Acceleration of fixed point algorithms via inertia

Juan José Maulén
Advisor: Juan Peypouquet

Dutch Optimization Seminar

October 28, 2021

rijksuniversiteit groningen

Table of contents

1. Introduction: Relaxed Inertial Proximal Point Algorithm
2. Convergence of inertial algorithms
3. Framework: Primal-Dual Splitting Algorithm
4. Numerical Experiments
5. Final comments

Table of contents

1. Introduction: Relaxed Inertial Proximal Point Algorithm
2. Convergence of inertial algorithms
3. Framework: Primal-Dual Splitting Algorithm
4. Numerical Experiments
5. Final comments

Introduction: Relaxed Inertial Proximal Point Algorithm

Let H be a Real Hilbert space and $A : H \rightrightarrows H$ a maximally monotone operator.

Introduction: Relaxed Inertial Proximal Point Algorithm

Let H be a Real Hilbert space and $A : H \rightrightarrows H$ a maximally monotone operator.

Problem

Find $x \in H$ such that $0 \in Ax$.

Introduction: Relaxed Inertial Proximal Point Algorithm

Let H be a Real Hilbert space and $A : H \rightrightarrows H$ a maximally monotone operator.

Problem

$$\text{Find } x \in H \text{ such that } 0 \in Ax.$$

Let $\lambda > 0$, $J_{\lambda A} = (I + \lambda A)^{-1}$.

$$0 \in Ax \Longleftrightarrow x \in \operatorname{Fix} J_{\lambda A} x$$

Introduction: Relaxed Inertial Proximal Point Algorithm

Let H be a Real Hilbert space and $A : H \rightrightarrows H$ a maximally monotone operator.

Problem

$$\text{Find } x \in H \text{ such that } 0 \in Ax.$$

Let $\lambda > 0$, $J_{\lambda A} = (I + \lambda A)^{-1}$.

$$0 \in Ax \Longleftrightarrow x \in \operatorname{Fix} J_{\lambda A} x$$

Relaxed Inertial Proximal Point Algorithm

The sequence generated by the algorithm

$$y_k = x_k + \alpha_k(x_k - x_{k-1})$$
$$x_{k+1} = (1 - \rho_k)y_k + \rho_k J_{\lambda_k A}(y_k)$$

converges weakly to a point in $\mathrm{Zer}\, A$.

Existing results for Inertial Algorithms

Objective

Our aim is to study the convergence of algorithms defined by the scheme

$$y_n = x_n + \alpha_n(x_n - x_{n-1})$$
$$x_{n+1} = (1 - \lambda_n)y_n + \lambda_n T y_n,$$

where T is an averaged operator.

Existing results for Inertial Algorithms

Objective

Our aim is to study the convergence of algorithms defined by the scheme

$$\begin{aligned} y_n &= x_n + \alpha_n(x_n - x_{n-1}) \\ x_{n+1} &= (1-\lambda_n)y_n + \lambda_n T y_n, \end{aligned}$$

where T is an averaged operator.

Attouch & Cabot on 2019 proved the convergence for a general scheme

$$\begin{aligned} y_n &= x_n + \alpha_n(x_n - x_{n-1}) \\ x_{n+1} &= y_n - M_n(y_n), \end{aligned}$$

where M_n is a sequence of β_n-cocoercive operators.

Table of contents

1 Introduction: Relaxed Inertial Proximal Point Algorithm

2 Convergence of inertial algorithms

3 Framework: Primal-Dual Splitting Algorithm

4 Numerical Experiments

5 Final comments

Convergence of inertial algorithms

Theorem (Fierro-M-Peypouquet 2021)

Let $T : H \mapsto H$ be an α-averaged operator with $\operatorname{Fix} T \neq \emptyset$, and x_0, $x_1 \in H$. Suppose that (α_n) and (λ_n) are two sequences satisfying the hypotheses presented later, then, the sequence generated by

$$
\begin{aligned}
y_n &= x_n + \alpha_n(x_n - x_{n-1}) \\
x_{n+1} &= (1 - \lambda_n)y_n + \lambda_n T y_n
\end{aligned}
$$

converges weakly to a point in $\operatorname{Fix} T$.

Convergence of inertial algorithms

Theorem (Fierro-M-Peypouquet 2021)

Let $T : H \mapsto H$ be an α-averaged operator with $\operatorname{Fix} T \neq \emptyset$, and x_0, $x_1 \in H$. Suppose that (α_n) and (λ_n) are two sequences satisfying the hypotheses presented later, then, the sequence generated by

$$y_n = x_n + \alpha_n(x_n - x_{n-1})$$
$$x_{n+1} = (1 - \lambda_n)y_n + \lambda_n T y_n$$

converges weakly to a point in $\operatorname{Fix} T$.

Proof idea: The scheme can be rewritten as

$$x_{n+1} = y_n - \lambda_n(I - T)y_n.$$

$M_n = \lambda_n(I - T)$ is β_n-cocoercive, with $\beta_n = 1/(2\alpha\lambda_n)$.

Hypotheses over α_n, λ_n

The sequences must satisfy $\alpha_n \in [0, 1)$, $\lambda_n \in [0, 1/\alpha)$

Hypotheses over α_n, λ_n

The sequences must satisfy $\alpha_n \in [0,1)$, $\lambda_n \in [0, 1/\alpha)$ and there must exist $c \in [0,1)$ and $c' \in [c, 1)$ such that

- $\lim_{n\to+\infty}\left(\frac{1}{1-\alpha_{n+1}} - \frac{1}{1-\alpha_n}\right) = c$

Hypotheses over α_n, λ_n

The sequences must satisfy $\alpha_n \in [0,1)$, $\lambda_n \in [0, 1/\alpha)$ and there must exist $c \in [0,1)$ and $c' \in [c,1)$ such that

- $\displaystyle\lim_{n\to+\infty} \left(\frac{1}{1-\alpha_{n+1}} - \frac{1}{1-\alpha_n} \right) = c$
- $\displaystyle\lim_{n\to+\infty} \frac{(\lambda_n - \lambda_{n+1})}{\lambda_{n+1}(1-\alpha\lambda_n)(1-\alpha_n)} = c'$

Hypotheses over α_n, λ_n

The sequences must satisfy $\alpha_n \in [0,1)$, $\lambda_n \in [0, 1/\alpha)$ and there must exist $c \in [0,1)$ and $c' \in [c,1)$ such that

- $\displaystyle \lim_{n\to+\infty} \left(\frac{1}{1-\alpha_{n+1}} - \frac{1}{1-\alpha_n} \right) = c$
- $\displaystyle \lim_{n\to+\infty} \frac{(\lambda_n - \lambda_{n+1})}{\lambda_{n+1}(1-\alpha\lambda_n)(1-\alpha_n)} = c'$
- $\displaystyle \liminf_{n\to+\infty} \left(\frac{1}{\alpha\lambda_n} - 1 \right)(1-\alpha_n)^2 > \limsup_{n\to+\infty} \frac{\alpha_n(1+\alpha_n)}{1-c'}$

Hypotheses over α_n, λ_n: Example

Let us consider $\lambda_n \equiv \lambda \in [0, 1/\alpha)$ and the sequence

$$\alpha_n = a - \frac{a}{n^q},$$

with $a \in (0,1)$, $q > 0$.

Hypotheses over α_n, λ_n: Example

Let us consider $\lambda_n \equiv \lambda \in [0, 1/\alpha)$ and the sequence

$$\alpha_n = a - \frac{a}{n^q},$$

with $a \in (0,1)$, $q > 0$. Clearly, $\alpha_n \in [0,1]$ and the two first hypotheses are satisfied with $c = c' = 0$.

Hypotheses over α_n, λ_n: Example

Let us consider $\lambda_n \equiv \lambda \in [0, 1/\alpha)$ and the sequence

$$\alpha_n = a - \frac{a}{n^q},$$

with $a \in (0,1)$, $q > 0$. Clearly, $\alpha_n \in [0,1]$ and the two first hypotheses are satisfied with $c = c' = 0$. The third condition implies that

$$\left(\frac{1}{\alpha\lambda} - 1\right) > \frac{a(1+a)}{(1-a)^2} \tag{1}$$

Table of contents

1. Introduction: Relaxed Inertial Proximal Point Algorithm
2. Convergence of inertial algorithms
3. Framework: Primal-Dual Splitting Algorithm
4. Numerical Experiments
5. Final comments

Application of the Primal-Dual Splitting Algorithm

Let H and G two real Hilbert spaces, $A : H \to 2^H$, $B : G \to 2^G$ maximally monotone operators and $L : H \to G$ a bounded linear operator.

Application of the Primal-Dual Splitting Algorithm

Let H and G two real Hilbert spaces, $A : H \to 2^H$, $B : G \to 2^G$ maximally monotone operators and $L : H \to G$ a bounded linear operator.

Primal-Dual Splitting Algorithm (Briceño-Roldan, 2019)

Briceño-Roldan proposed a fixed point algorithm, such that the iterations converges to a point (x, u) solution of the inclusion

$$0 \in Ax + L^*u, \quad 0 \in B^{-1}u - Lx$$

Application of the Primal-Dual Splitting Algorithm

Let H and G two real Hilbert spaces, $A : H \to 2^H$, $B : G \to 2^G$ maximally monotone operators and $L : H \to G$ a bounded linear operator.

Primal-Dual Splitting Algorithm (Briceño-Roldan, 2019)

Briceño-Roldan proposed a fixed point algorithm, such that the iterations converges to a point (x, u) solution of the inclusion

$$0 \in Ax + L^*u, \quad 0 \in B^{-1}u - Lx$$

Consider the optimization problem

$$\min_{x \in H} f(x) + g(Lx),$$

for the lower semicontinuous, convex and proper functions f and g.

Application of the Primal-Dual Splitting Algorithm

Let H and G two real Hilbert spaces, $A : H \to 2^H$, $B : G \to 2^G$ maximally monotone operators and $L : H \to G$ a bounded linear operator.

Primal-Dual Splitting Algorithm (Briceño-Roldan, 2019)

Briceño-Roldan proposed a fixed point algorithm, such that the iterations converges to a point (x, u) solution of the inclusion

$$0 \in Ax + L^*u, \quad 0 \in B^{-1}u - Lx$$

Consider the optimization problem

$$\min_{x \in H} f(x) + g(Lx),$$

for the lower semicontinuous, convex and proper functions f and g. Fenchel-Rockafellar duality conditions implies that we need to solve the inclusions

$$-L^*u \in \partial f(x),\ u \in \partial g^*(Lx),$$

for $x \in H$, $u \in G$.

Application of the Primal-Dual Splitting Algorithm

Algorithm 1:

Choose $x_0, x_1 \in \mathbb{R}^N$, $u_0, u_1 \in \mathbb{R}^n$, τ, $\sigma > 0$, $\varepsilon > 0$ and $r_0 > \varepsilon$;

while $r_n > \varepsilon$ **do**

$(y_n, v_n) = (x_n, u_n) + \alpha_n[(x_n, u_n) - (x_{n-1}, u_{n-1})]$;

$x_{n+1} = \mathrm{prox}_{\tau f}(y_n - \tau L^* v_n)$;

$u_{n+1} = \mathrm{prox}_{\sigma g^*}(u_n + \sigma L(2x_{n+1} - x_n)$;

$r_{n+1} = R((x_{n+1}, u_{n+1}), (x_n, u_n))$;

end

Return (x_{n+1}, u_{n+1})

Application of the Primal-Dual Splitting Algorithm

Algorithm 2:

Choose $x_0, x_1 \in \mathbb{R}^N$, $u_0, u_1 \in \mathbb{R}^n$, τ, $\sigma > 0$, $\varepsilon > 0$ and $r_0 > \varepsilon$;

while $r_n > \varepsilon$ **do**

$(y_n, v_n) = (x_n, u_n) + \alpha_n[(x_n, u_n) - (x_{n-1}, u_{n-1})]$;

$x_{n+1} = \mathrm{prox}_{\tau f}(y_n - \tau L^* v_n)$;

$u_{n+1} = \mathrm{prox}_{\sigma g^*}(u_n + \sigma L(2x_{n+1} - x_n)$;

$r_{n+1} = R((x_{n+1}, u_{n+1}), (x_n, u_n))$;

end

Return (x_{n+1}, u_{n+1})

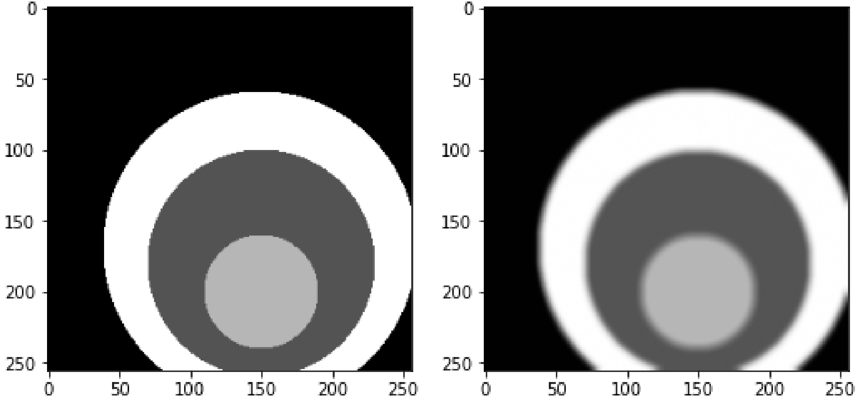
The scheme proposed by Briceño-Roldan converges if $\tau\sigma\|L\|^2 \leq 1$.

Table of contents

1. Introduction: Relaxed Inertial Proximal Point Algorithm
2. Convergence of inertial algorithms
3. Framework: Primal-Dual Splitting Algorithm
4. Numerical Experiments
5. Final comments

Image processing

The numerical problem aims to recover a noisy image.



(a) Original Image $\bar{x}$ (b) Blurred Image $b = R\bar{x} + e$

Image processing: Total Variation

Let us consider $\nabla : x \mapsto \nabla x = (D_1 x, D_2 x)$, the classical discrete gradient. The model can be formulated via the optimization problem

$$\min_{x \in \mathbb{R}^{N_1 \times N_2}} F^{TV}(x) := \frac{1}{2}\|Rx - b\|^2 + \gamma \|\nabla x\|_1$$

Image processing: Total Variation

Let us consider $\nabla : x \mapsto \nabla x = (D_1 x, D_2 x)$, the classical discrete gradient. The model can be formulated via the optimization problem

$$\min_{x \in \mathbb{R}^{N_1 \times N_2}} F^{TV}(x) := \frac{1}{2}\|Rx - b\|^2 + \gamma \|\nabla x\|_1$$

Setting $f = 0$, $g : (u, v^1, v^2) \mapsto \frac{1}{2}\|u - b\|^2 + \gamma \|v^1\|_1 + \gamma \|v^2\|_1$, $L : x \mapsto (Rx, D_1 x, D_2 x)$, the problem can be solved via the inertial Algorithm scheme showed before.

Image processing: Total Variation

Let us consider $\nabla : x \mapsto \nabla x = (D_1 x, D_2 x)$, the classical discrete gradient. The model can be formulated via the optimization problem

$$\min_{x \in \mathbb{R}^{N_1 \times N_2}} F^{TV}(x) := \frac{1}{2}\|Rx - b\|^2 + \gamma \|\nabla x\|_1$$

Setting $f = 0$, $g : (u, v^1, v^2) \mapsto \frac{1}{2}\|u - b\|^2 + \gamma\|v^1\|_1 + \gamma\|v^2\|_1$, $L : x \mapsto (Rx, D_1 x, D_2 x)$, the problem can be solved via the inertial Algorithm scheme showed before. The inertial algorithm is tested with $\lambda = 1$,

$$\alpha_n = \frac{1}{3+\delta} - \frac{1}{(3+\delta)n^2},$$

and 19 cases for τ and σ.

Image processing: Total Variation

		Original Algorithm			Inertial algorithm, $q=2$		
τ	σ	Time	Iterations	$F^{TV}(x)$	Time	Iterations	$F^{TV}(x)$
10	0,0125	61,08	1.238	0,1301	44,44	833	0,1301
23,06	0,0054	49,11	985	0,1302	36,77	694	0,1302
53,18	0,0024	71,53	1.449	0,1302	55,12	1.052	0,1302

Table: Original algorithm vs. inertial version comparison, $\varepsilon = 10^{-5}$.

Image processing: Total Variation

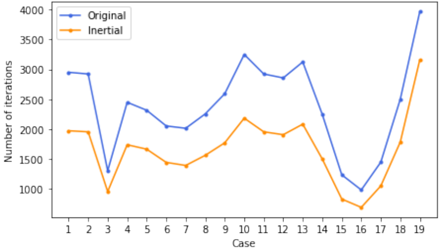
		Original Algorithm			Inertial algorithm, $q = 2$		
τ	σ	Time	Iterations	$F^{TV}(x)$	Time	Iterations	$F^{TV}(x)$
10	0,0125	61,08	1.238	0,1301	44,44	833	0,1301
23,06	0,0054	49,11	985	0,1302	36,77	694	0,1302
53,18	0,0024	71,53	1.449	0,1302	55,12	1.052	0,1302

Table: Original algorithm vs. inertial version comparison, $\varepsilon = 10^{-5}$.

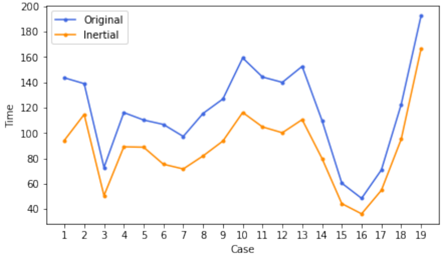
		Original Algorithm			Inertial algorithm, $q = 2$		
τ	σ	Time	Iterations	$F^{TV}(x)$	Time	Iterations	$F^{TV}(x)$
10	0,0125	89,52	1.789	0,1301	65,19	1.223	0,1301
23,06	0,0054	91,86	1.823	0,1301	70,20	1.333	0,1301
53,18	0,0024	161,33	3.230	0,1301	124,95	2.368	0,1301

Table: Original algorithm vs. inertial version comparison, $\varepsilon = 10^{-6}$.

Image processing: Total Variation



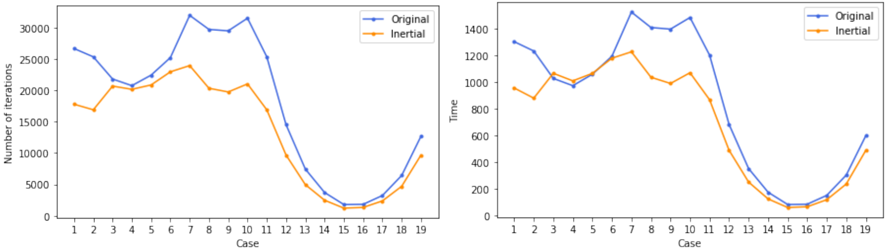
(c) Number of iterations.



(d) Time.

Figure: Comparison for the original and inertial algorithm using $\varepsilon = 10^{-5}$.

Image processing: Total Variation

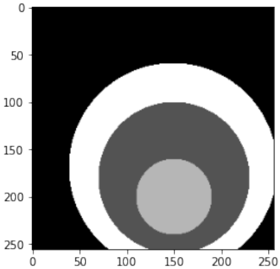


(a) Number of iterations.

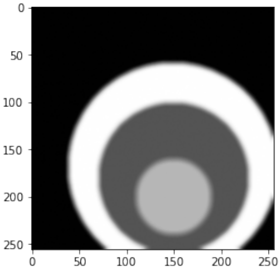
(b) Time.

Figure: Comparison for the original and inertial algorithm using $\varepsilon = 10^{-6}$.

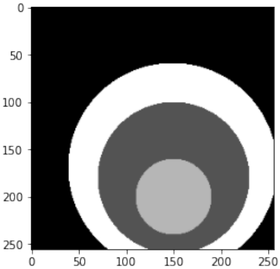
Image processing: Total Variation



(a) Original Image



(b) Blurred Image



(c) Recovered Image

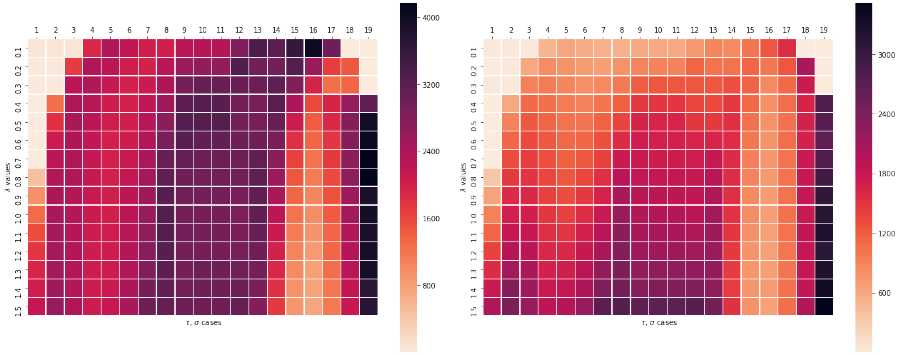
Figure: Recovered Image. $\tau = 23,06$, $\sigma = 0,0054$, $\varepsilon = 10^{-5}$.

Image processing: Total Variation

The performance of the algorithm is tested for several values of λ. For each one, a value of a in the sequence

$$\alpha_n = a - \frac{a}{n^2}$$

is proposed in order to satisfy the previous conditions.



(a) Original Algorithm (b) Inertial Algorithm

Figure: Mean amount of iterations performed by the original and inertial algorithm to reach the tolerance, for each value of λ, and each case of τ and σ, $\varepsilon = 10^{-5}$, using 5 starting points.

Table of contents

1. Introduction: Relaxed Inertial Proximal Point Algorithm
2. Convergence of inertial algorithms
3. Framework: Primal-Dual Splitting Algorithm
4. Numerical Experiments
5. Final comments

Final comments

- The inertial algorithm is tested in two other numerical simulations on the primal-dual splitting framework.

Final comments

- The inertial algorithm is tested in two other numerical simulations on the primal-dual splitting framework.
- An inertial scheme is proposed for the three-operator splitting scheme (Davis-Yin 2017): find $x \in H$ such that

$$0 \in Ax + Bx + Cx.$$

Final comments

- The inertial algorithm is tested in two other numerical simulations on the primal-dual splitting framework.
- An inertial scheme is proposed for the three-operator splitting scheme (Davis-Yin 2017): find $x \in H$ such that

$$0 \in Ax + Bx + Cx.$$

- Convergence rate for the inertial algorithms.

References

- Attouch, H., & Cabot, A. (2019). *Convergence of a relaxed inertial forward–backward algorithm for structured monotone inclusions*. Applied Mathematics & Optimization, 80(3), 547-598.
- Briceño-Arias, L., & Roldán, F. (2019). *Primal-dual splittings as fixed point iterations in the range of linear operators*. arXiv preprint arXiv:1910.02329.
- Davis, D., & Yin, W. (2017). *A three-operator splitting scheme and its optimization applications*. Set-valued and variational analysis, 25(4), 829-858