lambda_{k+1} + (1 - \eta)\lambda_k$.

Following the operator vision, the canonical relaxation on the ADMM leads to the following iterations (derivations can be found in [11]); with x_k being the meta-variable that is Féjer monotonous, and used in ORM for instance.

Relaxed ADMM

$$\begin{aligned} u_{k+1} &= \underset{w}{\operatorname{argmin}} \left\{ f(w) + \frac{\rho}{2} \left\| Mw - z_k + \frac{\lambda_k}{\rho} \right\|^2 \right\} \\ v_{k+1} &= \underset{w}{\operatorname{argmin}} \left\{ g(w) + \frac{\rho}{2} \left\| \eta Mu_{k+1} + (1 - \eta)v_k - w + \frac{\lambda_k}{\rho} \right\|^2 \right\} \\ \lambda_{k+1} &= \lambda_k + \rho(\eta Mu_{k+1} + (1 - \eta)v_k - v_{k+1}) \\ x_{k+1} &= \lambda_{k+1} + \rho v_{k+1} \end{aligned}$$

It was noted in [11] that “experiments [...] suggest that over-relaxation with $\eta \in [1.5, 1.8]$ can improve convergence” without further details. One can also mention [14] based on relaxation tuning by line search. Our ORM, with its particularly stable behavior bridges nicely the literature in this respect.

Inertia As previously, *inertial ADMM* cannot be derived simply by adding inertia *on top* of the above iterations. Following the operator vision, the canonical inertial

⁶If either i) g is the indicator function of a linear space, or ii) when g is quadratic; then the representation operation J_v of Eq. (6) becomes linear and relaxation can be performed as an *outer* modification.

version of the ADMM leads to the following iterations; with x_k and y_k being the meta-variables as in Eq. (3). See Apx. C for the derivation. To the best of our knowledge, this is an original algorithm.

Inertial ADMM

$$\begin{aligned}
u_{k+1} &= \underset{w}{\operatorname{argmin}} \left\{ f(w) + \frac{\rho}{2} \left\| Mw - v_k + \frac{\lambda_k}{\rho} \right\|^2 \right\} \\
x_{k+1} &= \lambda_k + \rho M u_{k+1} \\
v_{k+1} &= \underset{w}{\operatorname{argmin}} \left\{ g(w) + \frac{\rho}{2} \left\| M u_{k+1} - w + \frac{\lambda_k}{\rho} + \gamma \left(M(u_{k+1} - u_k) + \frac{\lambda_k - \lambda_{k-1}}{\rho} \right) \right\|^2 \right\} \\
\lambda_{k+1} &= \lambda_k + \rho(M u_{k+1} - v_{k+1}) + \gamma \rho \left(M(u_{k+1} - u_k) + \frac{\lambda_k - \lambda_{k-1}}{\rho} \right) \\
y_{k+1} &= \lambda_{k+1} + \rho v_{k+1}
\end{aligned}$$

As for relaxation, if g is either i) the indicator function of a linear space, or ii) quadratic; inertia can be performed as an *outer* modification. Note that ADMM + outer inertia with Nesterov-like parameter sequence corresponds to the algorithm named *Fast ADMM* studied in [15]. However, this algorithm is not convergent in the general case, unless a restart scheme is added. Interestingly, for the convergence proof of Fast ADMM in the strongly convex case, g is assumed quadratic.

Alternated Inertia It simply consists in alternating an iteration of ADMM with an iteration of Inertia ADMM. One can remark, that with this proper formulation of relaxed and inertial ADMM, applying inertia or relaxation every other iteration provably gives the same algorithm (use the fact that $\lambda_k - \lambda_{k-1} = (M u_k - v_k)/\rho$ for a standard ADMM iteration in the Inertial ADMM iteration). To the best of our knowledge, this kind of algorithm has never been considered before.

5.2.2. Numerical illustrations. In Fig. 4, we compare i) the standard ADMM; ii) our three proposed online methods; and iii) Fast ADMM with restart [15]. In all cases, the ADMM parameter ρ was set to 1. For logistic regression functions f_b and f_c , no explicit formulation of the update of the first variable is available so their have to be computed by an external solver (SciPy’s general-purpose `minimize` function in our case). We observe that, once again, the proposed online methods show remarkable performance for their computational cost. OIM performs best; however, ORM and OAIM, contrary to OIM and Fast ADMM show *steady* parameter sequences, this can be seen as *more monotonous behaviors*. Finally, ORM offers a better alternative to arbitrarily fixed relaxation.

5.3. a Primal Dual Algorithm. We investigate the primal-dual algorithm 3.1 from [10] with $F = 0$. For this algorithm, we will consider only⁷ the lasso problem F_a as it can be implemented so that, contrary to the ADMM, no matrix inversion is performed, with $M = A$, $g(\cdot) = 1/2 \|\cdot - b\|^2$ and $f(\cdot) = \lambda \|\cdot\|_1 = g_a(\cdot)$. We chose $\tau = 0.5$ and $\sigma = 1/(\tau \|A\|^2)$ as prescribed.

⁷for the other two problems, the algorithm boils down to previously investigated ADMM.

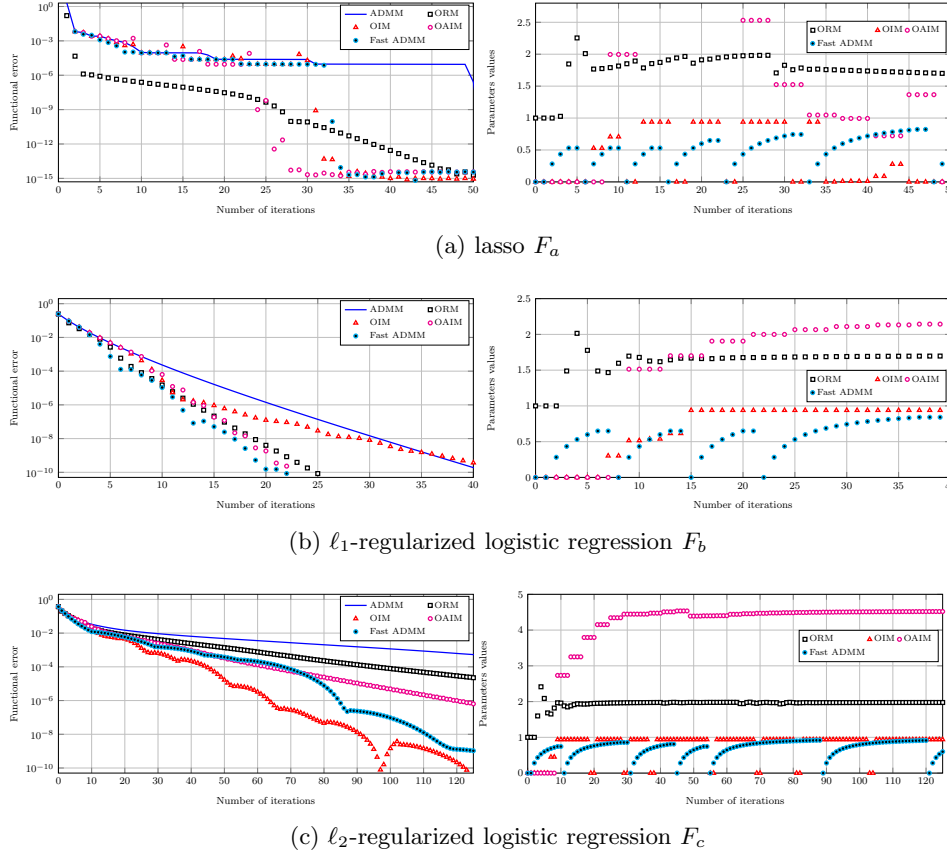


Fig. 4: Alternating Direction Method of Multipliers (ADMM)

a Primal-Dual algorithm [10, Alg. 3.1] for $\min_x f(x) + g(Mx)$

$$u_{k+1} = \underset{w}{\operatorname{argmin}} \left\{ f(w) + \frac{1}{2\tau} \left\| w - u_k + \tau M^T \lambda_k \right\|^2 \right\}$$

$$\lambda_{k+1} = \lambda_k + \sigma M(2u_{k+1} - u_k) - \sigma \underset{w}{\operatorname{argmin}} \left\{ h(w) + \frac{\sigma}{2} \left\| w - \frac{\lambda_k}{\sigma} - M(2u_{k+1} - u_k) \right\|^2 \right\}$$

With the prescribed choice of parameters, defining $x_k = [u_k; \lambda_k]$ as the stacked vector of the variables, the algorithm is a fixed point algorithm on x_k with an $1/2$ -averaged operator. Relaxation and Inertia can be simply performed as outer-modifications of the algorithm.

In Fig. 5, we plot the functional error and the parameters for the original algorithm and our three online methods. The formulation of all algorithms are again quite simple and we obtain significant speed improvements.

6. Conclusion. In this paper, we investigated the theoretical and practical interests of relaxation and inertia on averaged operators. Notably, we established the expression for optimal parameters and rate when possible and built upon it to pro-

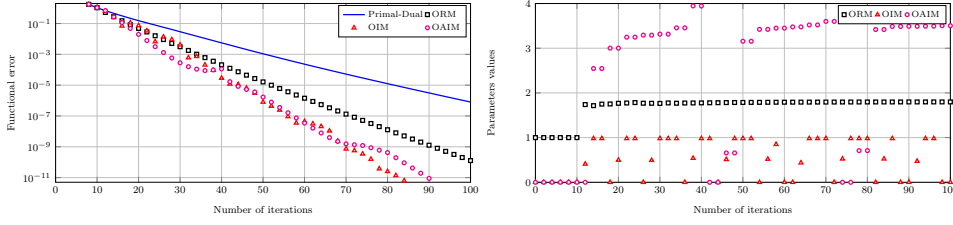


Fig. 5: a Primal-Dual algorithm on a lasso problem

pose novel online methods. Numerical illustrations have demonstrated the behavioral differences between relaxation and inertia and showed the remarkable performance of the proposed online methods.

REFERENCES

- [1] F. ALVAREZ, *Weak convergence of a relaxed and inertial hybrid projection-proximal point algorithm for maximal monotone operators in hilbert space*, SIAM Journal on Optimization, 14 (2004), pp. 773–782.
- [2] F. ALVAREZ AND H. ATTOUCH, *An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping*, Set-Valued Analysis, 9 (2001), pp. 3–11.
- [3] H. ATTOUCH AND J. PEYPOUQUET, *The rate of convergence of nesterov’s accelerated forward-backward method is actually $o(k^{-2})$* , arXiv preprint arXiv:1510.08740, (2015).
- [4] H. BAUSCHKE, J. BELLO CRUZ, T. NGHIA, H. PHAN, AND X. WANG, *Optimal rates of linear convergence of relaxed alternating projections and generalized douglas-rachford methods for two subspaces.*, Numerical Algorithms, (in press), pp. 1–44.
- [5] H. BAUSCHKE AND P. L. COMBETTES, *Convex analysis and monotone operator theory in Hilbert spaces*, CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, Springer, New York, 2011.
- [6] A. BECK AND M. TEOULLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM journal on imaging sciences, 2 (2009), pp. 183–202.
- [7] P. BIANCHI, W. HACHEM, AND F. IUTZELER, *A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization*, arXiv preprint arXiv:1407.0898, (2014).
- [8] S. BOYD, N. PARIKH, E. CHU, B. PELEATO, AND J. ECKSTEIN, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends in Machine Learning, 3 (2011), pp. 1–122.
- [9] A. CHAMBOLLE AND C. DOSSAL, *On the convergence of the iterates of “fista”.*, Preprint hal-01060130, September, (2014).
- [10] L. CONDAT, *A primal-dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms*, Journal of Optimization Theory and Applications, 158 (2013), pp. 460–479.
- [11] J. ECKSTEIN AND D. P. BERTSEKAS, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Mathematical Programming, 55 (1992), pp. 293–318.
- [12] N. FLAMMARION AND F. BACH, *From averaging to acceleration, there is only a step-size*, arXiv preprint arXiv:1504.01577, (2015).
- [13] E. GHADIMI, A. TEIXEIRA, I. SHAMES, AND M. JOHANSSON, *Optimal parameter selection for the alternating direction method of multipliers (admm): Quadratic problems*, arXiv preprint arXiv:1306.2454, (2013).
- [14] P. GISELSSON, M. FÄLT, AND S. BOYD, *Line search for averaged operator iteration*, arXiv preprint arXiv:1603.06772, (2016).
- [15] T. GOLDSTEIN, B. O’DONOGHUE, S. SETZER, AND R. BARANIUK, *Fast alternating direction optimization methods*, SIAM Journal on Imaging Sciences, 7 (2014), pp. 1588–1623.
- [16] R. A. HORN AND C. R. JOHNSON, *Matrix analysis*, Cambridge University Press, 2007.
- [17] F. IUTZELER, P. BIANCHI, P. CIBLAT, AND W. HACHEM, *Asynchronous distributed optimization*

- using a randomized Alternating Direction Method of Multipliers, in Proc. IEEE Conf. Decision and Control (CDC), Florence, Italy, Dec. 2013.
- [18] F. IUTZELER, P. BIANCHI, P. CIBLAT, AND W. HACHEM, *Explicit convergence rate of a distributed alternating direction method of multipliers*, arXiv preprint arXiv:1312.1085, (2013).
 - [19] F. IUTZELER, P. CIBLAT, AND W. HACHEM, *Analysis of sum-weight-like algorithms for averaging in wireless sensor networks*, IEEE Transactions on Signal Processing, 61 (2013), pp. 2802–2814.
 - [20] P. JOHNSTONE AND P. MOULIN, *A lyapunov analysis of fista with local linear convergence for sparse optimization*, arXiv preprint arXiv:1502.02281, (2015).
 - [21] H. LIN, J. MAIRAL, AND Z. HARCHAOUI, *A Universal Catalyst for First-Order Optimization*, ArXiv e-prints, (2015), [arXiv:1506.02186](#).
 - [22] Q. LIN AND L. XIAO, *An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization*, Computational Optimization and Applications, 60 (2014), pp. 633–674.
 - [23] P.-L. LIONS AND B. MERCIER, *Splitting algorithms for the sum of two nonlinear operators*, SIAM Journal on Numerical Analysis, 16 (1979), pp. 964–979.
 - [24] D. LORENZ AND T. POCK, *An inertial forward-backward algorithm for monotone inclusions*, Journal of Mathematical Imaging and Vision, 51 (2014), pp. 311–325.
 - [25] P.-E. MAINGÉ, *Convergence theorems for inertial km-type algorithms*, Journal of Computational and Applied Mathematics, 219 (2008), pp. 223–236.
 - [26] Z. MU AND Y. PENG, *A note on the inertial proximal point method*, Statistics, Optimization & Information Computing, 3 (2015), pp. 241–248.
 - [27] Y. NESTEROV, *A method of solving a convex programming problem with convergence rate $o(1/k^2)$* , Soviet Mathematics Doklady, 27 (1983), pp. 372–376.
 - [28] Y. NESTEROV, *Smooth minimization of non-smooth functions*, Mathematical programming, 103 (2005), pp. 127–152.
 - [29] B. O'DONOGHUE AND E. CANDÈS, *Adaptive restart for accelerated gradient schemes*, Foundations of computational mathematics, 15 (2013), pp. 715–732.
 - [30] B. POLYAK, *Some methods of speeding up the convergence of iteration methods*, USSR Computational Mathematics and Mathematical Physics, 4 (1964), pp. 1–17.
 - [31] L. F. RICHARDSON, *The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam*, Philosophical Transactions of the Royal Society of London, (1911), pp. 307–357.
 - [32] Y. SAAD, *Iterative methods for sparse linear systems*, Siam, 2003.
 - [33] W. SHI, Q. LING, K. YUAN, G. WU, AND W. YIN, *On the Linear Convergence of the ADMM in Decentralized Consensus Optimization*, ArXiv e-prints, (2013), [arXiv:1307.5561](#).
 - [34] E. SHIU, *Cyclically monotone linear operators*, Proceedings of the American Mathematical Society, 59 (1976), pp. 127–132.
 - [35] S. TAO, D. BOLEY, AND S. ZHANG, *Local linear convergence of ista and fista on the lasso problem*, arXiv preprint arXiv:1501.02888, (2015).
 - [36] A. TAYLOR, J. HENDRICKX, AND F. GLINEUR, *Smooth strongly convex interpolation and exact worst-case performance of first-order methods*, arXiv preprint arXiv:1502.05666, (2015).
 - [37] P. TSENG, *On accelerated proximal gradient methods for convex-concave optimization*, submitted to SIAM Journal on Optimization, (2008).
 - [38] E. WEI AND A. OZDAGLAR, *On the $O(1/k)$ convergence of asynchronous distributed Alternating Direction Method of Multipliers*, arXiv preprint arXiv:1307.8254, (2013).

Appendix A. Proof of the linear behavior of affine averaged operators (Sec. 2.2).

We consider the fixed point iterations $x_{k+1} = \mathsf{T}(x_k) = Rx_k + d$ with $\mathsf{T} = R \cdot + d$ an affine α -averaged operator. We assume that $\text{fix}\mathsf{T} \neq \emptyset$ that is, d lives in the column space of $I - R$.

Let us denote by \mathcal{N} the nullspace of $I - R$: $\mathcal{N} \triangleq \{x \in \mathbb{R}^N : Rx = x\}$. Any fixed point of T can be expressed as one particular fixed point plus a vector in \mathcal{N} .

Consider the Jordan decomposition of matrix R : $R = W\Lambda W^{-1}$ with W a non-singular matrix and Λ the Jordan block-diagonal for R (see [16, Chap. 3]). The proof of Lemma 1 (see [5, Prop. 5.15]) tells that $\sum_{k=0}^{+\infty} \|x_k - \mathsf{T}(x_k)\|^2 < \infty$ so

$$\sum_{k=0}^{+\infty} \|R_k(R - I)x^0 + R_k d\|^2 = \sum_{k=0}^{+\infty} \|W(\Lambda_k(\Lambda - I)W^{-1}x^0 + \Lambda_k W^{-1}d)\|^2 < \infty.$$

From the last line, we can deduce that:

- i) the eigenvalues of R are smaller than 1 in magnitude and 1 is the only one with this magnitude;
- ii) the algebraic and geometric multiplicities of eigenvalue 1 coincide as the Jor-

dan form of R does not have block of the form $J_1 = \begin{bmatrix} 1 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 1 \end{bmatrix}$. Indeed,

if it had one could take x^0 so that the terms in the sum are bounded away from zero (e.g. take $J = [1 \ 1; 0 \ 1]$, then $J_k(J - I) = [0 \ 1; 0 \ 0]$).

Thus, one can write $R = \begin{bmatrix} W_1 & W_2 \end{bmatrix} \begin{bmatrix} I & \\ & \tilde{\Lambda} \end{bmatrix} \begin{bmatrix} \frac{W_1^*}{W_2^*} \end{bmatrix}$ where:

- $\tilde{\Lambda}$ is the block diagonal matrix of the Jordan blocks corresponding to the eigenvalues of R with magnitude strictly smaller than 1;
- and $\begin{bmatrix} \frac{W_1^*}{W_2^*} \end{bmatrix} \begin{bmatrix} W_1 & W_2 \end{bmatrix} = \begin{bmatrix} \frac{W_1^* W_1}{W_2^* W_1} & \frac{W_1^* W_2}{W_2^* W_2} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$.

From the previous result, $R = W_1 \underline{W_1^*} + W_2 \tilde{\Lambda} \underline{W_2^*}$ where conveniently $\Pi_{\mathcal{N}} \triangleq W_1 \underline{W_1^*}$ defines a projection onto \mathcal{N} .

Define $\overline{\Pi_{\mathcal{N}}} \triangleq I - \Pi_{\mathcal{N}}$ the complementary projection. Let $\bar{x} \in \text{fix}\mathsf{T}$ and define $\Delta_k \triangleq \overline{\Pi_{\mathcal{N}}}(x_k - \bar{x})$ for all $k > 0$. We have

$$\begin{aligned} \Delta_{k+1} &\triangleq \overline{\Pi_{\mathcal{N}}}(x_{k+1} - \bar{x}) = \overline{\Pi_{\mathcal{N}}}R(x_k - \bar{x}) \\ (7) \quad &= W_2 \tilde{\Lambda} \underline{W_2^*} \underbrace{\overline{\Pi_{\mathcal{N}}}(x_k - \bar{x})}_{\Delta_k} = W_2 \tilde{\Lambda}^k \underline{W_2^*} \Delta^0 \end{aligned}$$

Thus $(\Delta_k)_{k>0}$ vanishes exponentially as a consequence of [16, Cor. 5.6.14] on Eq. (7) which states that there is constant $C \in \mathbb{R}^+$ such that $\|\Delta_k\| \leq Ck^n \rho(W_2 \tilde{\Lambda} \underline{W_2^*})^k$ where the factor k^n stems from the k -th power of the Jordan decomposition of R which introduces terms of the form $\nu^k k^\ell$. Using the ∞ norm and taking the log gives the stated result (see [19, Sec. III-C] or [16, Chap 3.2.5]). Recalling that $\tilde{\Lambda}$ contains the Jordan blocks associated to the non-unit eigenvalues of R , which are all strictly smaller than 1 in magnitude, $\rho(W_2 \tilde{\Lambda} \underline{W_2^*}) = \nu < 1$.

Finally, we can notice that $\Pi_{\mathcal{N}}(x_{k+1} - x_k) = 0$, and thus

$$v_k = \frac{\|x_{k+1} - x_k\|}{\|x_k - x_{k-1}\|} = \frac{\|\overline{\Pi_{\mathcal{N}}}(x_{k+1} - x_k)\|}{\|\overline{\Pi_{\mathcal{N}}}(x_k - x_{k-1})\|} = \frac{\|W_2 \tilde{\Lambda} \underline{W_2^*} \overline{\Pi_{\mathcal{N}}}(x_k - x_{k-1})\|}{\|\overline{\Pi_{\mathcal{N}}}(x_k - x_{k-1})\|} \leq \|\tilde{\Lambda}\|$$

where the inequality tends to be sharper as k grows and $\nu \leq \|\tilde{\Lambda}\| \leq 1$.

Appendix B. Proof of Lemma 6.

Let $\bar{x} \in \text{fixT}$, and take k even, then $x_{k+2} = \mathsf{T}(\mathsf{T}(x_k) + \gamma_{k+1}(\mathsf{T}(x_k) - x_k))$.

$$\begin{aligned}
\|x_{k+2} - \bar{x}\|^2 &= \|\mathsf{T}(\mathsf{T}(x_k) + \gamma_{k+1}(\mathsf{T}(x_k) - x_k)) - \mathsf{T}(\bar{x})\|^2 \\
&\leq \|\mathsf{T}(x_k) + \gamma_{k+1}(\mathsf{T}(x_k) - x_k) - \bar{x}\|^2 - \frac{1-\alpha}{\alpha} \|\mathsf{T}(x_k) + \gamma_{k+1}(\mathsf{T}(x_k) - x_k) - x_{k+2}\|^2 \\
&= (1 + \gamma_{k+1}) \|\mathsf{T}(x_k) - \bar{x}\|^2 - \gamma_{k+1} \|x_k - \bar{x}\|^2 + (1 + \gamma_{k+1})\gamma_{k+1} \|\mathsf{T}(x_k) - x_k\|^2 \\
&\quad - \frac{1-\alpha}{\alpha} \|\mathsf{T}(x_k) + \gamma_{k+1}(\mathsf{T}(x_k) - x_k) - x_{k+2}\|^2 \\
&\leq (1 + \gamma_{k+1}) \|x_k - \bar{x}\|^2 - \gamma_{k+1} \|x_k - \bar{x}\|^2 - (1 + \gamma_{k+1}) \frac{1-\alpha}{\alpha} \|\mathsf{T}(x_k) - x_k\|^2 \\
&\quad + (1 + \gamma_{k+1})\gamma_{k+1} \|\mathsf{T}(x_k) - x_k\|^2 - \frac{1-\alpha}{\alpha} \|\mathsf{T}(x_k) + \gamma_{k+1}(\mathsf{T}(x_k) - x_k) - x_{k+2}\|^2 \\
&= \|x_k - \bar{x}\|^2 - (1 + \gamma_{k+1}) \left(\frac{1-\alpha}{\alpha} - \gamma_{k+1} \right) \|\mathsf{T}(x_k) - x_k\|^2 \\
&\quad - \frac{1-\alpha}{\alpha} \|\mathsf{T}(x_k) + \gamma_{k+1}(\mathsf{T}(x_k) - x_k) - x_{k+2}\|^2
\end{aligned}$$

where we used successively: i) the fact that T is α -averaged; ii) the equality of [5, Cor. 2.14]; iii) a second time that T is α -averaged. The assumption on the sequence (γ_k) makes the second term negative or null hence it can be dropped.

We notice that $\|x_{k+2} - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2 - \frac{1-\alpha}{\alpha} \|\mathsf{T}(x_k) + \gamma_{k+1}(\mathsf{T}(x_k) - x_k) - x_{k+2}\|^2$ implies that the sequence of the *even* $(\|x_{2k} - \bar{x}\|^2)_{k \geq 0}$ is decreasing and non-negative, it is thus convergent and the $(x_{2k})_{k \geq 0}$ are bounded. Furthermore,

$$\sum_{k=0}^{\infty} \|\mathsf{T}(x_{2k}) + \gamma_{2k+1}(\mathsf{T}(x_{2k}) - x_{2k}) - x_{2(k+1)}\|^2 < \infty$$

implies that any limit point of the sequence $(x_{2k})_{k \geq 0}$ belongs to fixT .

Let us now take x^* , a limit point of $(x_{2k})_{k \geq 0}$, then $(\|x_{2k} - x^*\|^2)_{k \geq 0}$ converges and its limit is $\lim_{k \rightarrow \infty} \|x_{2k} - x^*\|^2 = 0$ which means that x^* is unique. Finally, using non-expansivity, we get that the *odd* sequence also converges to the same point x^* .

Appendix C. Derivation of Inertial ADMM (Sec. ref:admmacc).

The derivations follow nearly the same steps as the one of relaxed ADMM in [11] thus we will abridge the common parts. We build upon the ADMM-generating Lions-Mercier operator:

$$\mathsf{T}_{admm} = \{(\lambda + \rho v, w + \rho v) : (u, -M^T w) \in \partial f; (v, \lambda) \in \partial g; w - \rho M u = \lambda - \rho v\}.$$

but we will consider an *inertial version* of the proximal point algorithm⁸:

$$\begin{cases} x_{k+1} = \mathsf{T}_{admm}(y_k) \\ y_{k+1} = x_{k+1} + \gamma(x_{k+1} - x_k) \end{cases}$$

Representation step: The input, y_k , writes uniquely as $\lambda_k + \rho v_k$ from the representation lemma:

$$(8) \quad y_k = \lambda_k + \rho v_k.$$

⁸we chose to perform the operator *then* the inertia for the sake of clarity and consistency in the derivations.

Mapping step: The definition of T_{admm} implies that $\lambda_k - \rho v_k$ writes uniquely as $w - \rho Mu$ with $(u, -M^T w) \in \partial f$:

$$(9) \quad w_{k+1} - \rho Mu_{k+1} = \lambda_k - \rho v_k.$$

Secondly, the output of the resolvent is:

$$(10) \quad x_{k+1} = w_{k+1} + \rho v_k = \lambda_k + \rho Mu_{k+1}.$$

Re-representation step: Here, the proof is a bit different in the *inertial case* as one has to find the values of λ_{k+1} and v_{k+1} with $(v_{k+1}, \lambda_{k+1}) \in \partial g$, so that $y_{k+1} = x_{k+1} + \gamma(x_{k+1} - x_k)$ writes uniquely as:

$$(11) \quad y_{k+1} = \lambda_{k+1} + \rho v_{k+1}.$$

Writing Eq. (9) of the mapping step, leads to the same step *as for classical ADMM*:

$$\begin{aligned} w_{k+1} - \rho Mu_{k+1} &= \lambda_k - \rho v_k \quad \text{with} \quad (u_{k+1}, -M^T w_{k+1}) \in \partial f \\ \Rightarrow u_{k+1} &= \underset{u}{\operatorname{argmin}} \left\{ f(u) + \frac{\rho}{2} \left\| Mu - v_k + \frac{\lambda_k}{\rho} \right\|^2 \right\}. \end{aligned}$$

Now, combining Eqs. (10) and (11), we have (*different from classical ADMM*)

$$\begin{aligned} \lambda_{k+1} + \rho v_{k+1} &= \lambda_k + \rho Mu_{k+1} + \gamma(\lambda_k + \rho Mu_{k+1} - (\lambda_{k-1} + \rho Mu_k)) \quad \text{with} \quad (v_{k+1}, \lambda_{k+1}) \in \partial g \\ \Rightarrow \lambda_k + \rho Mu_{k+1} + \gamma(\lambda_k + \rho Mu_{k+1} - (\lambda_{k-1} + \rho Mu_k)) - \rho v_{k+1} &= \lambda_{k+1} \in \partial g(v_{k+1}) \\ \Rightarrow 0 \in \partial g(v_{k+1}) - \rho \left(Mu_{k+1} - v_{k+1} + \frac{\lambda_k}{\rho} + \gamma \left(Mu_{k+1} + \frac{\lambda_k}{\rho} - Mu_k - \frac{\lambda_{k-1}}{\rho} \right) \right) \\ \Rightarrow v_{k+1} &= \underset{v}{\operatorname{argmin}} \left\{ g(v) + \frac{\rho}{2} \left\| Mu_{k+1} - v + \frac{\lambda_k}{\rho} + \gamma \left(Mu_{k+1} + \frac{\lambda_k}{\rho} - Mu_k - \frac{\lambda_{k-1}}{\rho} \right) \right\|^2 \right\}. \end{aligned}$$

and the first line also tells us that

$$\lambda_{k+1} = \lambda_k + \rho (Mu_{k+1} - v_{k+1}) + \gamma (\lambda_k + \rho Mu_{k+1} - (\lambda_{k-1} + \rho Mu_k))$$

which we can identify as the iterations of *Inertial ADMM*.