Anderson acceleration of coordinate descent

Quentin Bertrand
Université Paris-Saclay
Inria, CEA
Palaiseau, France

Mathurin Massias
MaLGa
DIBRIS
University of Genova

Abstract

Acceleration of first order methods is mainly obtained via inertia à la Nesterov, or via non-linear extrapolation. The latter has known a recent surge of interest, with successful applications to gradient and proximal gradient techniques. On multiple Machine Learning problems, coordinate descent achieves performance significantly superior to full-gradient methods. Speeding up coordinate descent in practice is not easy: inertially accelerated versions of coordinate descent are theoretically accelerated, but might not always lead to practical speed-ups. We propose an accelerated version of coordinate descent using extrapolation, showing considerable speed up in practice, compared to inertial accelerated coordinate descent and extrapolated (proximal) gradient descent. Experiments on least squares, Lasso, elastic net and logistic regression validate the approach.

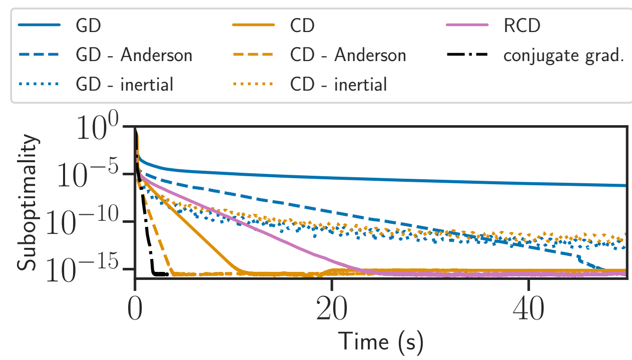

Figure 1: Suboptimality along time for a quadratic problem on the 5000 first features of the *rcv1* dataset. GD: gradient descent, (R)CD: (randomized) coordinate descent.

1 Introduction

Gradient descent is the workhorse of modern convex optimization (Nesterov, 2004; Beck, 2017). For composite problems, proximal gradient descent retains the nice properties enjoyed by the latter. In both techniques, inertial acceleration achieves accelerated convergence rates (Nesterov, 1983; Beck and Teboulle, 2009).

Coordinate descent is a variant of gradient descent, which updates the iterates one coordinate at a time (Tseng and Yun, 2009; Friedman et al., 2010). Proximal coordinate descent has been applied to numerous Machine Learning problems (Shalev-Shwartz and Zhang, 2013; Wright, 2015; Shi et al., 2016), in particular the Lasso (Tibshirani, 1996), elastic net (Zou and Hastie, 2005) or sparse logistic regression (Ng, 2004). It is used in preeminent packages such as scikit-learn (Pedregosa et al., 2011), glmnet (Friedman et al., 2009), libsvm (Fan et al., 2008) or lightning (Blondel and Pedregosa, 2016). On the theoretical side, inertial accelerated versions of coordinate descent (Nesterov, 2012; Lin et al., 2014; Fercoq and Richtárik, 2015) achieve accelerated rates. Note that usual lower bounds (Nesterov, 2004, Sec. 2.1.2) are derived for methods with iterates lying in the span of previous gradients, which is not the case for coordinate descent. However there also exists similar lower bounds for cyclic coordinate descent (Sun and Ye, 2019).

To obtain accelerated rates, Anderson extrapolation (Anderson, 1965) is an alternative to inertia: it provides acceleration by exploiting the iterates' structure. This procedure has been known for a long time, under various names and variants (Wynn, 1962; Eddy, 1979; Smith et al., 1987), see Sidi (2017); Brezinski et al. (2018) for reviews. Anderson acceleration enjoys accelerated rates on quadratics (Golub and Varga,

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s). Contact at quentin.bertrand@inria.fr.

Algorithm 1 Offline Anderson extrapolation

init: $x^{(0)} \in \mathbb{R}^p$

for $k = 1, \dots$ **do**

  $x^{(k)} = Tx^{(k-1)} + b$ // regular linear iteration

  $U = [x^{(1)} - x^{(0)}, \dots, x^{(k)} - x^{(k-1)}]$

  $c = (U^\top U)^{-1} \mathbf{1}_k / \mathbf{1}_k^\top (U^\top U)^{-1} \mathbf{1}_k \in \mathbb{R}^k$

  $x^{(k)}_{\text{e-off}} = \sum_{i=1}^k c_i x^{(i)}$ // does not affect $x^{(k)}$

return $x^{(k)}_{\text{e-off}}$

Algorithm 2 Online Anderson extrapolation

init: $x^{(0)} \in \mathbb{R}^p$

for $k = 1, \dots$ **do**

  $x^{(k)} = Tx^{(k-1)} + b$ // regular iteration

  if $k = 0 \mod K$ **then**

    $U = [x^{(k-K+1)} - x^{(k-K)}, \dots, x^{(k)} - x^{(k-1)}]$

    $c = (U^\top U)^{-1} \mathbf{1}_K / \mathbf{1}_K^\top (U^\top U)^{-1} \mathbf{1}_K \in \mathbb{R}^K$

    $x^{(k)}_{\text{e-on}} = \sum_{i=1}^K c_i x^{(k-K+i)}$

    $x^{(k)} = x^{(k)}_{\text{e-on}}$ // base sequence changes

return $x^{(k)}$

1961), but theoretical guarantees in the nonquadratic case are weaker (Scieur et al., 2016). Interestingly, numerical performances still show significant improvements on nonquadratic objectives. Anderson acceleration has been adapted to various algorithms such as Douglas-Rachford (Fu et al., 2019), ADMM (Poon and Liang, 2019) or proximal gradient descent (Zhang et al., 2018; Mai and Johansson, 2019; Poon and Liang, 2020). Among main benefits, the practical version of Anderson acceleration is memory efficient, easy to implement, line search free, has a low cost per iteration and does not require knowledge of the strong convexity constant. Finally, it introduces a single additional parameter, which often does not require tuning (see Section 3.1).

In this work:

- We propose an Anderson acceleration scheme for coordinate descent, which, as visible on Figure 1, outperforms inertial and extrapolated gradient descent, as well as inertial and randomized coordinate descent.
- The acceleration is obtained even though the iteration matrix is not symmetric, a notable problem in the analysis of Anderson extrapolation.
- We empirically highlight that the proposed acceleration technique can generalize in the nonquadratic case (Algorithm 3) and significantly improve proximal coordinate descent algorithms (Section 3), which are state-of-the-art first order methods on the considered problems.

Notation The j-th line of the matrix A is $A_{j:}$ and its j-th column is $A_{:j}$. The canonical basis vectors of $\mathbb{R}^p$ are e_j. The vector of size K with all one entries is $\mathbf{1}_K$. The spectral radius of the matrix A, $\rho(A)$, is the largest eigenvalue modulus of A. The set of p by p symmetric positive semidefinite matrices is $\mathbb{S}^p_+$. The condition number $\kappa(A)$ of a matrix A is its largest singular value divided by its smallest. A positive definite matrix A induces the norm $\|x\|_A = \sqrt{x^\top A x}$. The proximity operator of the function g is $\text{prox}_g(x) = \arg\min_y g(y) + \frac{1}{2}\|x - y\|^2$.

2 Anderson extrapolation

2.1 Background

Anderson extrapolation is designed to accelerate the convergence of sequences based on fixed point linear iterations, that is:

$$x^{(k+1)} = Tx^{(k)} + b \ , \tag{1}$$

where the *iteration matrix* $T \in \mathbb{R}^{p \times p}$ has spectral radius $\rho(T) < 1$. There exist two variants: offline and online, which we recall briefly.

Offline extrapolation (Algorithm 1), at iteration k, looks for a fixed point as an affine combination of the k first iterates: $x^{(k)}_{\text{e-off}} = \sum_1^k c_i^{(k)} x^{(i-1)}$, and solves for the coefficients $c^{(k)} \in \mathbb{R}^k$ as follows:

$$\begin{aligned} c^{(k)} &= \underset{\sum_1^k c_i = 1}{\arg\min} \|\textstyle\sum_1^k c_i x^{(i-1)} - T \sum_1^k c_i x^{(i-1)} - b\|^2 \\ &= \underset{\sum_1^k c_i = 1}{\arg\min} \|\textstyle\sum_1^k c_i \big(x^{(i)} - x^{(i-1)}\big)\|^2 \\ &= (U^\top U)^{-1} \mathbf{1}_k / \mathbf{1}_k^\top (U^\top U)^{-1} \mathbf{1}_k \ , \end{aligned} \tag{2}$$

where $U = [x^{(1)} - x^{(0)}, \dots, x^{(k)} - x^{(k-1)}] \in \mathbb{R}^{p \times k}$ (and hence the objective rewrites $\|Uc\|^2$). In practice, since $x^{(k)}$ is available when $c^{(k)}$ is computed, one uses $x_{\text{e}}^{(k)} = \sum_1^k c_i^{(k)} x^{(i)}$ instead of $\sum_1^k c_i^{(k)} x^{(i-1)}$. The motivation for introducing the coefficients $c^{(k)}$ is discussed in more depth after Prop. 6 in Massias et al. (2020), and details about the closed-form solution can be found in Scieur et al. (2016, Lem. 2.4). In offline acceleration, more and more base iterates are used to produce the extrapolated point, but the extrapolation sequence does not affect the base sequence. This may not scale well since it requires solving larger and larger linear systems.

A more practical variant is the *online* version (Algorithm 2), considered in this paper. The number of points to be extrapolated is fixed to K; $x^{(1)}, \dots, x^{(K)}$ are computed normally with the fixed point iterations, but $x_{\text{e}}^{(K)}$ is computed by extrapolating the iterates from $x^{(1)}$ to $x^{(K)}$, and $x^{(K)}$ is taken equal to $x_{\mathrm{e}}^{(K)}$. K normal iterates are then computed from $x^{(K+1)}$ to $x^{(2K)}$ then extrapolation is performed on these last K iterates, etc.

As we recall below, results on Anderson acceleration mainly concern fixed-point iterations with symmetric iteration matrices T, and results concerning non-symmetric iteration matrices are weaker (Bollapragada et al., 2018). Poon and Liang (2020, Thm 6.4) do not assume that T is symmetric, but only diagonalizable, which is still a strong requirement.

Proposition 1 (Symmetric T, Scieur 2019)**.** Let the iteration matrix T be symmetric semi-definite positive, with spectral radius $\rho = \rho(T) < 1$. Let x^* be the limit of the sequence $(x^{(k)})$. Let $\zeta = (1 - \sqrt{1-\rho})/(1 + \sqrt{1-\rho})$. Then the iterates of offline Anderson acceleration satisfy, with $B = (\mathrm{Id} - T)^2$:

$$\|x_{\text{e-off}}^{(k)} - x^*\|_B \leq \tfrac{2\zeta^{k-1}}{1+\zeta^{2(k-1)}} \|x^{(0)} - x^*\|_B \ , \tag{3}$$

and thus those of online extrapolation satisfy:

$$\|x_{\text{e-on}}^{(k)} - x^*\|_B \leq \left(\tfrac{2\zeta^{K-1}}{1+\zeta^{2(K-1)}}\right)^{k/K} \|x^{(0)} - x^*\|_B \ . \tag{4}$$

Scieur et al. (2016) showed that the offline version in Proposition 1 matches the accelerated rate of the conjugate gradient (Hestenes and Stiefel, 1952). As it states, gradient descent can be accelerated by Anderson extrapolation on quadratics.

Application to least squares The canonical application of Anderson extrapolation is gradient descent on least squares. Consider a quadratic problem, with $b \in \mathbb{R}^p$, $H \in \mathbb{S}_{++}^p$ such that $0 \prec H \preceq L$ and $L > 0$:

$$x^* = \mathop{\arg\min}_{x \in \mathbb{R}^p} \frac{1}{2} x^\top H x + \langle b, x \rangle \ . \tag{5}$$

A typical instance is overdetermined least squares with full-column rank design matrix $A \in \mathbb{R}^{n \times p}$, and observations $y \in \mathbb{R}^n$, such that $H = A^\top A$ and $b = -A^\top y$. On Problem (5) gradient descent with step size $1/L$ reads:

$$x^{(k+1)} = \underbrace{(\mathrm{Id}_p - \tfrac{1}{L} H)}_{T^{\mathrm{GD}} \in \mathbb{S}_+^p} x^{(k)} + \underbrace{(-b/L)}_{b^{\mathrm{GD}}} \ . \tag{6}$$

Because they have this linear structure, iterates of gradient descent can benefit from Anderson acceleration, observing that the fixed point of $x \mapsto T^{\mathrm{GD}} x + b^{\mathrm{GD}}$ solves (5), with $T^{\mathrm{GD}} \in \mathbb{S}_+^p$. Anderson acceleration of gradient descent has therefore been well-studied beyond the scope of Machine Learning (Pulay, 1980; Eyert, 1996). However, on many Machine Learning problems, coordinate descent achieves far superior performance, and it is interesting to determine whether or not it can also benefit from Anderson extrapolation.

2.2 Linear iterations of coordinate descent

To apply Anderson acceleration to coordinate descent, we need to show that its iterates satisfy linear iterations as in (6). An epoch of cyclic coordinate descent for Problem (5) consists in updating the vector x one coordinate at a time, sequentially, i.e. for $j = 1, \dots, p$:

$$x_j \leftarrow x_j - \frac{1}{H_{jj}} (H_{j:} x + b_j) \ , \tag{7}$$

which can be rewritten, for $j = 1, \dots, p$:

$$x \leftarrow \left(\mathrm{Id}_p - \frac{e_j e_j^\top}{H_{jj}} H\right) x - \frac{b_j}{H_{jj}} e_j \ . \tag{8}$$

Thus, for primal iterates, as observed by Bertrand et al. (2020, Sec. A.3), one full pass (updating coordinates from 1 to p) leads to a linear iteration:

$$x^{(k+1)} = T^{\mathrm{CD}} x^{(k)} + b^{\mathrm{CD}} \ , \tag{9}$$

with $T^{\mathrm{CD}} = \left(\mathrm{Id}_p - \frac{e_p e_p^\top}{H_{pp}} H\right) \dots \left(\mathrm{Id}_p - \frac{e_1 e_1^\top}{H_{11}} H\right)$. Note that in the case of coordinate descent we write $x^{(k)}$ for the iterates after one pass of coordinate descent on all features, and not after each update (7). The iterates of coordinate therefore also have a fixed-point structure, but contrary to gradient descent, their iteration matrix T^{CD} is not symmetric, which we address in Section 2.3.

2.3 Anderson extrapolation for nonsymmetric iteration matrices

Even on quadratics, Anderson acceleration with nonsymmetric iteration matrices is less developed, and the only results concerning its theoretical acceleration are recent and weaker than in the symmetric case.

Proposition 2 (Bollapragada et al. 2018, Thm 2.2)**.** When T is not symmetric, and $\rho(T) < 1$,

$$\|x_{\text{e-off}}^{(k)} - T x_{\text{e-off}}^{(k)} - b\| \leq \\ \|\mathrm{Id} - \rho(T - \mathrm{Id})\|_2 \, \|P^*(T)(x^{(1)} - x^{(0)})\| \ ,$$

where the unavailable polynomial P^* minimizes $\left\|P(T)(x^{(1)} - x^{(0)})\right\|$ amongst all polynomials P of degree exactly $k-1$ whose coefficients sum to 1.

The quality of the bound (in particular, its eventual convergence to 0) crucially depends on $\|P(T)\|$. Using the Crouzeix conjecture (Crouzeix, 2004) Bollapragada et al. (2018) managed to bound $\|P(T)\|$, with P a polynomial:

$$\|P(T)\| \leq c \max_{z \in W(T)} |P(z)| \ , \tag{10}$$

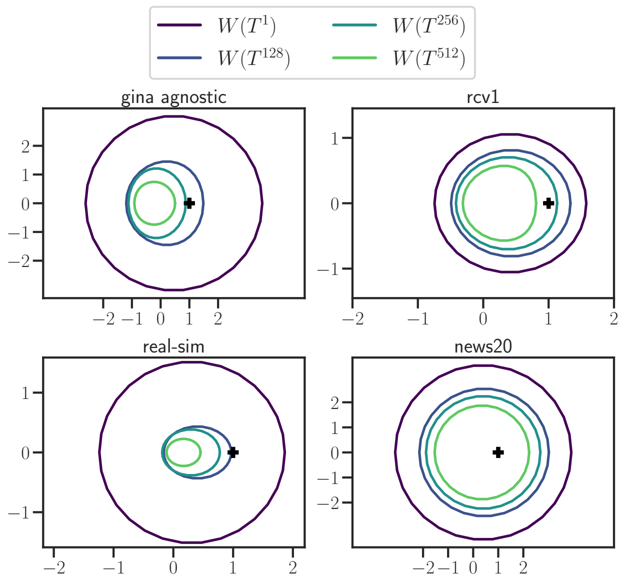

Figure 2: Numerical range of T^q as q varies; T is the iteration matrix of Ridge regression problems with conditioning $\kappa = 10^3$, on 4 datasets. The black cross marks the $(1, 0)$ point, which should lie outside the range for the theoretical bound to be useful.

with $c \geq 2$ (Crouzeix, 2007; Crouzeix and Palencia, 2017), and $W(T)$ the numerical range:

$$W(T) \triangleq \{x^* T x : \|x\|_2 = 1, x \in \mathbb{C}^p\} \ . \tag{11}$$

Since there is no general formula for this bound, Bollapragada et al. (2018) used numerical bounds on $W(T^q)$ to ensure convergence. Figure 2 displays the numerical range $W(T^q)$ in the complex plane for $q \in \{1, 128, 256, 512\}$. In order to be able to apply the theoretical result from Bollapragada et al. (2018), one must chose q such that the point $(1, 0)$ is not contained in $W(T^q)$, and extrapolate $x^{(0)}, x^{(q)}, x^{(2q)}, \dots$ One can see on Figure 2 that large values of q are needed, unusable in practice: $q = 512$ is greater than the number of iterations needed to converge on some problems. Moreover, Anderson acceleration seems to provide speed up on coordinate descent even with $q = 1$ as we perform, which highlights the need for refined bounds for Anderson acceleration on nonsymmetric matrices.

We propose two means to fix this lack of theoretical results: to modify the algorithm in order to have a more amenable iteration matrix (Section 2.4), or to perform a simple cost function decrease check (Section 2.5).

2.4 Pseudo-symmetrization of T

A first idea to make coordinate descent theoretically amenable to extrapolation is to perform updates of coefficients from indices 1 to p, followed by a reversed pass from p to 1. This leads to an iteration matrix which is not symmetric either but friendlier: it writes

$$T^{\text{CD-sym}} \triangleq H^{-1/2} S H^{1/2} \ , \tag{12}$$

with

$$S = \left(\mathrm{Id}_p - H^{1/2} \tfrac{e_1 e_1^\top}{H_{11}} H^{\frac{1}{2}}\right) \times \dots \times \left(\mathrm{Id}_p - H^{\frac{1}{2}} \tfrac{e_p e_p^\top}{H_{pp}} H^{\frac{1}{2}}\right) \times \left(\mathrm{Id}_p - H^{\frac{1}{2}} \tfrac{e_p e_p^\top}{H_{pp}} H^{\frac{1}{2}}\right) \times \dots \times \left(\mathrm{Id}_p - H^{\frac{1}{2}} \tfrac{e_1 e_1^\top}{H_{11}} H^{\frac{1}{2}}\right) \ . \tag{13}$$

S is symmetric, thus, S and T (which has the same eigenvalues as S), are diagonalisable with real eigenvalues. We call these iterations pseudo-symmetric, and show that this structure allows to preserve the guarantees of Anderson extrapolation.

Proposition 3 (Pseudosym. $T = H^{-1/2} S H^{1/2}$)**.** Let T be the iteration matrix of pseudo-symmetric coordinate descent: $T = H^{-1/2} S H^{1/2}$, with S the symmetric positive semidefinite matrix of (12). Let x^* be the limit of the sequence $(x^{(k)})$. Let $\zeta = (1 - \sqrt{1-\rho})/(1 + \sqrt{1-\rho})$. Then $\rho = \rho(T) = \rho(S) < 1$ and the iterates of offline Anderson acceleration satisfy:

$$\|x^{(k)}_{\text{e-off}} - x^*\|_B \leq \sqrt{\kappa(H)} \tfrac{2\zeta^{k-1}}{1+\zeta^{2(k-1)}} \|x^{(0)} - x^*\|_B \ , \tag{14}$$

and thus those of online extrapolation satisfy:

$$\|x^{(k)}_{\text{e-on}} - x^*\|_B \leq \left(\sqrt{\kappa(H)} \tfrac{2\zeta^{K-1}}{1+\zeta^{2(K-1)}}\right)^{k/K} \|x^{(0)} - x^*\|_B \ . \tag{15}$$

Proof of Proposition 3 can be found in Appendix B. Proposition 3 shows accelerated convergence rates for the offline Anderson acceleration, but a $\sqrt{\kappa(H)}$ appears in the rate of the online Anderson acceleration, meaning that K must be large enough that ζ^K mitigates this effect. This factor however seems like a theoretical artefact of the proof, since we observed significant speed up of the online Anderson acceleration, even with bad conditioning of H (see Figure 3).

Figure 3 illustrates the convergence speed of cyclic and pseudo-symmetric coordinate descent on the *rcv1* dataset. Anderson acceleration provides speed up for both versions. Interestingly, on this quadratic problem, the non extrapolated pseudo-symmetric iterations perform poorly, worse than cyclic coordinate descent. However, the performances are reversed for their extrapolated counterparts: the pseudo-symmetrized version is better than the cyclic one (which has a nonsym-

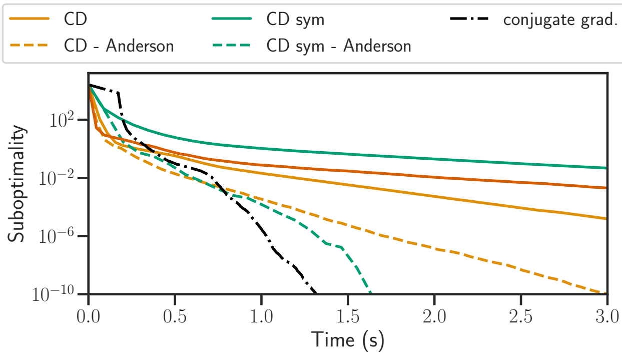

Figure 3: **OLS, *rcv1*.** Suboptimality as a function of time on the 5000 first columns of the dataset *rcv1*.

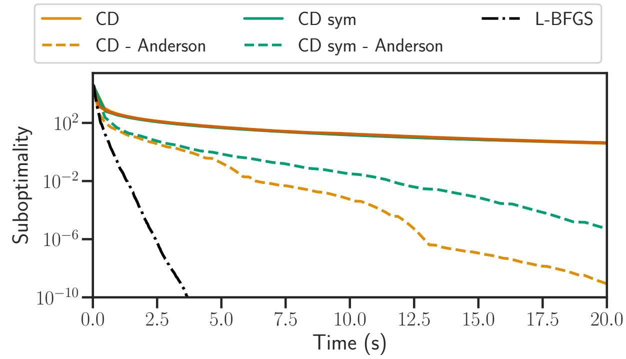

Figure 4: **ℓ_2-regularised logistic regression, *real-sim*.** Suboptimality as a function of time on the 2000 first features of the *real-sim* dataset, Tikhonov strength set so that $\kappa = 10^5$.

Algorithm 3 Online Anderson PCD (proposed)

init: $x^{(0)} \in \mathbb{R}^p$
for $k = 1, \ldots$ **do**
  $x = x^{(k-1)}$
  for $j = 1, \ldots p$ **do**
    $\tilde{x}_j = x_j$
    $x_j = \text{prox}_{\frac{\lambda}{L_j} g_j}(x_j - A_{:j}^\top \nabla f(Ax)/L_j)$
    $Ax \mathrel{+}= (x_j - \tilde{x}_j) A_{:j}$
  $x^{(k)} = x$ // regular iter. $\mathcal{O}(np)$
  if $k = 0 \mod K$ **then** // extrapol., $\mathcal{O}(K^3 + pK^2)$
    $U = [x^{(k-K+1)} - x^{(k-K)}, \ldots, x^{(k)} - x^{(k-1)}]$
    $c = (U^\top U)^{-1} \mathbf{1}_K / \mathbf{1}_K^\top (U^\top U)^{-1} \mathbf{1}_K \in \mathbb{R}^K$
    $x_{\mathrm{e}} = \sum_{i=1}^K c_i x^{(k-K+i)}$
    if $f(Ax_{\mathrm{e}}) + \lambda g(x_{\mathrm{e}}) \leq f(x^{(k)}) + \lambda g(x^{(k)})$ **then**
      $x^{(k)} = x_{\mathrm{e}}$ // guaranteed convergence
return $x^{(k)}$

metric iteration matrix). Finally, Anderson extrapolation on the pseudo-symmetrized version even reaches the conjugate gradient performance.

2.5 Generalization to nonquadratic and proposed algorithm

After devising and illustrating an Anderson extrapolated coordinate descent procedure for a simple quadratic objective, our goal is to apply Anderson acceleration on problems where coordinate descent achieve state-of-the-art results, *i.e.*, of the form:

$$\min_{x \in \mathbb{R}^p} f(Ax) + \lambda g(x) \triangleq f(Ax) + \lambda \sum_{j=1}^p g_j(x_j) \enspace ,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex, γ-smooth and g_j's are proper, closed and convex functions As examples, we allow $g = 0$, $g = \|x\|_1$, $g = \frac{1}{2}\|x\|_2^2$, $g = \|x\|_1 + \frac{\rho}{2\lambda}\|x\|^2$. In the nonquadratic case, for the proximal coordinate descent, following Klopfenstein et al. (2020), the update of the j-st coordinate can be written $\psi_j : \mathbb{R}^p \to \mathbb{R}^p$, $x \mapsto x + (\text{prox}_{\lambda g_j / L_j}(x_j - \gamma_j \nabla_j f(x)) - x_j) e_j$. The nonlinear operator of one pass of coordinate descent (i.e. one pass on all the features) can thus be written: $\psi = \psi_p \circ \cdots \circ \psi_1$. One pass of proximal coordinate descent from 1 to p can be seen as a nonlinear fixed point iteration:

$$x^{(k+1)} = \psi(x^{(k)}) \enspace . \tag{16}$$

In this case, T is not a matrix, but a nonlinear operator. However, as stated in Proposition 4, asymptotically, this operator T is linear.

Proposition 4. If f is convex and smooth and $\mathcal{C}^2$, g_j are convex smooth and $\mathcal{C}^2$, then ψ is differentiable and

$$x^{(k+1)} = D\psi(x^*)(x^{(k)} - x^*) + x^* + o(\|x^{(k)} - x^*\|) \enspace .$$

Therefore, iterations of proximal coordinate descent for this problem lead to noisy linear iterations. Proof of Proposition 4 can be found in Appendix B.

Figure 4 shows the performance of Anderson extrapolation on a ℓ^2-regularised logistic regression problem:

$$\mathop{\arg\min}_{x \in \mathbb{R}^p} \sum_{i=1}^n \log(1 + e^{-y_i A_{i:} x}) + \frac{\lambda}{2} \|x\|_2^2 \enspace . \tag{17}$$

One can see that despite the better theoretical properties of the pseudo-symmetrized coordinate descent, Anderson acceleration on coordinate descent seems to work better on the cyclic coordinate descent. We thus choose to apply Anderson extrapolation on the cyclic coordinate descent (Algorithm 3), while adding a step checking the decrease of the objective function in order to ensure convergence. Finally, we can also use Algorithm 3 in the non smooth case where $g = \|\cdot\|_1$,

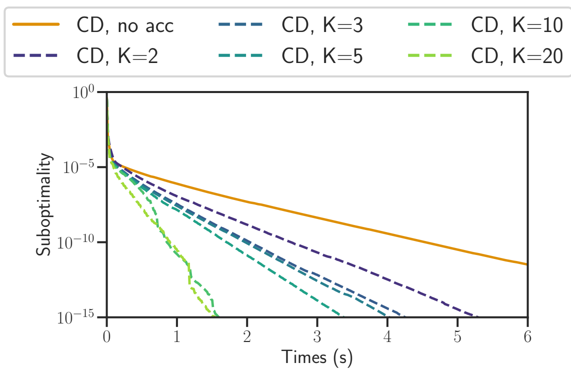

Figure 5: **Influence of K, quadratic, *rcv1*.** Influence of the number of iterates K used to perform Anderson extrapolation with coordinate descent (CD) on a quadratic with the *rcv1* dataset (2000 first columns).

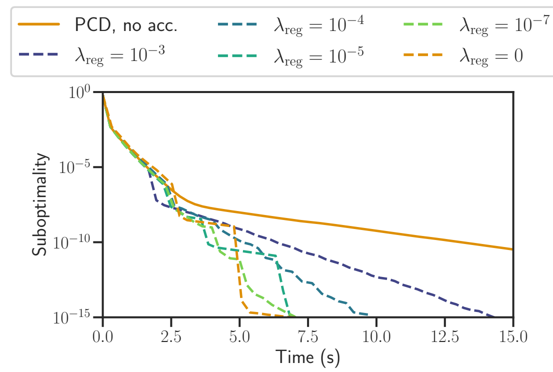

Figure 6: **Influence of λ_{reg}, sparse logitic regression, *rcv1*.** Influence of the regularization amount when solving a sparse logistic regression using Anderson extrapolation with proximal coordinate descent (PCD) on the *rcv1* dataset, $K = 5$, $\lambda = \lambda_{\max}/30$.

since coordinate descent achieves support identification when the solution is unique, after which the objective becomes differentiable. There is therefore a linear structure after a sufficient number of iterations (Massias et al., 2020, Prop. 10).

3 Experiments

An implementation relying on numpy, numba and cython (Harris et al., 2020; Lam et al., 2015; Behnel et al., 2011), with scripts to reproduce the figures, is available at `https://mathurinm.github.io/andersoncd`

We first show how we set the hyperparameters of Anderson extrapolation (Section 3.1). Then we show that Anderson extrapolation applied to proximal coordinate descent outperforms other first order algorithms on standard Machine Learning problems (Section 3.2).

3.1 Parameter setting

Anderson extrapolation relies on 2 hyperparameters: the number of extrapolated points K, and the amount of regularization eventually used when solving the linear system to obtain the coefficients $c \in \mathbb{R}^K$. Based on the conclusions of this section, we fix these parameters for all the subsequent experiments in Section 3.2: *no regularization and* $K = 5$.

Influence of the regularization. Scieur et al. (2016) provided accelerated complexity rates for *regularized* Anderson extrapolation: a term $\lambda_{\text{reg}} \|c\|^2$ is added to the objective of Equation (2). The closed-form formula for the coefficients is then $(U^\top U + \lambda_{\text{reg}} \operatorname{Id}_K)^{-1} \mathbf{1}_K / \mathbf{1}_K^\top (U^\top U + \lambda_{\text{reg}} \operatorname{Id}_K)^{-1} \mathbf{1}_K$.

However, similarly to Mai and Johansson (2019) and Poon and Liang (2020) we observed that regularizing the linear system does not seem necessary, and can even hurt the convergence speed. Figure 6 shows the influence of the regularization parameter on the convergence on the *rcv1* dataset for a sparse logistic regression problem, with $K = 5$ and $\lambda = \lambda_{\max}/30$. The more the optimization problem is regularized, the more the convergence speed is deteriorated. Thus *we choose not to regularize* when solving the linear system for the extrapolation coefficients. We simply check if the extrapolated point yields a lower objective function than the current regular iterate (see Algorithm 3).

Influence of K. Figure 5 shows the impact of K on the convergence speed. Although the performance depends on K, it seems that the dependency is loose, as for $K \in \{10, 20\}$ the acceleration is roughly the same. Therefore, we do not treat K as a parameter and fix it to $K = 5$.

Computational overhead of Anderson extrapolation. With nnz the number of nonzero coefficients, K epochs (*i.e.,* K updates of all coordinates) of CD *without Anderson acceleration* cost:

$$K \operatorname{nnz}(A) \enspace .$$

Every K epochs, Algorithm 3 requires to solve a $K \times K$ linear system. Thus, K epochs of CD *with Anderson acceleration* cost:

$$\underbrace{K \operatorname{nnz}(A)}_{K \text{ passes of CD}} + \underbrace{K^2 \operatorname{nnz}(w)}_{\text{form } U^\top U} + \underbrace{K^3}_{\text{solve system}} \enspace .$$

With our choice, $K = 5$, the overhead of Anderson acceleration is marginal compared to a gradient call: $K^2 + K \operatorname{nnz}(w) \ll \operatorname{nnz}(A)$. This can be observed in

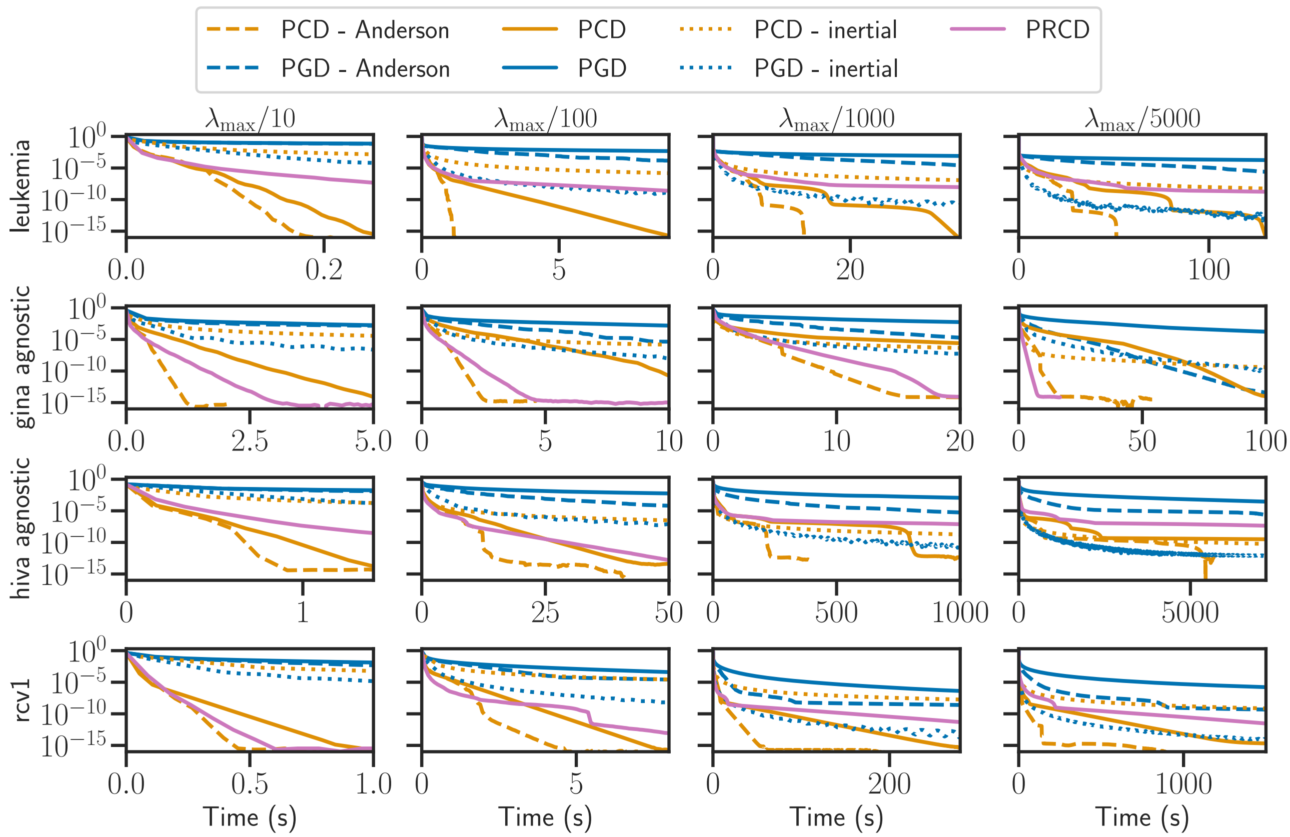

Figure 7: **Lasso, suboptimality.** Suboptimality as a function of time for the Lasso on multiple datasets and values of λ.

Figures 7 to 9: even before acceleration actually occurs, Anderson PCD is not slower than regular PCD.

3.2 Numerical comparison on Machine Learning problems

We compare multiple algorithms to solve popular Machine Learning problems: the Lasso, the elastic net, and sparse logistic regression (experiments on group Lasso are in Appendix A.2). The compared algorithms are the following:

- Proximal gradient descent (PGD, Combettes and Wajs 2005).
- Nesterov-like inertial PGD (FISTA, Beck and Teboulle 2009).
- Anderson accelerated PGD (Mai and Johansson, 2019; Poon and Liang, 2020).
- Proximal coordinate descent (PCD, Tseng and Yun 2009).
- Proximal coordinate descent with random index selection (PRCD, Richtárik and Takáč 2014).
- Inertial PCD (Lin et al., 2014; Fercoq and Richtárik, 2015).
- Anderson accelerated PCD (ours, Algorithm 3).

We use datasets from libsvm (Fan et al., 2008) and openml (Feurer et al., 2019) (Table 1), varying as much as possible to demonstrate the versatility of our approach. We also vary the convergence metric: we use suboptimality in the main paper, while graphs measuring the duality gaps are in Appendix A.

Table 1: Datasets characteristics

name	n	p	density
gina agnostic	3468	970	1
hiva agnostic	4229	1617	1
leukemia	72	7129	1
rcv1_train	20 242	19 960	$3.7\,10^{-3}$
real-sim	72 309	20 958	$2.4\,10^{-3}$
news20	19 996	632 983	$6.1\,10^{-4}$

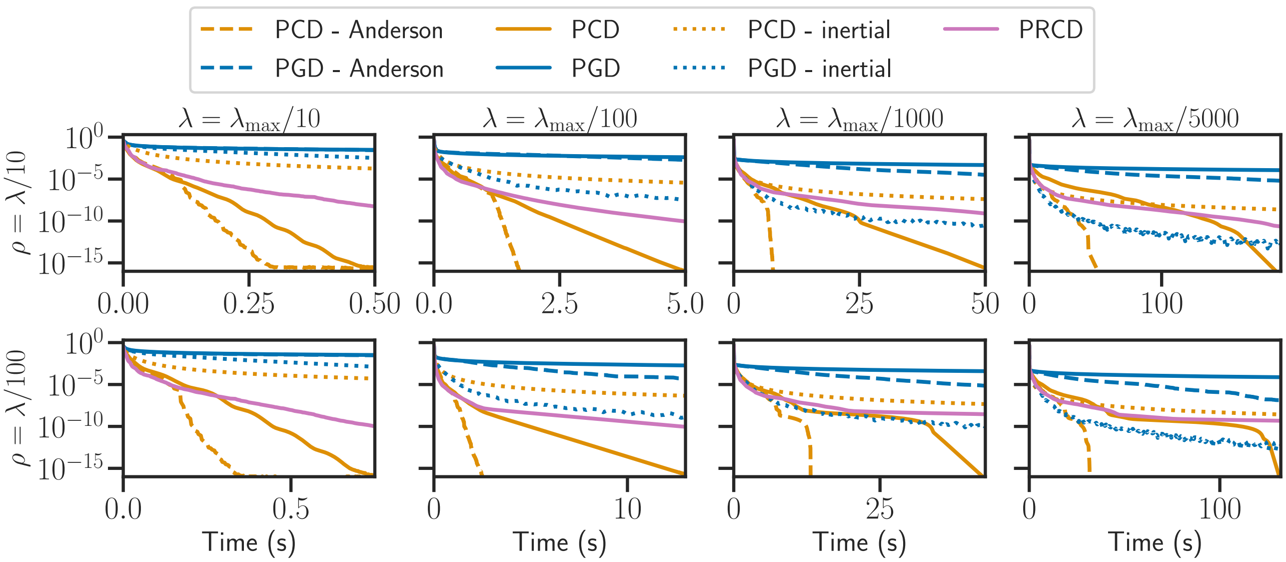

Figure 8: **Enet, suboptimality.** Suboptimality as a function of time for the elastic net on Leukemia dataset, for multiple values of λ and ρ.

Lasso. Figure 7 shows the suboptimality $f(x^{(k)}) - f(x^*)$ of the algorithms on the Lasso problem:

$$\arg\min_{x \in \mathbb{R}^p} \frac{1}{2} \|y - Ax\|^2 + \lambda \|x\|_1 \quad , \tag{18}$$

as a function of time for multiple datasets and values of λ. We parametrize λ as a fraction of $\lambda_{\max} = \|A^\top y\|_\infty$, smallest regularization strength for which $x^* = 0$. Figure 7 highlights the superiority of proximal coordinate descent over proximal gradient descent for Lasso problems on real-world datasets, and the benefits of extrapolation for coordinate descent. It shows that Anderson extrapolation can lead to a significant gain of performance. In particular Figure 7 shows that without restart, inertial coordinate descent (Lin et al., 2014; Fercoq and Richtárik, 2015) can slow down the convergence, despite its accelerated rate. Note that the smaller the value of λ, the harder the optimization: when λ decreases, more time is needed to reach a fixed suboptimality. The smaller λ is (*i.e.,* the harder the problem), the more efficient Anderson extrapolation is.

Other convergence metrics can be considered, since $f(x^*)$ is unknown to the practitioner: for the Lasso, it is also common to use the duality gap as a stopping criterion (Massias et al., 2018). Thus, for completeness, we provide Figure 10 in appendix, which shows the duality gap as a function of the number of iterations. With this metric of convergence, Anderson PCD also significantly outperforms its competitors.

Elastic net. Anderson extrapolation is easy to extend to other estimators than the Lasso. Figure 8 (and Figure 11 in appendix) show the superiority of the Anderson extrapolation approach over proximal gradient descent and its accelerated version for the elastic net problem (Zou and Hastie, 2005):

$$\arg\min_{x \in \mathbb{R}^p} \frac{1}{2n} \|y - Ax\|^2 + \lambda \|x\|_1 + \frac{\rho}{2} \|x\|_2^2 \quad . \tag{19}$$

In particular, we observe that the more difficult the problem, the more useful the Anderson extrapolation: it is visible on Figures 8 and 11 that going from $\rho = \lambda/10$ to $\rho = \lambda/100$ lead to an increase in time to achieve similar suboptimality for the classical proximal coordinate descent, whereas the impact is more limited on the coordinate descent with Anderson extrapolation.

Finally, for a nonquadratic data-fit, here sparse logistic regression, we still demonstrate the applicability of extrapolated coordinate descent.

Sparse logistic regression. Figure 9 represents the suboptimality as a function of time on a sparse logistic regression problem:

$$\arg\min_{x \in \mathbb{R}^p} \sum_{i=1}^{n} \log(1 + e^{-y_i A_{i:} x}) + \lambda \|x\|_1 \quad , \tag{20}$$

for multiple datasets and values of λ. We parametrize λ as a fraction of $\lambda_{\max} = \|A^\top y\|_\infty / 2$. As for the Lasso and the elastic net, the smaller the value of λ, the harder the problem and Anderson CD outperforms its competitors.

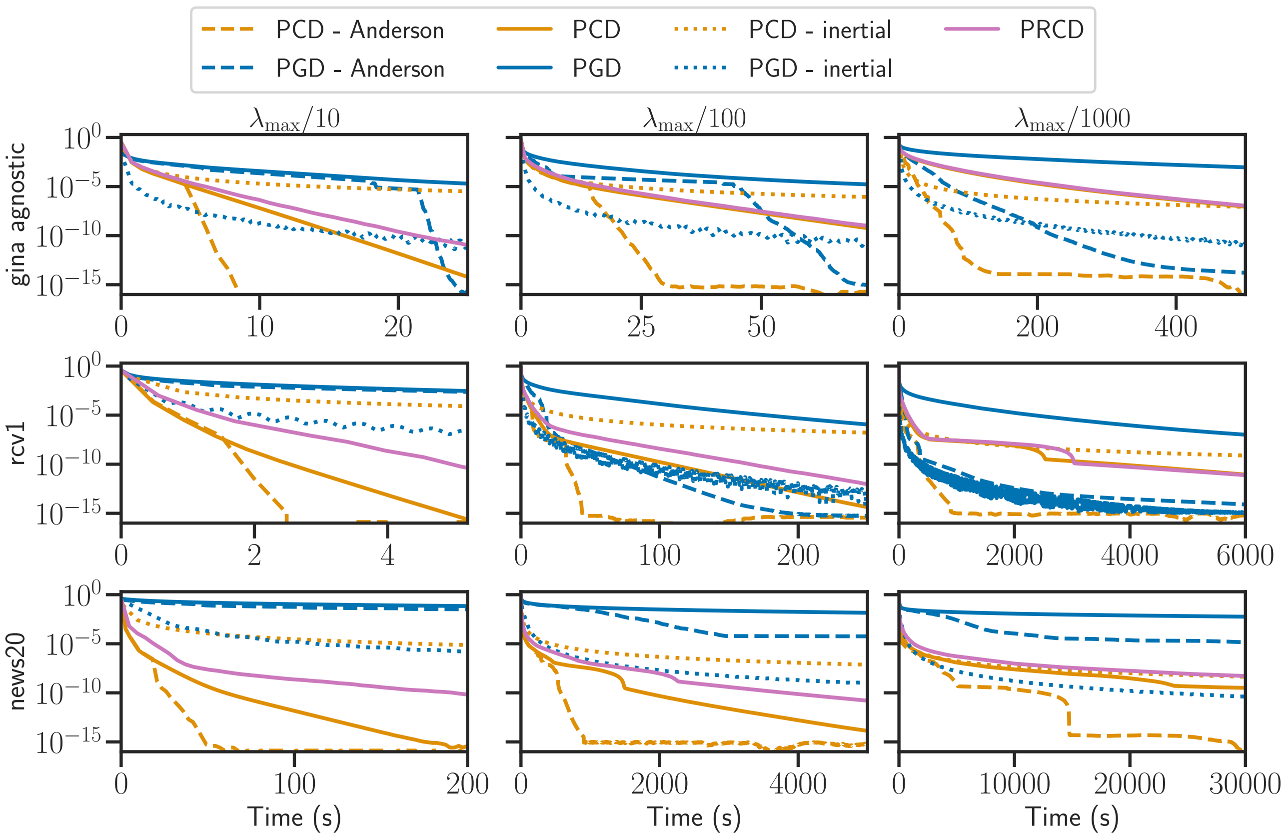

Figure 9: ℓ_1**-regularised logistic regression, suboptimality.** Suboptimality as a function of time for ℓ_1-regularized logistic regression on multiple datasets and values of λ.

Conclusion In this work, we have proposed to accelerate coordinate descent using Anderson extrapolation. We have exploited the fixed point iterations followed by coordinate descent iterates on multiple Machine Learning problems to improve their convergence speed. We have circumvented the non-symmetricity of the iteration matrices by proposing a pseudo-symmetric version for which accelerated convergence rates have been derived. In practice, we have performed an extensive validation to demonstrate large benefits on multiple datasets and problems of interests. For future works, the excellent performance of Anderson extrapolation for cyclic coordinate descent calls for a more refined analysis of the known bounds, through a better analysis of the spectrum and numerical range of the iteration matrices.

Acknowledgements

Part of this work has been carried out at the Machine Learning Genoa (MaLGa) center, Università di Genova (IT). M. M. acknowledges the financial support of the European Research Council (grant SLING 819789). This work was partially funded by the ERC Starting Grant SLAB ERC-StG-676943.

References

D. G. Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM*, 12(4):547–560, 1965.

M. Barré, A. Taylor, and A. d'Aspremont. Convergence of constrained anderson acceleration. *arXiv preprint arXiv:2010.15482*, 2020.

A. Beck. *First-Order Methods in Optimization*, volume 25. SIAM, 2017.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31 –39, 2011.

Q. Bertrand, Q. Klopfenstein, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation of lasso-type models for hyperparameter optimization. *ICML*, 2020.

M. Blondel and F. Pedregosa. Lightning: large-scale linear classification, regression and ranking in python, 2016.

R. Bollapragada, D. Scieur, and A. d'Aspremont. Nonlinear acceleration of momentum and primal-dual algorithms. *arXiv preprint arXiv:1810.04539*, 2018.

C. Brezinski, M. Redivo-Zaglia, and Y. Saad. Shanks sequence transformations and anderson acceleration. *SIAM Review*, 60(3):646–669, 2018.

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

M. Crouzeix. Bounds for analytical functions of matrices. *Integral Equations and Operator Theory*, 48 (4):461–477, 2004.

M. Crouzeix. Numerical range and functional calculus in hilbert space. *Journal of Functional Analysis*, 244(2):668–690, 2007.

M. Crouzeix and C. Palencia. The numerical range is a (1+2)-spectral set. *SIAM Journal on Matrix Analysis and Applications*, 38(2):649–655, 2017.

R. P. Eddy. Extrapolating to the limit of a vector sequence. In *Information linkage between applied mathematics and industry*, pages 387–396. Elsevier, 1979.

V. Eyert. A comparative study on methods for convergence acceleration of iterative vector sequences. *Journal of Computational Physics*, 124(2):271–285, 1996.

R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

O. Fercoq and P. Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

M. Feurer, J. N. van Rijn, A. Kadra, P. Gijsbers, N. Mallik, S. Ravi, A. Müller, J. Vanschoren, and F. Hutter. Openml-python: an extensible python api for openml. *arXiv:1911.02490*, 2019.

J. Friedman, T. Hastie, and R. Tibshirani. glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1(4), 2009.

J. Friedman, T. J. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.*, 33(1):1, 2010.

A. Fu, J. Zhang, and S. Boyd. Anderson accelerated Douglas-Rachford splitting. *arXiv preprint arXiv:1908.11482*, 2019.

G. H. Golub and R. S. Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order richardson iterative methods. *Numerische Mathematik*, 3(1):147–156, 1961.

R. Gribonval and M. Nikolova. A characterization of proximity operators. *Journal of Mathematical Imaging and Vision*, 62(6):773–789, 2020.

C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al. Array programming with numpy. *arXiv preprint arXiv:2006.10256*, 2020.

M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6): 409–436, 1952.

Q. Klopfenstein, Q. Bertrand, A. Gramfort, J. Salmon, and S. Vaiter. Model identification and local linear convergence of coordinate descent. *arXiv preprint arXiv:2010.11825*, 2020.

S. K. Lam, A. Pitrou, and S. Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6, 2015.

Q. Lin, Z. Lu, and L. Xiao. An accelerated proximal coordinate gradient method. In *NeurIPS*, pages 3059–3067. 2014.

V. V. Mai and M. Johansson. Anderson acceleration of proximal gradient methods. In *ICML*. 2019.

J. Mairal. *Sparse coding for machine learning, image processing and computer vision*. PhD thesis, École normale supérieure de Cachan, 2010.

M. Massias, A. Gramfort, and J. Salmon. Celer: a fast solver for the lasso with dual extrapolation. 2018.

M. Massias, S. Vaiter, A. Gramfort, and J. Salmon. Dual extrapolation for sparse generalized linear models. *J. Mach. Learn. Res.*, 2020.

Y. Nesterov. A method for solving a convex programming problem with rate of convergence $O(1/k^2)$. *Soviet Math. Doklady*, 269(3):543–547, 1983.

Y. Nesterov. *Introductory lectures on convex optimization*, volume 87 of *Applied Optimization*. Kluwer Academic publishers, Boston, MA, 2004.

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *ICML*, page 78, 2004.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.

C. Poon and J. Liang. Trajectory of alternating direction method of multipliers and adaptive acceleration. In *NeurIPS*, pages 7357–7365, 2019.

C. Poon and J. Liang. Geometry of first-order methods and adaptive acceleration. *arXiv preprint arXiv:2003.03910*, 2020.

P. Pulay. Convergence acceleration of iterative sequences. the case of scf iteration. *Chemical Physics Letters*, 73(2):393–398, 1980.

P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

D. Scieur. Generalized framework for nonlinear acceleration. *arXiv preprint arXiv:1903.08764*, 2019.

D. Scieur, A. d'Aspremont, and F. Bach. Regularized nonlinear acceleration. In *Advances In Neural Information Processing Systems*, pages 712–720, 2016.

S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *arXiv preprint arXiv:1309.2375*, 2013.

H.-J. Shi, S. Tu, Y. Xu, and W. Yin. A primer on coordinate descent algorithms. *arXiv preprint arXiv:1610.00040*, 2016.

A. Sidi. *Vector extrapolation methods with applications*. SIAM, 2017.

D. A. Smith, W. F. Ford, and A. Sidi. Extrapolation methods for vector sequences. *SIAM review*, 29(2): 199–233, 1987.

R. Sun and Y. Ye. Worst-case complexity of cyclic coordinate descent: $\mathcal{O}(n^2)$ gap with randomized version. *Mathematical Programming*, pages 1–34, 2019.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 58 (1):267–288, 1996.

P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, 2009.

S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

P. Wynn. Acceleration techniques for iterated vector and matrix problems. *Mathematics of Computation*, 16(79):301–322, 1962.

J. Zhang, B. O'Donoghue, and S. Boyd. Globally convergent type-I Anderson acceleration for non-smooth fixed-point iterations. *arXiv preprint arXiv:1808.03971*, 2018.

H. Zou and T. J. Hastie. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 67(2):301–320, 2005.