REGULARIZED NONLINEAR ACCELERATION

DAMIEN SCIEUR, ALEXANDRE D'ASPREMONT, AND FRANCIS BACH

ABSTRACT. We describe a convergence acceleration technique for generic optimization problems. Our scheme computes estimates of the optimum from a nonlinear average of the iterates produced by any optimization method. The weights in this average are computed via a simple linear system, whose solution can be updated online. This acceleration scheme runs in parallel to the base algorithm, providing improved estimates of the solution on the fly, while the original optimization method is running. Numerical experiments are detailed on classical classification problems.

1. INTRODUCTION

Suppose we seek to solve the following optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1}$$

in the variable $x \in \mathbb{R}^n$, where $f(x)$ is strongly convex with parameter μ with respect to the Euclidean norm, and has a Lipschitz continuous gradient with parameter L with respect to the same norm. Assume we solve this problem using an iterative algorithm of the form

$$x_{i+1} = g(x_i), \quad \text{for } i = 1, ..., k, \tag{2}$$

where $x_i \in \mathbb{R}^n$ and k is the number of iterations. Here, we will focus on improving our estimates of the solution to problem (1) by tracking only the sequence of iterates x_i produced by an optimization algorithm, without any further calls to oracles on $g(x)$.

Since the publication of Nesterov's optimal first-order smooth convex minimization algorithm [Nesterov, 1983], a significant effort has been focused on either providing more interpretable views on current acceleration techniques, or on replicating these complexity gains using different, more intuitive schemes. Early efforts sought to directly extend the original acceleration result in [Nesterov, 1983] to broader function classes [Nemirovskii and Nesterov, 1985], allow for generic metrics, line searches or simpler proofs [Beck and Teboulle, 2009; Nesterov, 2003] or produce adaptive accelerated algorithms [Nesterov, 2015], etc. More recently however, several authors [Drori and Teboulle, 2014; Lessard et al., 2016] have started using classical results from control theory to obtain numerical bounds on convergence rates that match the optimal rates. Others have studied the second order ODEs obtained as the limit for small step sizes of classical accelerated schemes, to better understand their convergence [Su et al., 2014; Wibisono and Wilson, 2015]. Finally, recent results have also shown how to wrap classical algorithms in an outer optimization loop, to accelerate convergence and reach optimal complexity bounds [Lin et al., 2015] for certain structured problems.

Here, we take a significantly different approach to convergence acceleration stemming from classical results in numerical analysis. We use the iterates produced by any (converging) optimization algorithm, and estimate the solution directly from this sequence, assuming only some regularity conditions on the function to minimize. Our scheme is based on the idea behind Aitken's Δ^2-algorithm [Aitken, 1927], generalized as the Shanks transform [Shanks, 1955], whose recursive formulation is known as the ε-algorithm [Wynn,

Date: July 29, 2016.

Key words and phrases. Acceleration, ε-algorithm, extrapolation.

1956] (see e.g. [Brezinski, 1977; Sidi et al., 1986] for a survey). In a nutshell, these methods fit geometrical models to linearly converging sequences, then extrapolate their limit from the fitted model.

In a sense, this approach is more statistical in nature. It assumes an approximately linear model holds for iterations near the optimum, and estimates this model using the iterates. In fact, Wynn's algorithm [Wynn, 1956] is directly connected to the Levinson-Durbin algorithm [Levinson, 1949; Durbin, 1960] used to solve Toeplitz systems recursively and fit autoregressive models (the Shanks transform solves Hankel systems, but this is essentially the same problem [Heinig and Rost, 2011]). The key difference in these extrapolation techniques is that estimating the autocovariance operator A is not required, as we only focus on the limit. Moreover, the method presents strong links with the conjugate gradient when applied to unconstrained quadratic optimization.

We start from a slightly different formulation of these techniques known as minimal polynomial extrapolation (MPE) [Sidi et al., 1986; Smith et al., 1987] which uses the minimal polynomial of the linear operator driving iterations to estimate the optimum by nonlinear averaging (i.e. using weights in the average which are nonlinear functions of the iterates). So far, for all the techniques cited above, no proofs of convergence of these estimates were given when the estimation process became unstable.

Our contribution here is first to give a novel formulation of approximate MPE. We then regularize this procedure to produce explicit bounds on the distance to optimality by controlling stability, thus explicitly quantifying the acceleration provided by these techniques. We show in several numerical examples that these stabilized estimates often speed up convergence by an order of magnitude. Furthermore this acceleration scheme runs in parallel with the original algorithm, providing improved estimates of the solution on the fly, while the original method is progressing.

The paper is organized as follows. In Section 2.1 we recall basic results behind MPE for linear iterations and introduce in Section 2.3 an approximate version of MPE, connecting it to the conjugate gradient method. Then, in Section 2.4, we generalize these results to nonlinear iterations and show in Section 3.1 how to fully control the impact of nonlinearity. We use these results to derive explicit bounds on the acceleration performance of our estimates. Finally, we present numerical results in Section 4.

2. Approximate Minimal Polynomial Extrapolation

In what follows, we recall the key arguments behind *minimal polynomial extrapolation (MPE)* as derived in [Cabay and Jackson, 1976] or also [Smith et al., 1987]. We then explain a variant called *approximate minimal polynomial extrapolation (AMPE)* which allows to control the number of iterates used in the extrapolation, hence reduces its computational complexity. We begin by a simple description of the method for linear iterations, then extend these results to the generic nonlinear case. Finally, we characterize the acceleration factor provided by a regularized version of AMPE, using regularity properties of the function $f(x)$, and the result of a Chebyshev-like, tractable polynomial optimization problem.

2.1. Linear Iterations.

Let's assume for now that the iterative algorithm in (2) is in fact linear, with

$$x_i = A(x_{i-1} - x^*) + x^*, \tag{3}$$

where $A \in \mathbb{R}^{n\times n}$ (not necessarily symmetric) and $x^* \in \mathbb{R}^n$. We assume that 1 is not an eigenvalue of A (a sufficient condition is that the spectral radius of A is below one, so the algorithm converges to x^*), implying that the iterations in (3) admits a unique fixed point x^*.

Example: gradient method for minimizing a quadratic function. We illustrate (3) on a concrete example, minimizing the quadratic function $f(x) = \frac{1}{2}\|Bx - b\|_2^2$. The fixed-step gradient descend on $f(x)$ works as follows

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

where $f(x)$ is a L-smooth convex function. This scheme becomes here

$$x_{k+1} = x_k - \frac{1}{L}(B^T B x_k/2 - B^T b)$$

Using optimality conditions on x^*, this is also

$$x_{k+1} = x_k - \frac{1}{L} B^T B(x_k - x^*)/2$$

or again

$$x_{k+1} - x^* = \left(I - \frac{1}{2L} B^T B\right)(x_k - x^*),$$

which is exactly equation (3) with $A = I - \frac{1}{2L} B^T B$.

2.2. **Minimal polynomial extrapolation.** We now recall the *minimal polynomial extrapolation* (MPE) method as described in [Smith et al., 1987], starting with the following definition.

Definition 2.1. *Given $A \in \mathbb{R}^{n\times n}$, such that 1 is not an eigenvalue of A and $v \in \mathbb{R}^n$, the minimal polynomial of A with respect to the vector v is the lowest degree polynomial $p(x)$ such that*

$$p(A)v = 0, \qquad p(1) = 1.$$

Note that the degree of $p(x)$ is always less than n and that condition $p(1) = 1$ makes p unique. Note also that assuming one is not an eigenvalue of A means $p(1) \neq 0$, since $p(A) = 0$ if and only if $p(\lambda_i(A)) = 0$ for $i = 1, \ldots, n$, where $\lambda(A)$ is the spectrum of A. This means that we can set $p(1) = 1$ without loss of generality. Given an initial iterate x_0, MPE starts by forming a matrix U whose columns are the increments $x_{i+1} - x_i$, with

$$U_i = x_{i+1} - x_i = (A - I)(x_i - x^*) = (A - I)A^i(x_0 - x^*). \tag{4}$$

Let $p(x) = \sum_{i=0}^k c_i x^i$ be the minimal polynomial of A with respect to the vector U_0, so

$$0 = p(A)U_0 = \sum_{i=0}^k c_i A^i U_0 = \sum_{i=0}^k c_i U_i, \quad \text{and} \quad p(1) = \sum_{i=0}^k c_i = 1. \tag{5}$$

Writing $U = [U_0, U_1, ..., U_k]$, this means we can find the coefficients of p by solving the linear system

$$Uc = 0, \ \mathbf{1}^T c = 1.$$

In this case, the fixed point x^* of (3) can be computed *exactly* as follows. We have

$$\begin{aligned} 0 = \sum_{i=0}^k c_i A^i U_0 = \sum_{i=0}^k c_i A^i (A - I)(x_0 - x^*) &= (A - I)\sum_{i=0}^k c_i A^i (x_0 - x^*) \\ &= (A - I)\sum_{i=0}^k c_i (x_i - x^*), \end{aligned}$$

hence, using the fact that one is not an eigenvalue of A, together with $\mathbf{1}^T c = p(1) = 1$, we finally get

$$x^* = \sum_{i=0}^k c_i x_i.$$

This means that x^* is obtained by *averaging* iterates using the coefficients in c, but the averaging is called nonlinear here because the coefficients of c vary with the iterates themselves.

2.3. **Approximate Minimal Polynomial Extrapolation (AMPE).** Suppose now that we only compute a fraction of the iterates x_i used in the MPE procedure. While the number of iterates k might be smaller than the degree of the minimal polynomial of A with respect to U_0, we can still try to make the quantity $p_k(A)U_0$ small, with $p_k(x)$ now a polynomial of degree at most k. The corresponding difference matrix $U = [U_0, U_1, ..., U_k] \in \mathbb{R}^{n\times(k+1)}$ is rectangular.

This is also known as the Eddy-Mešina method [Mešina, 1977; Eddy, 1979] or Reduced Rank Extrapolation with arbitrary k (see [Smith et al., 1987, §10]). The objective here is similar to (5), but the system is now overdetermined because $k < \deg(P)$. We will thus choose c to make $\|Uc\|_2 = \|p(A)U_0\|_2$ as small as possible, for some polynomial p such that $p(1) = 1$. We thus solve for

$$c^* \triangleq \underset{\mathbf{1}^T c=1}{\operatorname{argmin}} \|Uc\|_2 \tag{AMPE}$$

in the variable $c \in \mathbb{R}^{k+1}$. The optimal value of this problem is decreasing with k, satisfies $\|Uc^*\|_2 = 0$ when k is greater than the degree of the minimal polynomial at U_0, and controls the approximation error in x^* through equation (4). This basic extrapolation algorithm suffers from stability issues and we will see in further sections how to stabilize its solution.

2.3.1. *Error bounds.* We can get a crude bound on $\|Uc^*\|_2$ using Chebyshev polynomials, together with an assumption on the range of the spectrum of the matrix A.

Proposition 2.2. *Let A be symmetric, $0 \preceq A \preceq \sigma I$ with $\sigma < 1$ and c^* be the solution of (AMPE). Then*

$$\left\| \sum_{i=0}^{k} c_i^* x_i - x^* \right\|_2 \leq \kappa(A-I) \frac{2\zeta^k}{1+\zeta^{2k}} \|x_0 - x^*\|_2 \tag{6}$$

where $\kappa(A-I)$ is the condition number of the matrix $A-I$ and ζ is given by

$$\zeta = \frac{1-\sqrt{1-\sigma}}{1+\sqrt{1-\sigma}}, \tag{7}$$

which is always smaller than σ.

Proof. Setting $U_i = (A-I)(x_i - x^*)$, we have

$$\begin{aligned} \left\| \sum_{i=0}^{k} c_i^* x_i - x^* \right\|_2 &= \left\| (I-A)^{-1} \sum_{i=0}^{k} c_i^* U_i \right\|_2 \\ &\leq \left\| (I-A)^{-1} \right\|_2 \|Uc^*\|_2. \end{aligned}$$

Assume A symmetric and $0 \preceq A \preceq \sigma I \prec I$, we have

$$\|Uc^*\|_2 = \|p^*(A)U_0\|_2 \leq \|U_0\|_2 \min_{\substack{p\in\mathbb{R}[x] \\ p(1)=1}} \|p(A)\|_2 \leq \|U_0\|_2 \min_{\substack{p\in\mathbb{R}[x] \\ p(1)=1}} \max_{0\preceq A\preceq\sigma I} \|p(A)\|_2. \tag{8}$$

We have $A = Q\,\mathbf{diag}(\lambda)Q^T$ where Q is unitary and $\lambda \in \mathbb{R}^n$, hence

$$\max_{0\preceq A\preceq\sigma I} \|p(A)\|_2 = \max_{0\preceq\lambda\preceq\sigma\mathbf{1}} \|p(\mathbf{diag}(\lambda))\|_2 = \max_{0\preceq\lambda\preceq\sigma\mathbf{1}} \max_i |p(\lambda_i)| = \max_{0\leq x\leq\sigma} |p(x)|.$$

We then get

$$\|Uc^*\|_2 \leq \|U_0\|_2 \min_{\substack{p\in\mathbb{R}[x] \\ p(1)=1}} \max_{0\leq x\leq\sigma} |p(x)|.$$

Now, Golub and Varga [1961] show that the polynomial optimization problem on the right-hand side can be solved exactly using Chebyshev polynomials. Let C_k be the Chebyshev polynomial of degree k. By definition, $C_k(x)$ is a monic polynomial (i.e. a polynomial whose leading coefficient is one) which solves

$$C_k(x) = \underset{\{p\in\mathbb{R}_k[x]:\, p_k=1\}}{\operatorname{argmin}} \max_{x\in[0,1]} |p(x)|.$$

Golub and Varga [1961] use a variant of $C_k(x)$ to solve the related minimax problem

$$\min_{\substack{p\in\mathbb{R}[x]\\ p(1)=1}} \max_{0\le x\le\sigma} |p(x)| \tag{9}$$

whose solution is a rescaled Chebyshev polynomial given by

$$T(x) = \frac{C_k(t(x))}{C_k(t(1))}, \quad \text{where} \quad t(x) = \frac{2x-\sigma}{\sigma}, \tag{10}$$

where $t(x)$ is simply a linear mapping from interval $[0,\sigma]$ to $[0,1]$. Moreover,

$$\min_{\substack{p\in\mathbb{R}[x]\\ p(1)=1}} \max_{0\le x\le\sigma} |p(x)| = \max_{0\le x\le\sigma} |T_k(x)| = |T_k(\sigma)| = \frac{2\zeta^k}{1+\zeta^{2k}}, \tag{11}$$

where ζ is given by

$$\zeta = \frac{1-\sqrt{1-\sigma}}{1+\sqrt{1-\sigma}} < \sigma < 1.$$

Since $\|U_0\|_2 = \|(A-I)(x_0-x^*)\|_2 \le \|A-I\|_2\|x_0-x^*\|$, we can bound (8) by

$$\|Uc^*\|_2 \le \|U_0\|_2 \min_{\substack{p\in\mathbb{R}[x]\\ p(1)=1}} \max_{0\le x\le\sigma} |p(x)| \le \frac{2\zeta^k}{1+\zeta^{2k}}\|A-I\|_2\|x_0-x^*\|_2,$$

which yields the desired result. ■

Note that, when solving quadratic optimization problems (where gradient iterations are linear), the rate in this bound matches that obtained using the optimal method in [Nesterov, 2003]. Outside of the normalization constraint, this is very similar to the convergence analysis of Lanczos' method.

2.3.2. *AMPE versus conjugate gradient.* The rate of convergence obtained above also matches that of the conjugate gradient within a factor $\kappa(A-I)$. Indeed, AMPE has a strong link with the conjugate gradient. Denote $\|v\|_B = \sqrt{v^T B v}$ the norm induced by the positive definite matrix B. Also, assume we want to solve $Bx = b$ using conjugate gradient method. By definition, at the k-th iteration, the conjugate gradient computes an approximation s of x^* which follows

$$s = \operatorname*{argmin}_{x\in\mathcal{K}_k} \|x-x^*\|_B,$$

where $\mathcal{K}_k = \text{span}\{b, Bb, ..., B^{k-1}b\} = \text{span}\{Bx^*, B^2x^*, ..., B^kx^*\}$ is called a Krylov subspace. Since the constraint $x\in\mathcal{K}_k$ impose us to build x from a linear combination of the basis of $\mathcal{K}_k$, we can write

$$x = \sum_{i=1}^{k} c_i B^i x^* = q(B)x^*,$$

where $q(x)$ is a polynomial of degree k. So the conjugate gradient method solves

$$s = \operatorname*{argmin}_{q\in\mathbb{R}_k[x]q(0)=0} \|q(B)x^*-x^*\|_B = \operatorname*{argmin}_{\substack{\hat{q}\in\mathbb{R}_k[x]\\ \hat{q}(0)=1}} \|\hat{q}(B)x^*\|_B,$$

which is very similar to the equations in (AMPE). However, while conjugate gradient has access to an oracle giving the result of the product between B and any vector v, the AMPE procedure can only use the iterations produced by (3) (meaning that AMPE procedure do not require knowledge of B). Moreover, the convergence of AMPE is analyzed in another norm ($\|\cdot\|_2$ instead of $\|\cdot\|_B$), which explains why a condition number appears in the rate of convergence of AMPE (6).

2.3.3. *Chebyshev's acceleration and Nesterov's accelerated gradient.* In proposition 2.2, we bounded the rate of convergence of the AMPE procedure using Chebyshev polynomials. In fact, this is exactly the idea behind Chebyshev's semi-iterative method, which uses these coefficients in order to accelerate gradient descent on quadratic functions. Here, we present Chebyshev semi-iterative acceleration and its analysis, then use the same analysis on Nesterov's method. These points were also discussed in [Hardt, 2013].

Assume as above that we use the gradient method to minimize $f(x) = \frac{1}{2}\|Bx - b\|_2$, we get the following recurrence

$$x_{k+1} - x^* = \left(I - \frac{1}{L}B\right)(x_k - x^*) = A(x_k - x^*).$$

Where $A = I - \frac{1}{L}B$. We see easily that

$$x_k = x^* + A^k(x_0 - x^*).$$

Since $\|A\|_2 \leq 1 - \frac{\mu}{L}$, the rate of convergence is $\|x_k - x^*\|_2 \leq (1 - \frac{\mu}{L})^k \|x_0 - x^*\|_2$. Moreover, if we combine linearly the vectors x_i using coefficients c_i from 0 to k, we get

$$\sum_{i=0}^{k} c_i x_i = \sum_{i=0}^{k} c_i A^i(x_0 - x^*) + \sum_{i=0}^{k} c_i x^* = p_k(A)(x_0 - x^*) + p_k(1)x^*$$

If we force $p_k(1) = 1$, we see that we need to make the value $\|p_k(A)(x_0 - x^*)\|_2$ as small as possible in order to have the best approximation of x^*. Fixing the coefficients c_i a priori (unlike AMPE method, which computes these coefficient from the iterates x_i) so that $\|p_k(A)\|_2$ is small for any A such that $\mu I \preceq A \preceq LI$, means solving

$$p_k(x) = \arg \min_{\substack{p \in \mathbb{R}[x] \\ p(1)=1}} \max_{0 \preceq A \preceq \sigma I} \|p(A)\|_2 .$$

As above, the solution to this problem is $p_k(x) = T_k(x)$, defined in (10), using parameter $\sigma = 1 - \mu/L$. Furthermore, the Chebyshev polynomials can be constructed using a three-terms recurrence

$$C_k(x) = xC_{k-1}(x) - C_{k-2}(x)$$

and the same holds for $T_k(x)$ (see Appendix B for more details), with

$$\begin{aligned} \alpha_k &= t(1)\alpha_{k-1} - \alpha_{k-2} \\ z_{k-1} &= y_{k-1} - \frac{1}{L}(By_{k-1} - b) \\ y_k &= \frac{\alpha_{k-1}}{\alpha_k}\left(\frac{2z_{k-1}}{\sigma} - y_{k-1}\right) - \frac{\alpha_{k-2}}{\alpha_k} y_{k-2} \end{aligned}$$

This scheme looks very similar to Nesterov's accelerated gradient method, which reads

$$\begin{aligned} z_{k-1} &= y_{k-1} - \frac{1}{L}(By_{k-1} - b) \\ y_k &= z_{k-1} + \beta_k(z_{k-1} - z_{k-2}) \end{aligned}$$

Compared with Chebyshev acceleration, Nesterov's scheme is iteratively building a polynomial $N_k(x)$ with $y_k - y^* = N_k(A)(y_0 - x^*)$. If we replace z_k by its definition in the expression of y_k in the Nesterov's scheme we get the following recurrence of order two

$$\begin{aligned} y_k - x^* &= (1 + \beta_k)A(y_{k-1} - y^*) - \beta_k A(y_{k-2} - y^*) \\ &= A\left((1 + \beta_k)N_{k-1}(A) - \beta_k N_{k-2}(A)\right)(y_0 - x^*). \end{aligned}$$

which also reads

$$N_k(x) = x((1 + \beta_k)N_{k-1}(x) - \beta_k N_{k-2}(x)),$$

with initial conditions $N_0(x) = 1$ and $N_1(x) = x$. Notice that as for Chebyshev polynomial, $N_k(1) = 1$ for all k.

When minimizing smooth strongly convex functions with Nesterov's method, we use

$$\beta_k = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}.$$

Moreover, empirically at least, the maximum value of $N_k(x)$ in the interval $[0, \sigma]$ is $N_k(\sigma)$. We conjecture that this always holds. We thus have the following recurrence

$$N_k(\sigma) \quad = \quad \sigma\left((1+\beta)\, N_{k-1}(\sigma) - \beta N_{k-2}(\sigma)\right)$$

To get linear convergence with rate r, we need $N_k \leq r N_{k-1} \leq r^2 N_{k-2}$, or again

$$N_k(\sigma) \leq \sigma\left((1+\beta)\, r N_{k-2}(\sigma) - \beta N_{k-2}(\sigma)\right) = \sigma\left((1+\beta)\, r - \beta\right) N_{k-2}(\sigma).$$

Now, consider the condition

$$\sigma\left((1+\beta)\, r - \beta\right) \leq r^2.$$

We have that Nesterov's coefficients and rate, i.e. $\beta = (1 - \sqrt{\mu/L})/(1 + \sqrt{\mu/L})$ and $r = (1 - \sqrt{\mu/L})$, satisfy this condition, showing that Nesterov's method converges with a rate at least $r = (1 - \sqrt{\mu/L})$ on quadratic problems. This provides an alternate proof of Nesterov's acceleration result on these problemss using Chebyshev polynomials (provided the conjecture on $N(\sigma)$ holds).

2.4. **Nonlinear Iterations.** We now go back to the general case where the iterative algorithm is nonlinear, with

$$\tilde{x}_{i+1} = g(\tilde{x}_i), \quad \text{for } i = 1, ..., k, \tag{12}$$

where $\tilde{x}_i \in \mathbb{R}^n$ and function $g(x)$ has a symmetric Jacobian at point x^*. We also assume that the method has a unique fixed point written x^* and linearize these iterations around x^*, to get

$$\tilde{x}_i - x^* = A(\tilde{x}_{i-1} - x^*) + e_i, \tag{13}$$

where A is now the Jacobian matrix (i.e. the first derivative) of g taken at the fixed point x^* and $e_i \in \mathbb{R}^n$ is a second order error term $\|e_i\|_2 = O(\|\tilde{x}_{i-1} - x^*\|_2^2)$. Note that, by construction, the linear and nonlinear models share the same fixed point x^*. We write x_i the iterates that would be obtained using the asymptotic linear model (starting at x_0)

$$x_i - x^* = A(x_{i-1} - x^*).$$

After k iterations, the accumulated error with respect to this asymptotic linear model is

$$\tilde{x}_k - x_k = \sum_{i=1}^{k} A^{k-i} e_i.$$

As in §2.1, we define the increment matrix U such that $U_i = x_{i+1} - x_i$ for $i = 0, \ldots, k-1$, and a matrix $\tilde{U}$ constructed from $\tilde{x}$ in a similar way. Writing $E = \tilde{U} - U$ the corresponding error matrix, we have

$$E_{j+1} = \sum_{i=1}^{j+1} A^{j-i+1} e_i - \sum_{i=1}^{j} A^{j-i} e_i = (A - I)\left(\sum_{i=1}^{j} A^{j-i} e_i\right) + e_{j+1}, \quad j = 0, \ldots, k-1.$$

Assuming that A is a contraction, i.e. that $\|A\|_2 < 1$, we can derive a crude bound on $\|E\|_2$ using the fact that

$$\|E_{j+1}\|_2 \leq \|A - I\|_2 \sum_{i=1}^{j} \|A\|_2^{j-i} \|e_i\|_2 + \|e_{j+1}\|_2 \leq \frac{(1 + \|I - A\|_2) \max_{i=1,\ldots,j+1} \|e_i\|_2}{1 - \|A\|_2},$$

which yields the following bound on the spectral norm of E

$$\|E\|_2 \leq \frac{\sqrt{k}\,(1 + \|I - A\|_2) \max_{i=1,\ldots,k} \|e_i\|_2}{1 - \|A\|_2}. \tag{14}$$

Running the algorithm described in (12), we thus observe the iterates $\tilde{x}_i$ and build $\tilde{U}$ from their differences. As in (AMPE) we then compute $\tilde{c}$ using matrix $\tilde{U}$ and finally estimate

$$\tilde{x}^* = \sum_{i=0}^{k} \tilde{c}_j \tilde{x}_j.$$

In this case, our estimate for x^* is based on the coefficient $\tilde{c}$, computed using the iterates $\tilde{x}_i$. The error induced in this estimate by the nonlinearity can be decomposed as

$$\left\| \sum_{i=0}^{k} \tilde{c}_i \tilde{x}_i - x^* \right\|_2 \leq \left\| \sum_{i=0}^{k} (\tilde{c}_i - c_i) x_i \right\|_2 + \left\| \sum_{i=0}^{k} \tilde{c}_i (\tilde{x}_i - x_i) \right\|_2 + \left\| \sum_{i=0}^{k} c_i x_i - x^* \right\|_2. \tag{15}$$

and we begin by showing the following proposition computing the perturbation $\Delta c = \tilde{c}^* - c^*$ of the optimal solution c^* induced by $E = \tilde{U} - U$, which will allow us to bound the first term on the right-hand side of (15).

Proposition 2.3. *Let c^* be the optimal solution to* (AMPE)

$$c^* = \underset{\mathbf{1}^T c = 1}{\operatorname{argmin}} \ \|Uc\|_2$$

for some matrix $U \in \mathbb{R}^{n,k}$. Suppose U becomes $\tilde{U} = U + E$, let $M = \tilde{U}^T \tilde{U}$ and write the perturbation matrix $P = \tilde{U}^T \tilde{U} - U^T U$, with $c^* + \Delta c$ the perturbed solution to (AMPE)*, then*

$$\Delta c = -\left(I - \frac{M^{-1} \mathbf{1} \mathbf{1}^T}{\mathbf{1}^T M^{-1} \mathbf{1}} \right) M^{-1} P c^* \tag{16}$$

where

$$\left(I - \frac{M^{-1} \mathbf{1} \mathbf{1}^T}{\mathbf{1}^T M^{-1} \mathbf{1}} \right)$$

is a projector of rank $k - 1$.

Proof. Let μ be the dual variable corresponding to the equality constraint. Both $c^* + \Delta c$ and $\mu^* + \Delta \mu$ must satisfy the KKT system

$$\begin{bmatrix} 2M & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{pmatrix} c^* + \Delta c \\ \mu^* + \Delta \mu \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

writing $P = U^T E + E^T U + E^T E$, this means again

$$\begin{aligned} \begin{bmatrix} 2M & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{pmatrix} c^* + \Delta c \\ \mu^* + \Delta \mu \end{pmatrix} &= \begin{bmatrix} 2P & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} c^* + \Delta c \\ \mu^* + \Delta \mu \end{pmatrix} + \begin{bmatrix} 2U^T U & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{pmatrix} c^* + \Delta c \\ \mu^* + \Delta \mu \end{pmatrix} \\ &= \begin{pmatrix} 2P(c^* + \Delta c) \\ 0 \end{pmatrix} + \begin{bmatrix} 2U^T U & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{pmatrix} \Delta c \\ \Delta \mu \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{aligned}$$

hence

$$\begin{bmatrix} 2M & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{pmatrix} \Delta c \\ \Delta \mu \end{pmatrix} = \begin{pmatrix} -2Pc^* \\ 0 \end{pmatrix}$$

The block matrix can be inverted explicitly, with

$$\begin{bmatrix} 2M & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}^{-1} = \frac{1}{\mathbf{1}^T M^{-1} \mathbf{1}} \begin{bmatrix} \frac{1}{2} M^{-1} \left((\mathbf{1}^T M^{-1} \mathbf{1}) I - \mathbf{1} \mathbf{1}^T M^{-1} \right) & M^{-1} \mathbf{1} \\ \mathbf{1}^T M^{-1} & -2 \end{bmatrix}$$

leading to an expression of Δc and $\Delta \mu$ in terms of c^* and μ^*:

$$\begin{pmatrix} \Delta c \\ \Delta \mu \end{pmatrix} = \frac{1}{\mathbf{1}^T M^{-1} \mathbf{1}} \begin{bmatrix} \frac{1}{2} M^{-1} \left((\mathbf{1}^T M^{-1} \mathbf{1}) I - \mathbf{1} \mathbf{1}^T M^{-1} \right) & M^{-1} \mathbf{1} \\ \mathbf{1}^T M^{-1} & -2 \end{bmatrix} \begin{pmatrix} -2Pc^* \\ 0 \end{pmatrix}$$

After some simplification, we get

$$\Delta c = -\left(I - \frac{M^{-1}\mathbf{1}\mathbf{1}^T}{\mathbf{1}^T M^{-1}\mathbf{1}}\right) M^{-1}Pc^* = -WM^{-1}Pc^*$$

where W is a projector of rank $k-1$, which is the desired result. ■

We see here that this perturbation can be arbitrarily large, which is the key issue with the convergence results in [Smith et al., 1987, §7]. Even if $\|c^*\|_2$ and $\|P\|_2$ are small, M^{-1} is usually ill-conditioned. In fact, it can be shown that U^TU, which is the square of a Krylov matrix, has a condition number typically growing exponentially with the dimension [Tyrtyshnikov, 1994]. Moreover, the eigenvalues are perturbed by P which can make the situation even worse.

3. Regularized Minimal Polynomial Extrapolation

In the section that follows, we show how to regularize (AMPE) to solve the stability issues detailed above and better control the error term in (16).

3.1. Regularized AMPE.

The condition number of the matrix U^TU in problem (AMPE) can be arbitrary large which, together with nonlinear errors, can lead to highly unstable solutions c^*. We thus study a regularized formulation of problem (AMPE), which reads

$$\begin{array}{ll} \text{minimize} & c^T(U^TU + \lambda I)c \\ \text{subject to} & \mathbf{1}^Tc = 1 \end{array} \tag{RMPE}$$

The solution of this problem may be computed by solving a linear system, and the regularization parameter controls the norm of the solution, as shown in the following lemma which will allow us to bound the second term on the right-hand side of (15).

Lemma 3.1. *Let c^*_λ be the optimal solution of problem (RMPE) with $\lambda > 0$. Then*

$$c^*_\lambda = \frac{(U^TU + \lambda I)^{-1}\mathbf{1}}{\mathbf{1}^T(U^TU + \lambda I)^{-1}\mathbf{1}} \tag{17}$$

Therefore

$$\|c^*_\lambda\|_2 \leq \sqrt{\frac{\lambda + \|U\|_2^2}{k\lambda}}. \tag{18}$$

Proof. Let c^*_λ the optimal solution of the primal and ν^*_λ the optimal dual variable of problem (RMPE). Let $M_\lambda = U^TU + \lambda I$. Then both c^*_λ and ν^*_λ must satisfy the KKT system

$$\begin{bmatrix} 2M_\lambda & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{pmatrix} c^*_\lambda \\ \mu^*_\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

hence

$$\begin{pmatrix} c^*_\lambda \\ \mu^*_\lambda \end{pmatrix} = \begin{bmatrix} 2M_\lambda & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

This block matrix can be inverted explicitly, with

$$\begin{bmatrix} 2M_\lambda & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}^{-1} = \frac{1}{\mathbf{1}^TM_\lambda^{-1}\mathbf{1}} \begin{bmatrix} \frac{1}{2}M_\lambda^{-1}((\mathbf{1}^TM_\lambda^{-1}\mathbf{1})I - \mathbf{1}\mathbf{1}^TM_\lambda^{-1}) & M_\lambda^{-1}\mathbf{1} \\ \mathbf{1}^TM_\lambda^{-1} & -2 \end{bmatrix},$$

leading to

$$c^*_\lambda = \frac{M_\lambda^{-1}\mathbf{1}}{\mathbf{1}^TM_\lambda^{-1}\mathbf{1}}.$$

Since

$$\|M_\lambda^{-1}\|_2 \leq \frac{1}{\sigma_{\min}(U^TU)+\lambda} \leq \frac{1}{\lambda}$$

and

$$\mathbf{1}^T M_\lambda^{-1}\mathbf{1} \geq \frac{\|\mathbf{1}\|^2}{\sigma_{\max}(M_\lambda)} \geq \frac{k}{\|U\|_2^2+\lambda}$$

we obtain

$$\|c_\lambda^*\|_2 = \frac{\|M_\lambda^{-1/2}M_\lambda^{-1/2}\mathbf{1}\|_2}{\mathbf{1}^T M_\lambda^{-1}\mathbf{1}} \leq \frac{\|M_\lambda^{-1}\|_2^{1/2}}{\sqrt{\mathbf{1}^T M_\lambda^{-1}\mathbf{1}}} \leq \sqrt{\frac{\lambda+\|U\|_2^2}{k\lambda}}$$

which is the desired result. ■

This allows us to obtain the following immediate corollary extending Proposition 2.3 to the regularized AMPE problem in (RMPE) where λ now explicitly controls the perturbation of c.

Corollary 3.2. *Let c_λ^*, defined in (17), be the solution of problem (RMPE). Then the solution of problem (RMPE) for the perturbed matrix $\tilde{U} = U + E$ is given by $c_\lambda^* + \Delta c_\lambda$ where*

$$\Delta c_\lambda = -WM_\lambda^{-1}Pc_\lambda^* = -M_\lambda^{-1}W^TPc_\lambda^* \qquad \text{and} \qquad \|\Delta c_\lambda^*\|_2 \leq \frac{\|P\|_2}{\lambda}\|c_\lambda^*\|_2$$

where $M_\lambda = (U^TU + P + \lambda I)$ and $W = \left(I - \frac{M_\lambda^{-1}\mathbf{1}\mathbf{1}^T}{\mathbf{1}^TM_\lambda^{-1}\mathbf{1}}\right)$ is a projector of rank $k-1$.

These results lead us to the Regularized Approximate Minimal Polynomial Extrapolation method described as Algorithm 1. Its computational complexity (with online updates or in batch mode) is $O(nk^2)$ and explicit numerical procedures (batch and online) are discussed in Section 3.5 below. Note that the algorithm never calls the oracle $g(x)$ which means that, in an optimization context, this acceleration method does not require access to either $f(x)$ or $\nabla f(x)$ to compute the extrapolation. Moreover, unlike classical accelerated techniques, it does not rely on a priori information on the function, e.g. access to the smoothness parameter L and or the strong convexity parameter μ as in the algorithm in [Nesterov, 1983].

Algorithm 1 Regularized Approximate Minimal Polynomial Extrapolation (RMPE)

Input: Sequence $\{x_0, x_1, ..., x_{k+1}\}$, parameter $\lambda > 0$
Compute $U = [x_1 - x_0, ..., x_{k+1} - x_k]$
Solve the linear system $(U^TU + \lambda I)z = \mathbf{1}$
Set $c = z/(z^T\mathbf{1})$
Return $\sum_{i=0}^k c_i x_i$
Output: Approximation of the fixed point x^*

3.2. **Convergence Bounds on Regularized AMPE.** To fully characterize convergence of our estimate sequence, we still need to bound the last term on the right-hand side of (15), namely $\|\sum_{i=0}^k c_i x_i - x^*\|_2$. A coarse bound can be provided using Chebyshev polynomials, however the norm of the Chebyshev coefficients grows exponentially with k, which one of the root causes of instability in the classical Eddy-Mešina algorithm [Mešina, 1977; Eddy, 1979]. Here, we refine this bound to improve our estimate of acceleration performance.

Consider the following Chebyshev-like optimization problem, written

$$S(k,\alpha) \triangleq \min_{\{q\in\mathbb{R}_k[x]:\, q(1)=1\}} \left\{ \max_{x\in[0,\sigma]} \left((1-x)q(x)\right)^2 + \alpha\|q\|_2^2 \right\}, \tag{19}$$

where $\mathbb{R}_k[x]$ is the set of polynomials of degree at most k and $q \in \mathbb{R}^{k+1}$ is the vector of coefficients of the polynomial $q(x)$. This problem can be solved exactly as it can be reduced to a semidefinite program, which will be detailed explicitly in Section 3.6. We now show our main result which describes how $S(k, \alpha)$ bounds the error between our estimate of the optimum constructed using the solution of (RMPE) on the iterates $\tilde{x}_i$, and the optimum x^* of problem (1).

Proposition 3.3. *Let matrices $X = [x_0, x_1, ..., x_k]$, $\tilde{X} = [x_0, \tilde{x}_1, ..., \tilde{x}_k]$, $\mathcal{E} = (X - \tilde{X})$ and scalar $\kappa = \|(A - I)^{-1}\|_2$. Suppose $\tilde{c}_\lambda^*$ solves problem (RMPE)*

$$\begin{array}{ll} \textit{minimize} & c^T(\tilde{U}^T\tilde{U} + \lambda I)c \\ \textit{subject to} & \mathbf{1}^T c = 1 \end{array}$$

in the variable $c \in \mathbb{R}^{k+1}$, with parameters $\tilde{U} \in \mathbb{R}^{n\times(k+1)}$, then

$$\tilde{c}_\lambda^* = \frac{(\tilde{U}^T\tilde{U} + \lambda I)^{-1}\mathbf{1}}{\mathbf{1}^T(\tilde{U}^T\tilde{U} + \lambda I)^{-1}\mathbf{1}} \tag{20}$$

Assume $A = g'(x^*)$ symmetric with $0 \preceq A \preceq \sigma I$ where $\sigma < 1$. Then

$$\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2 \leq \left(\kappa^2 + \frac{1}{\lambda}\left(1 + \frac{\|P\|_2}{\lambda}\right)^2\left(\|\mathcal{E}\|_2 + \kappa\frac{\|P\|_2}{2\sqrt{\lambda}}\right)^2\right)^{\frac{1}{2}}\left(S(k, \lambda/\|x_0 - x^*\|_2^2)\right)^{\frac{1}{2}}\|x_0 - x^*\|_2$$

with the perturbation matrix $P = \tilde{U}^T\tilde{U} - U^TU$, and $S(k, \alpha)$ is defined in (19) above.

Proof. Writing the error decomposition (15) in matrix format, we get

$$\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2 \leq \|Xc_\lambda^* - x^*\|_2 + \|(X - X^*)\Delta c\|_2 + \|\mathcal{E}\tilde{c}_\lambda^*\|_2.$$

The first term can be bounded as follows

$$\begin{aligned} \|Xc_\lambda^* - x^*\|_2 &\leq \kappa\|Uc_\lambda^*\|_2 \\ &\leq \kappa\sqrt{\|Uc_\lambda^*\|_2^2 + (\lambda - \lambda)\|c_\lambda^*\|_2^2} \\ &\leq \kappa\sqrt{\|(A - I)p(A)\|_2^2\|x_0 - x^*\|_2^2 + \lambda\|c_\lambda^*\|_2^2 - \lambda\|c_\lambda^*\|_2^2} \\ &\leq \kappa\sqrt{S(k, \lambda/\|x_0 - x^*\|_2^2)\|x_0 - x^*\|_2^2 - \lambda\|c_\lambda^*\|_2^2}. \end{aligned}$$

The second one becomes, using Corollary 3.2,

$$\begin{aligned} \|(X - X^*)\Delta c_\lambda^*\|_2 &\leq \kappa\|U\Delta c_\lambda^*\|_2 \\ &\leq \kappa\|U(U^TU + \lambda I + P)^{-1}\tilde{W}^TP\|_2\|c_\lambda^*\|_2 \\ &\leq \kappa\|U(U^TU + \lambda I + P)^{-1}\|_2\|P\|_2\|c_\lambda^*\|_2. \end{aligned}$$

Let us write $(U^TU + \lambda I + P)^{-1} = [(U^TU + \lambda I)^{-1} + S]$ for some perturbation S. Indeed,

$$((U^TU + \lambda I)^{-1} + S)(U^TU + \lambda I + P) = I,$$

leads to

$$S = -(U^TU + \lambda I)^{-1}P(U^TU + \lambda I + P)^{-1}.$$

Plugging this expression in $\|U(U^TU + \lambda I + P)^{-1}\|_2$ we obtain

$$\begin{aligned} \|U(U^TU + \lambda I + P)^{-1}\|_2 &= \|U(U^TU + \lambda I)^{-1}(I - P(U^TU + \lambda I + P)^{-1})\|_2 \\ &\leq \|U(U^TU + \lambda I)^{-1}\|_2\left(1 + \|P\|_2\|(U^TU + \lambda I + P)^{-1}\|_2\right) \\ &\leq \frac{\sigma}{\sigma^2 + \lambda}\left(1 + \frac{\|P\|_2}{\lambda}\right). \end{aligned}$$

For some value of $\sigma \in [\sigma_{\min}^{1/2}(U^TU), \sigma_{\max}^{1/2}(U^TU)]$. The maximum is attained at $\sigma = \sqrt{\lambda}$, so it becomes

$$\|U(U^TU + \lambda I + P)^{-1}\|_2 \leq \frac{1}{2\sqrt{\lambda}}\left(1 + \frac{\|P\|_2}{\lambda}\right).$$

So the second term can be bounded by

$$\|(X - X^*)\Delta c_\lambda^*\|_2 \leq \kappa\frac{\|P\|_2}{2\sqrt{\lambda}}\left(1 + \frac{\|P\|_2}{\lambda}\right)\|c_\lambda^*\|.$$

The third term can be bounded as follows

$$\begin{aligned}\|\mathcal{E}\tilde{c}_\lambda^*\|_2 &\leq \|\mathcal{E}\|_2(\|c_\lambda^*\|_2 + \|\Delta c_\lambda^*\|_2)\\ &\leq \|\mathcal{E}\|_2\left(1 + \frac{\|P\|_2}{\lambda}\right)\|c_\lambda^*\|_2.\end{aligned}$$

If we combine all bounds, we obtain

$$\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2^2 \leq \kappa\sqrt{S(k, \lambda/\|x_0 - x^*\|_2^2)\|x_0 - x^*\|_2^2 - \lambda\|c_\lambda^*\|_2^2} + \|c_\lambda^*\|_2\left(1 + \frac{\|P\|_2}{\lambda}\right)\left(\|\mathcal{E}\|_2 + \kappa\frac{\|P\|_2}{2\sqrt{\lambda}}\right).$$

To make this bound uniform in $\|c_\lambda^*\|_2$, we maximize it according to this term. For simplicity, let us write it using parameters a, b and $c = \|c_\lambda^*\|_2$, to get

$$\kappa\sqrt{a^2 - \lambda c^2} + bc.$$

We want to solve

$$\max_{0 \leq c \leq (a/\sqrt{\lambda})} \kappa\sqrt{a^2 - \lambda c^2} + bc,$$

in the variable c. The solution is given by

$$c = \frac{a}{\sqrt{\lambda}}\frac{b}{\sqrt{\kappa^2\lambda + b^2}} \in \left[0, \frac{a}{\sqrt{\lambda}}\right].$$

and the optimal value becomes

$$\max_{0 \leq c \leq (a/\sqrt{\lambda})} \kappa\sqrt{a^2 - \lambda c^2} + bc = \frac{a}{\sqrt{\lambda}}\sqrt{\kappa^2\lambda + b^2}.$$

Replacing a, b by their actual values, we have

$$\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2 \leq \sqrt{S(k, \lambda/\|x_0 - x^*\|_2^2)}\|x_0 - x^*\|_2\sqrt{\kappa^2 + \frac{1}{\lambda}\left(1 + \frac{\|P\|_2}{\lambda}\right)^2\left(\|\mathcal{E}\|_2 + \kappa\frac{\|P\|_2}{2\sqrt{\lambda}}\right)^2},$$

which is the desired result. ■

3.3. **Explicit Bounds for the Gradient Method.** The non-asymptotic bound in Proposition 3.3 can be heavily simplified when k is large enough and we are using the simple gradient method on smooth, strongly convex functions with Lipschitz-continuous Hessian. In this scenario, the fixed point iteration becomes

$$\tilde{x}_{k+1} = \tilde{x}_k - \frac{1}{L}\nabla f(\tilde{x}_k)$$

where $\mu I \preceq \nabla^2 f(x) \preceq LI$. Also, for simplicity, we assume $\lambda_{\min}(\nabla^2 f(x^*)) = \mu$ and $\lambda_{\max}(\nabla^2 f(x^*)) = L$. This restriction can easily be lifted and produces much simpler expressions. The general case where $\lambda_{\min}(\nabla^2 f(x^*)) \geq \mu$ and/or $\lambda_{\max}(\nabla^2 f(x^*)) \leq L$ leads to tighter bounds but does not change the general conclusion. We will also assume the Lipschitz-continuity of the Hessian, i.e.

$$\|\nabla^2 f(y) - \nabla^2 f(x)\|_2 \leq M\|y - x\|_2$$

Using these assumptions, the rate of convergence of the gradient method [Nesterov, 2003] is

$$\|\tilde{x}_k - x^*\|_2 \leq \left(\sqrt{\frac{L-\mu}{L+\mu}}\right)^k \|x_0 - x^*\|_2 = r^k \|x_0 - x^*\|_2$$

Note that $(1/L)$ is not the optimal fixed-step gradient method, but this version simplifies the analysis. The asymptotic linear model is here

$$\begin{aligned}
\tilde{x}_{k+1} &= \tilde{x}_k - \frac{1}{L}\nabla f(\tilde{x}_k) \\
&= \tilde{x}_k - \frac{1}{L}\left(\nabla f(x^*) + \nabla^2 f(x^*)(\tilde{x}_k - x^*) + O(\|\tilde{x}_k - x^*\|_2^2)\right) \\
&= \tilde{x}_k - \frac{1}{L}\nabla^2 f(x^*)(\tilde{x}_k - x^*) + O(\|\tilde{x}_k - x^*\|_2^2) \\
\tilde{x}_{k+1} - x^* &= A(\tilde{x}_k - x^*) + O(\|\tilde{x}_k - x^*\|_2^2),
\end{aligned}$$

where $A = I - \frac{1}{L}\nabla^2 f(x^*)$, meaning that $\|A\|_2 \leq 1 - \frac{\mu}{L}$. The asymptotic model is thus written

$$x_{k+1} = x^* + A(x_k - x^*)$$

Using this recursion (see Appendix A for full details), we can bound $\|\tilde{X} - X^*\|_2$, $\|U\|_2$, $\|\mathcal{E}\|_2$ and $\|E\|_2$ as follows

$$\begin{aligned}
\|\tilde{X} - X^*\|_2 &\leq \frac{1-r^k}{1-r}\|x_0 - x^*\|_2 \\
\|U\|_2 &\leq \frac{L}{\mu}\left(1 - \left(1 - \frac{\mu}{L}\right)^k\right)\|x_0 - x^*\|_2 \\
\|\mathcal{E}\|_2 &\leq \left(1 + \frac{L}{\mu}\right)^2 \frac{M}{2L}\left(\frac{1}{2} - \left(1 - \frac{\mu}{L}\right)^k + \frac{1}{2}\left(\frac{1 - \frac{\mu}{L}}{1 + \frac{\mu}{L}}\right)^k\right)\|x_0 - x^*\|_2^2 \\
\|E\|_2 &\leq 2\|\mathcal{E}\|_2
\end{aligned}$$

Plugging these quantities in the bounds of Proposition 3.3 allows us to explicitly characterize acceleration performance on the gradient method. Setting for instance $L = 100$, $\mu = 10$, $M = 10^{-1}$, $\|x_0 - x^*\|_2 = 10^{-4}$ and finally $\lambda = \|P\|_2$. In Figure 1 (*Left*) we plot the relative value for λ (i.e. $\|P\|_2/\|x_0 - x^*\|^2$) used in regularizing (20), while the speedup factor forecast by the bound in Proposition 3.3 compared to the gradient method is plotted in Figure 1 (*Right*). While the speedup implied in the bound is highly conservative, it remains significantly larger than one for a range of values of k.

3.4. **Asymptotic behavior.** We will now analyze the asymptotic behavior of the bound in Proposition 3.3, assuming

$$\|\mathcal{E}\|_2 = O(\|x_0 - x^*\|_2^2), \qquad \|U\|_2 = O(\|x_0 - x^*\|_2) \quad \Rightarrow \quad \|P\|_2 = O(\|x_0 - x^*\|_2^3),$$

which holds for example when minimizing a smooth strongly convex function with Lipschitz-continuous Hessian using fixed-step gradient method (as discussed in Section 3.3 above). We will now show that when $\|x_0 - x^*\|_2$ is close to zero, we recover the rate of Chebyshev acceleration up to a constant depending of the regularization. So, according to Proposition (2.2), when $\|x_0 - x^*\|_2$ is close to zero, the regularized version of AMPE tends to converge as fast as AMPE up to a small constant. These results leads us to the following corollary.

Corollary 3.4. *Assume we used the gradient method with stepsize in* $]0, \frac{2}{L}[$ *on a* L*-smooth* μ*-strongly convex function* f *with Lipschitz-continuous Hessian of constant* M*. Then the asymptotic convergence of RMPE*

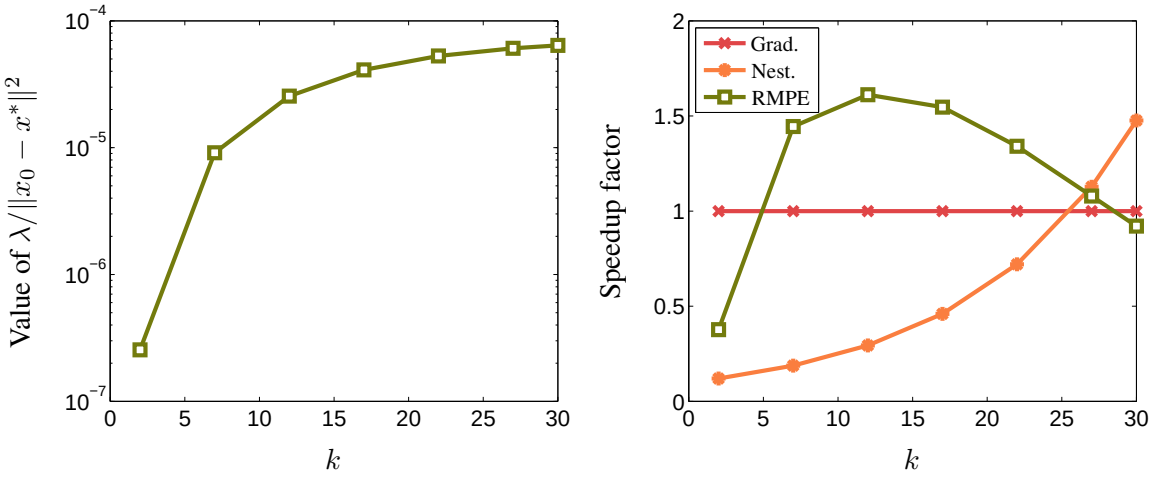

FIGURE 1. *Left:* Relative value for the regularization parameter λ used in the theoretical bound. *Right:* Convergence speedup relative to the gradient method, for Nesterov's accelerated method and the theoretical RMPE bound in Proposition 3.3. We see that our (highly conservative) bound shows a significant speedup when k is well chosen.

algorithm (i.e. when x_0 is close to x^*), with parameter k and $\lambda = \beta\|P\|_2$, is controlled by

$$\|\tilde{X}\tilde{c}^*_\lambda - x^*\|_2 \leq \left(1 + \frac{(1+\frac{1}{\beta})^2}{4\beta^2}\right)^{1/2} \kappa\, T_k(\sigma)\|x_0 - x^*\|$$

where $T(\sigma)$, defined in (11), satisfies

$$|T_k(\sigma)| = \frac{2\zeta^k}{1+\zeta^{2k}}$$

in the Chebyshev acceleration bound, and $\sigma = 1 - \frac{\mu}{L}$ is the rate of convergence of the asymptotic model of the gradient descend and $\kappa = \frac{L}{\mu}$ is the condition number of the function $f(x)$.

Proof. Let $\|x_0 - x^*\|_2 \to 0$ (i.e. we start closer and closer to the optimal point) and assume $\lambda = \beta\|P\|_2$ for $\beta > 0$. In this case, we can approximate the term

$$\begin{aligned}
&\lim_{\|x_0-x^*\|_2\to 0} \sqrt{\kappa^2 + \frac{1}{\lambda}\left(1+\frac{\|P\|_2}{\lambda}\right)^2 \left(\|\mathcal{E}\|_2 + \kappa\frac{\|P\|_2}{2\sqrt{\lambda}}\right)^2} \\
= &\lim_{\|x_0-x^*\|_2\to 0} \sqrt{\kappa^2 + \frac{1}{\beta\|P\|_2}\left(1+\frac{1}{\beta}\right)^2 \left(\|\mathcal{E}\|_2 + \kappa\frac{\sqrt{\|P\|_2}}{2\sqrt{\beta}}\right)^2} \\
= &\lim_{\|x_0-x^*\|_2\to 0} \sqrt{\kappa^2 + \left(1+\frac{1}{\beta}\right)^2 \left(\frac{\|\mathcal{E}\|_2}{\beta\sqrt{\|P\|_2}} + \kappa\frac{1}{2\beta}\right)^2}
\end{aligned}$$

Since $\|\mathcal{E}\| = O(\|x - x^*\|_2^2)$ and $\sqrt{\|P\|_2} = O(\|x - x^*\|_2^{\frac{3}{2}})$, the limit becomes

$$\sqrt{\kappa^2 + \left(1+\frac{1}{\beta}\right)^2 \left(\frac{\kappa}{2\beta}\right)^2} = \kappa\left(1 + \frac{(1+\frac{1}{\beta})^2}{4\beta^2}\right)^{1/2}.$$

Moreover, we have $\frac{\lambda}{\|x_0-x^*\|_2^2} \to 0$ since

$$\lim_{\|x_0-x^*\|_2\to 0} \frac{\lambda}{\|x_0-x^*\|_2^2} = \lim_{\|x_0-x^*\|_2\to 0} \frac{\|P\|_2}{\beta\|x_0-x^*\|_2^2} = \lim_{\|x_0-x^*\|_2\to 0} \frac{O(\|x_0-x^*\|^3)}{O(\|x_0-x^*\|_2^2)} = 0,$$

so the asymptotic relative rate of convergence of the extrapolation method becomes

$$\lim_{\|x_0-x^*\|_2\to 0} \frac{\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2}{\|x_0-x^*\|_2} \;\leq\; \kappa\left(1+\frac{(1+\frac{1}{\beta})^2}{4\beta^2}\right)^{1/2}\sqrt{S\,(k,0)}$$

We can compare the values $\sqrt{S\,(k,0)}$ and $T_k(\sigma)$ in (11), with

$$\begin{aligned}
\sqrt{S\,(k,0)} &= \sqrt{\min_{\{q\in\mathbb{R}_k[x]:\, q(1)=1\}}\left\{\max_{x\in[0,\sigma]}\;((1-x)q(x))^2\right\}} \\
&\leq \sqrt{\left\{\max_{x\in[0,\sigma]}\;((1-x)T(x))^2\right\}} \\
&\leq \sqrt{\left\{\max_{x\in[0,\sigma]}\;(T(x))^2\right\}} \;=\; T(\sigma).
\end{aligned}$$

Finally,

$$\lim_{\|x_0-x^*\|_2\to 0} \frac{\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2}{\|x_0-x^*\|_2} \;\leq\; \left(1+\frac{(1+\frac{1}{\beta})^2}{4\beta^2}\right)^{1/2}\kappa\, T(\sigma)$$

which is the desired result. ■

3.5. **Computational Complexity of the RMPE Algorithm.** In Algorithm 1, computing the coefficients $\tilde{c}_\lambda^*$ means solving the $k\times k$ system $(\tilde{U}^T\tilde{U}+\lambda I)z = \mathbf{1}$. We then get $\tilde{c}_\lambda^* = z/(\mathbf{1}^T z)$. This can be done in both batch and online mode.

3.5.1. *Online updates.* Here, we receive the vectors u_i one by one from the optimization algorithm. In this case, we perform low-rank updates on the Cholesky factorization of the system matrix. At iteration i, we have the Cholesky factorization $LL^T = \tilde{U}^T\tilde{U}+\lambda I$. We receive a new vector u_+ and we want

$$L_+L_+^T = \begin{bmatrix} L & 0 \\ a^T & b\end{bmatrix}\begin{bmatrix} L^T & a \\ 0 & b\end{bmatrix} = \begin{bmatrix} \tilde{U}^T\tilde{U}+\lambda I & \tilde{U}^Tu_+ \\ (\tilde{U}^Tu_+)^T & u_+^Tu_+ + \lambda\end{bmatrix}.$$

We can explicitly solve this system in variables a and b, and the solutions are

$$a = L^{-1}\tilde{U}^Tu_+, \quad b = a^Ta+\lambda.$$

The complexity of this update is thus $O(in+i^2)$, i.e. the matrix-vector multiplication of $\tilde{U}^Tu_+$ and solving the triangular system. Since we need to do it k times, the final complexity is thus $O(nk^2+k^3)$. Notice that, at the end, it takes only $O(k^2)$ iteration to solve the system $LL^Tz = \mathbf{1}$.

3.5.2. *Batch mode.* The complexity is divided in two parts: First, we need to build the linear system itself. Since $U\in\mathbb{R}^{n\times k}$, it takes $O(nk^2)$ flops to perform the multiplication. Then we need to solve the linear system $(\tilde{U}^T\tilde{U}+\lambda I)z = \mathbf{1}$ which can be done by a direct solver like Gaussian elimination (if k is small) or Cholesky factorization, or using an iterative method like conjugate gradient method. It takes $O(k^3)$ flops to solve the linear system in the worst case, meaning that the complexity at the end is $O(nk^2+k^3)$. In practice, the eigenvalues of the system tend to be clustered around λ, which means that the conjugate gradient solver converges very quickly to a good solution.

3.6. **Regularized Chebyshev Polynomials.** We first briefly recall basic results on Sum of Squares (SOS) polynomials and moment problems [Nesterov, 2000; Lasserre, 2001; Parrilo, 2000], which will allow us to formulate problem (19) as a (tractable) semidefinite program. A univariate polynomial is positive if and only if it is a sum of squares. Furthermore, if we let $m(x) = (1, x, \ldots, x^k)^T$ we have, for any $p(x) \in \mathbb{R}_{2k}[x]$,

$$\begin{array}{c} p(x) \geq 0, \text{ for all } x \in \mathbb{R} \\ \Updownarrow \\ p(x) = m(x)^T C m(x), \text{ for some } C \succeq 0, \end{array}$$

which means that checking if a polynomial is positive on the real line is equivalent to solving a linear matrix inequality (see e.g. [Ben-Tal and Nemirovski, 2001, §4.2] for details). We can thus write the problem of computing the maximum of a polynomial over the real line as

$$\begin{array}{lll} \text{minimize} & t & \\ \text{subject to} & t - p(x) = m(x)^T C m(x), & \text{for all } x \in \mathbb{R} \\ & C \succeq 0, & \end{array} \tag{21}$$

which is a semidefinite program in the variables $p \in \mathbb{R}^{k+1}$, $C \in \mathbf{S}_{k+1}$ and $t \in \mathbb{R}$, because the first contraint is equivalent to a set of linear equality constraints. Then, showing that $p(x) \geq 0$ on the segment $[0, \sigma]$ is equivalent to showing that the rational fraction

$$p\left(\frac{\sigma x^2}{1+x^2}\right)$$

is positive on the real line, or equivalently, that the polynomial

$$(1+x^2)^k \, p\left(\frac{\sigma x^2}{1+x^2}\right)$$

is positive on the real line. Overall, this implies that problem (19) can be written

$$\begin{array}{lll} S(k,\alpha) = & \min. & t^2 + \alpha^2 \|q\|_2^2 \\ & \text{s.t.} & (1+x^2)^{k+1} \left(\left(1 - \frac{\sigma x^2}{1+x^2}\right) q\left(\frac{\sigma x^2}{1+x^2}\right)\right) = m(x)^T C m(x), \quad \text{for all } x \in \mathbb{R} \\ & & \mathbf{1}^T q = 1, \; C \succeq 0, \end{array} \tag{22}$$

which is a semidefinite program in the variables $q \in \mathbb{R}^{k+1}$, $C \in \mathbf{S}_{k+2}$ and $t \in \mathbb{R}$.

3.7. **Adaptive regularization.** In section 3.2 we have seen that λ controls the tradeoff between precision in the coefficient $\tilde{c}_\lambda^*$ and regularization. We can explicitly minimize the bound in Proposition (3.3), but this assumes a lot of prior knowledge about the function, and the resulting bound is very conservative. In practice, we can often pick a much smaller λ than the one which minimize the theoretical bound.

We use an adaptive algorithm in the parameter λ using a simple line-search technique between two parameters λ_0 and $\lambda_{\min}$ (the gap between the two value can be big). This is a simple dichotomy strategy: we start at some λ_0, and compute the associated c_λ^*. Then, we compute the coefficients for a smaller λ, say $\lambda/2$, and compare the two values $f(Xc_\lambda^*)$ and $f(Xc_{\lambda/2}^*)$. If the value increase when λ becomes smaller or if $\lambda < \lambda_{\min}$ we stop, otherwise we repeat the process. Even if an access to the oracle $f(x)$ is needed, we use it as a complete black-box. Moreover, we still do not need to access to the oracle $\nabla f(x)$.

3.8. **Smooth minimization.** We can extend our results to smooth functions that are not strongly convex using a simple regularization trick. Suppose we seek to solve

$$\min_{x \in \mathbb{R}^n} f(x)$$

in the variable $x \in \mathbb{R}^n$, where $f(x)$ has a Lipschitz continuous gradient with parameter L with respect to the Euclidean norm, but is not strongly convex. Assume for simplicity that the initial iterate x_0 is close enough

to the optimum so that $D \triangleq \|x_0 - x^*\| \geq \|x_k - x^*\|$ for any $k \geq 0$. We can approximate the above problem by

$$\min_{x \in \mathbb{R}^n} f_\varepsilon(x) \triangleq f(x) + \frac{\varepsilon}{2D^2}\|x\|_2^2 \tag{23}$$

in the variable $x \in \mathbb{R}^n$, where $f_\varepsilon(x)$ has a Lipschitz continuous gradient with parameter $L + \varepsilon/D^2$ with respect to the Euclidean norm, is strongly convex with parameter ε/D^2 with respect to the same norm. Furthermore $f_\varepsilon(x)$ is an ε approximation of $f(x)$ near the optimum and we get

$$\begin{aligned} f(x_k) - f(x^*) &= f_\varepsilon(x_k) - f_\varepsilon(x^*) + \frac{\varepsilon}{2D^2}(\|x_k\|_2^2 - \|x^*\|_2^2) \\ &\leq f_\varepsilon(x_k) - f_\varepsilon(x_\varepsilon^*) + \frac{\varepsilon}{2D^2}(\|x_k - x^*\|_2^2) \\ &\leq \frac{LD^2 + \epsilon}{2D^2}\|x_k - x_\varepsilon^*\|_2^2 + \frac{\varepsilon}{2D^2}(\|x_k - x^*\|_2^2) \end{aligned}$$

using the smoothness of $f_\varepsilon(x)$ and writing x_ε^* the optimum of problem (23). As in §3.3, linear convergence of gradient algorithms guarantees

$$\|x_k - x_\varepsilon^*\|_2 = O(r^k \|x_0 - x_\varepsilon^*\|_2)$$

which, combined with the previous inequality means

$$\begin{aligned} f(x_k) - f(x^*) &= O\left(\frac{LD^2 + \epsilon}{2D^2} r^{2k} \|x_0 - x_\varepsilon^*\|_2^2 + \frac{\varepsilon}{2D^2}(\|x_k - x^*\|_2^2)\right) \\ &= O\left(\frac{LD^2 + \epsilon}{2} r^k + \varepsilon\right) \end{aligned}$$

the number of iterations required to reach a target precision 2ϵ is thus bounded by

$$k = O\left(\frac{\log(LD^2/\varepsilon)}{\log(1/r)}\right).$$

When using a simple gradient method $r = \sqrt{1 - \varepsilon/(LD^2 + \varepsilon)}$, we have

$$\log(1/r) \sim 1 + \frac{LD^2}{\varepsilon}$$

while accelerated algorithms have $r = 1 - \sqrt{\varepsilon/(LD^2 + \varepsilon)}$ which yields

$$\log(1/r) \sim \sqrt{1 + \frac{LD^2}{\varepsilon}}.$$

Overall, this means that when the function $f(x)$ is not strongly convex, we can always approximate the minimization problem (1) by an equivalent strongly convex problem to which our acceleration analysis applies.

4. Numerical Experiments

In this section, we evaluate the performance of the RMPE acceleration method detailed in Algorithm 1, using the line-search strategy detailed in Section 3.7.

4.1. **Minimizing logistic regression.** We begin by testing our methods on a regularized logistic regression problem written

$$f(w) = \sum_{i=1}^{m} \log\left(1 + \exp(-y_i z_i^T w)\right) + \frac{\tau}{2}\|w\|_2^2,$$

where $Z = [z_1, ..., z_m]^T \in \mathbb{R}^{m\times n}$ is the design matrix and y is a $\{-1, 1\}^m$ vector of labels. The Lipchitz constant of the logistic regression is $L = \|Z\|_2^2/4 + \tau$ and the strong convexity parameter is $\mu = \tau$. We solve this problem using several algorithms:

- Fixed-step gradient method for smooth strongly convex functions [Nesterov, 2003, Th. 2.1.15] with iterations
$$x_{k+1} = x_k - \frac{2}{L+\mu}\nabla f(x_k)$$
- Accelerated gradient method for smooth strongly convex functions [Nesterov, 2003, Th. 2.2.3] with iterations
$$\begin{aligned} x_{k+1} &= y_k - \frac{1}{L}\nabla f(y_k) \\ y_{k+1} &= x_{k+1} + \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\left(x_{k+1} - x_k\right) \end{aligned}$$
- The adaptive RMPE algorithm, detailed in Section 3.7, with restart each k iteration.

The results are reported in Figure 2. Using very few iterates, the solution computed using our estimate (a nonlinear average of the gradient iterates) are markedly better than those produced by the accelerated method. This is only partially reflected by the theoretical bound from Proposition 3.3 which shows significant speedup in some regions but remains highly conservative (cf. Figure 1). Also, Figure 3 shows the impact of regularization: The (unregularized) AMPE process becomes unstable because of the condition number of matrix M, which significantly impacts the precision of the estimate.

5. Conclusion and Perspectives

In this paper, we developed a method which is able to accelerate, under some regularity conditions, the convergence of a sequence $\{x_i\}$ without any knowledge of the algorithm which generated this sequence. The regularization parameter used in the acceleration method can be computed easily using some inexact line-search. The algorithm itself is simple as it only requires solving a small linear system. Also, we showed (using gradient method on logistic regression) that the strategy which consists in restarting the algorithm after an extrapolation method can lead to significantly improved convergence rates. Future work will consist in improving the performance of the algorithm by exploiting the structure of the noise matrix E in some cases (for example, using gradient method, the norm of the column E_k in the matrix E is decreasing when k grows), extending the algorithm to the stochastic case and to the non-symmetric case, and to refine the term (19) present in the theoretical bound.

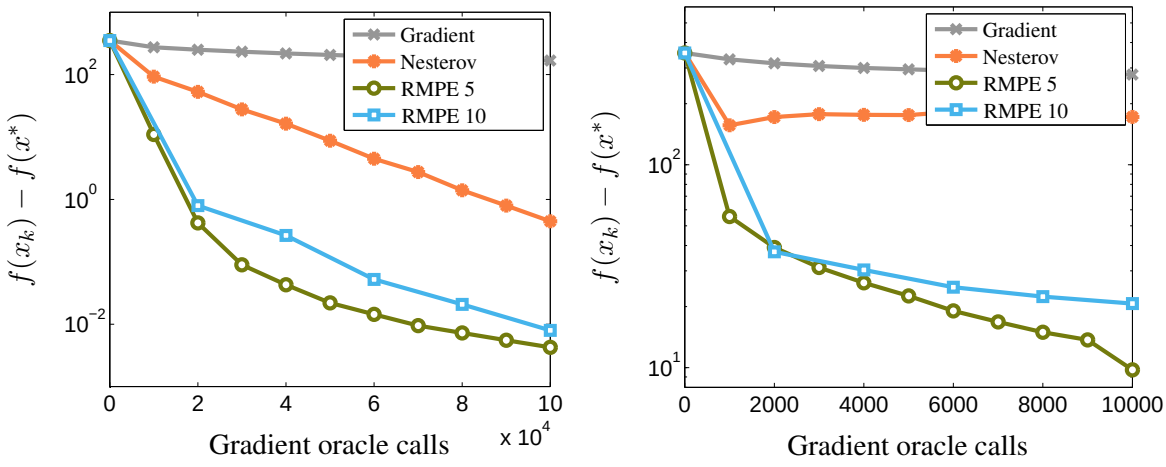

FIGURE 2. Solving logistic regression on *UCI Madelon dataset* (500 features, 2000 samples) using the gradient method, Nesterov's accelerated method and RMPE with $k = 5, 10$, with penalty parameter $\tau = 10^2$ so the condition number becomes 1.2×10^9 (*Left*), and $\tau = 10^{-3}$ in order to have a condition number equal to 1.2×10^{14} (*Right*). In this graph, we see that our algorithm has a similar behavior to the conjugate gradient: unlike Nesterov's method, where we need to provide information on the spectrum of the function (i.e. parameters μ and L), the RMPE algorithm adapts itself to the spectrum of $g(x^*)$ and exploits the good local strong convexity of the objective, without any prior information.

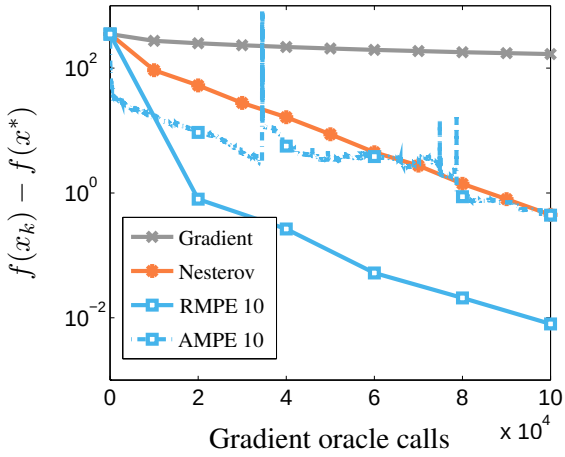

FIGURE 3. Logistic regression on *Madelon UCI Dataset*, solved using the gradient method, Nesterov's method and AMPE (i.e. RMPE with $\lambda = 0$). The condition number is equal to 1.2×10^9. We see that without regularization, AMPE is becomes unstable as $\|(\tilde{U}^T\tilde{U})^{-1}\|_2$ gets too large (cf. Proposition 2.3).

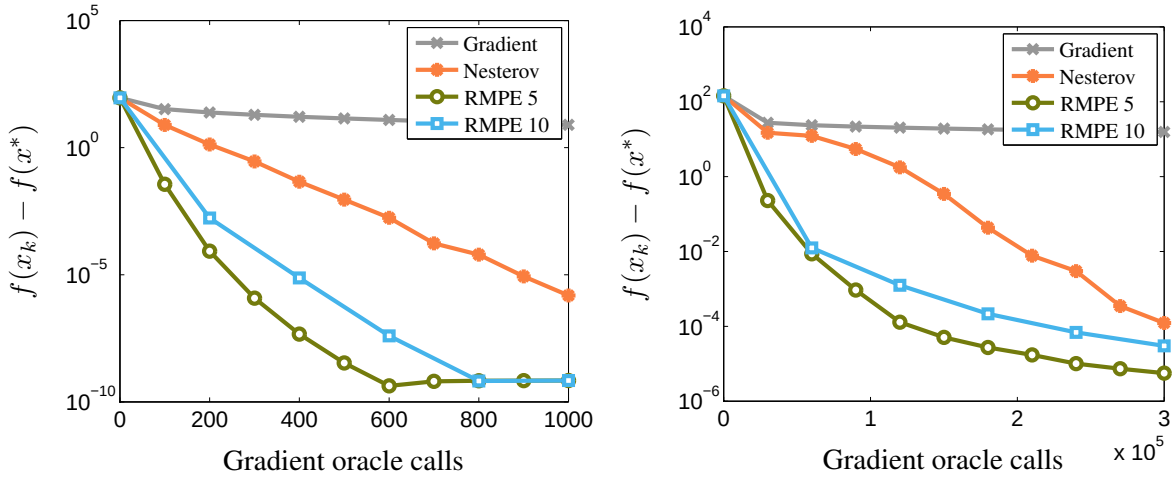

FIGURE 4. Logistic regression on *Sonar Scaled UCI Dataset* (60 features, 208 data points), solved using several algorithms. *Left:* the penalty parameter has been set to 10^{-1} in order to have a condition number equal to 1.4×10^4. *Right:* penalty parameter equal to 10^{-6}, so the condition number is equal to 1.4×10^9.

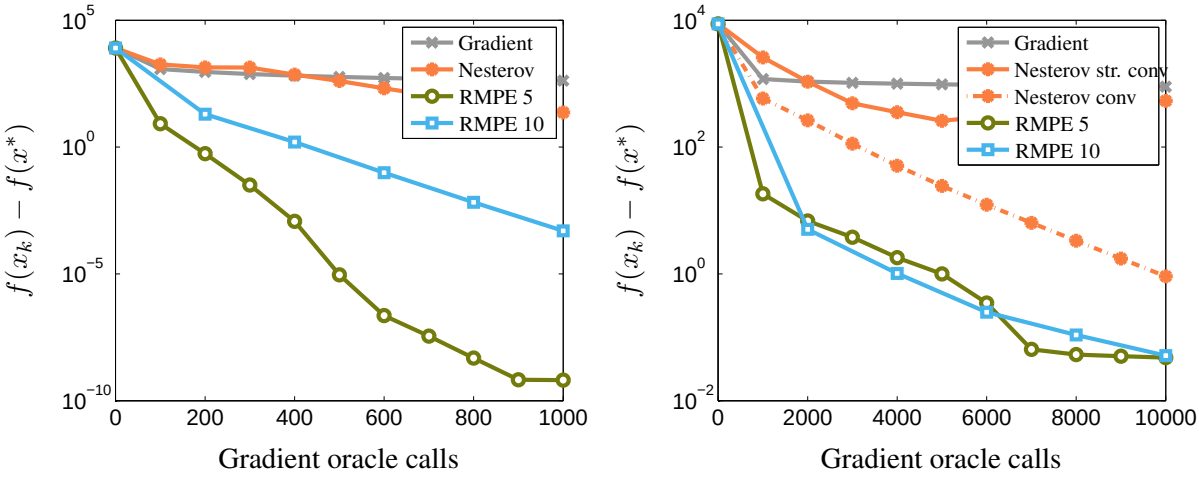

FIGURE 5. Logistic regression on *Sido0 Scaled UCI Dataset* (4932 features, 12678 data points), solved using several algorithms. *Left:* the penalty parameter has been set to 10^2 in order to have a condition number equal to 1.57×10^5. *Right:* penalty parameter equal to 10^{-1}, so the condition number is equal to 1.57×10^8.

REFERENCES

Aitken, A. C. [1927], 'On Bernoulli's numerical solution of algebraic equations', *Proceedings of the Royal Society of Edinburgh* **46**, 289–305.

Beck, A. and Teboulle, M. [2009], 'A fast iterative shrinkage-thresholding algorithm for linear inverse problems', *SIAM Journal on Imaging Sciences* **2**(1), 183–202.

Ben-Tal, A. and Nemirovski, A. [2001], *Lectures on modern convex optimization : analysis, algorithms, and engineering applications*, MPS-SIAM series on optimization, SIAM.

Brezinski, C. [1977], 'Accélération de la convergence en analyse numérique', *Lecture notes in mathematics (ISSN 0075-8434* (584).

Cabay, S. and Jackson, L. [1976], 'A polynomial extrapolation method for finding limits and antilimits of vector sequences', *SIAM Journal on Numerical Analysis* **13**(5), 734–752.

Drori, Y. and Teboulle, M. [2014], 'Performance of first-order methods for smooth convex minimization: a novel approach', *Mathematical Programming* **145**(1-2), 451–482.

Durbin, J. [1960], 'The fitting of time-series models', *Revue de l'Institut International de Statistique* pp. 233–244.

Eddy, R. [1979], 'Extrapolating to the limit of a vector sequence', *Information linkage between applied mathematics and industry* pp. 387–396.

Golub, G. H. and Varga, R. S. [1961], 'Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order richardson iterative methods', *Numerische Mathematik* **3**(1), 157–168.

Hardt, M. [2013], 'The zen of gradient descent', *Mimeo* .

Heinig, G. and Rost, K. [2011], 'Fast algorithms for Toeplitz and Hankel matrices', *Linear Algebra and its Applications* **435**(1), 1–59.

Lasserre, J. B. [2001], 'Global optimization with polynomials and the problem of moments', *SIAM Journal on Optimization* **11**(3), 796–817.

Lessard, L., Recht, B. and Packard, A. [2016], 'Analysis and design of optimization algorithms via integral quadratic constraints', *SIAM Journal on Optimization* **26**(1), 57–95.

Levinson, N. [1949], 'The Wiener RMS error criterion in filter design and prediction, appendix b of wiener, n.(1949)', *Extrapolation, Interpolation, and Smoothing of Stationary Time Series* .

Lin, H., Mairal, J. and Harchaoui, Z. [2015], A universal catalyst for first-order optimization, *in* 'Advances in Neural Information Processing Systems', pp. 3366–3374.

Mešina, M. [1977], 'Convergence acceleration for the iterative solution of the equations $x = ax + f$', *Computer Methods in Applied Mechanics and Engineering* **10**(2), 165–173.

Nemirovskii, A. and Nesterov, Y. E. [1985], 'Optimal methods of smooth convex minimization', *USSR Computational Mathematics and Mathematical Physics* **25**(2), 21–30.

Nesterov, Y. [1983], 'A method of solving a convex programming problem with convergence rate $O(1/k^2)$', *Soviet Mathematics Doklady* **27**(2), 372–376.

Nesterov, Y. [2000], Squared functional systems and optimization problems, *in* 'High performance optimization', Springer, pp. 405–440.

Nesterov, Y. [2003], *Introductory Lectures on Convex Optimization*, Springer.

Nesterov, Y. [2015], 'Universal gradient methods for convex optimization problems', *Mathematical Programming* **152**(1-2), 381–404.

Parrilo, P. [2000], Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization, PhD thesis, California Institute of Technology.

Shanks, D. [1955], 'Non-linear transformations of divergent and slowly convergent sequences', *Journal of Mathematics and Physics* **34**(1), 1–42.

Sidi, A., Ford, W. F. and Smith, D. A. [1986], 'Acceleration of convergence of vector sequences', *SIAM Journal on Numerical Analysis* **23**(1), 178–196.

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

Su, W., Boyd, S. and Candes, E. [2014], A differential equation for modeling nesterov's accelerated gradient method: Theory and insights, *in* 'Advances in Neural Information Processing Systems', pp. 2510–2518.

Tyrtyshnikov, E. E. [1994], 'How bad are Hankel matrices?', *Numerische Mathematik* **67**(2), 261–269.

Wibisono, A. and Wilson, A. C. [2015], 'On accelerated methods in optimization', *arXiv preprint arXiv:1509.03616* .

Wynn, P. [1956], 'On a device for computing the $e_m(s_n)$ transformation', *Mathematical Tables and Other Aids to Computation* **10**(54), 91–96.

ACKNOWLEDGEMENTS

AA is at CNRS, attached to the Département d'Informatique at École Normale Supérieure in Paris, INRIA-Sierra team, PSL Research University. The authors would like to acknowledge support from a starting grant from the European Research Council (ERC project SIPA), as well as support from the chaire *Économie des nouvelles données* with the *data science* joint research initiative with the *fonds AXA pour la recherche* and a gift from Société Générale Cross Asset Quantitative Research.

APPENDIX A. COMPUTATION OF ERROR BOUNDS WHEN ACCELERATING GRADIENT METHOD

Here, we fully explicit the bounds used in Section 3.4 for the simple gradient method on smooth, strongly convex functions with Lipschitz-continuous Hessian.

A.1. **Upper bound for $\|X - X^*\|_2$ and $\|U\|_2$.** Since we assumed that we used the gradient method, the sequence $\|\tilde{x}_k - x^*\|_2$ is monotone, following

$$\|\tilde{x}_{k+1} - x^*\| \leq r\|\tilde{x}_k - x^*\|$$

In this case, we can bound easily $\|X - X^*\|_2$ with

$$\begin{aligned}\|\tilde{X} - X^*\|_2 &\leq \sum_{i=0}^{k} \|\tilde{x}_i - x^*\|_2 \\ &= \frac{1-r^k}{1-r}\|x_0 - x^*\|_2\end{aligned}$$

Moreover, in the asymptotic model we have

$$\|x_{k+1} - x^*\|_2 = \|A(x_k - x^*)\|_2 \leq \|A\|_2\|x_k - x^*\|_2$$

where $A = I - \frac{1}{L}\nabla^2 f(x^*)$, so $\|A\|_2 \leq 1 - \frac{L}{\mu}$. In that case,

$$\begin{aligned}\|U\|_2 &\leq \|A - I\|_2 \sum_{i=0}^{k} \|x_i - x^*\|_2 \\ &\leq \sum_{i=0}^{k} \|A\|^i \|x_0 - x^*\|_2 \\ &\leq \frac{1 - \|A\|_2^k}{1 - \|A\|_2}\|x_0 - x^*\|_2 \\ &\leq \frac{L}{\mu}\left(1 - \left(1 - \frac{\mu}{L}\right)^k\right)\|x_0 - x^*\|_2\end{aligned}$$

A.2. **Upper bound for $\|\mathcal{E}\|_2 = \|\tilde{X} - X\|_2$ and $\|E\|_2 = \|\tilde{U} - U\|$.** Since $\mathcal{E}_i = \tilde{x}_i - x_i$, we have

$$\|\mathcal{E}\|_2 \leq \sum_{i=0}^{k} \|\tilde{x}_i - x_i\|_2$$

We will express $\|\tilde{x}_{i+1} - x_{i+1}\|_2$ in function of $\|\tilde{x}_0 - x_0\|_2$ using a recursion with $\|\tilde{x}_i - x_i\|_2$:

$$\begin{aligned}
\tilde{x}_{i+1} - x_{i+1} &= \tilde{x}_i - \frac{1}{L}\nabla f(\tilde{x}_i) - x_i + \frac{1}{L}\nabla^2 f(x^*)(x_i - x^*) \\
&= \tilde{x}_i - x_i - \frac{1}{L}(\nabla f(\tilde{x}_i) - \nabla^2 f(x^*)(x_i - x^*)) \\
&= \left(I - \frac{\nabla^2 f(x^*)}{L}\right)(\tilde{x}_i - x_i) - \frac{1}{L}(\nabla f(\tilde{x}_i) - \nabla^2 f(x^*)(\tilde{x}_i - x^*))
\end{aligned}$$

Since our function has a Lipschitz-continuous Hessian, it is possible to show that (Nesterov [2003], Lemma 1.2.4)

$$\left\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y - x)\right\|_2 \leq \frac{M}{2}\|y - x\|^2. \tag{24}$$

We can thus bound the norm of the error at the i^{th} iteration

$$\begin{aligned}
\|x_{i+1} - \tilde{x}_{i+1}\|_2 &\leq \left\|I - \frac{\nabla^2 f(x^*)}{L})\right\|_2 \|x_i - \tilde{x}_i\|_2 + \frac{1}{L}\left\|\nabla f(\tilde{x}_i) - \nabla^2 f(x^*)(\tilde{x}_i - x^*)\right\|_2 \\
&= \left\|I - \frac{\nabla^2 f(x^*)}{L})\right\|_2 \|x_i - \tilde{x}_i\|_2 + \frac{1}{L}\left\|\nabla f(\tilde{x}_i) - \nabla f(x^*) - \nabla^2 f(x^*)(\tilde{x}_i - x^*)\right\|_2
\end{aligned}$$

By equation (24), we have

$$\begin{aligned}
\|x_{i+1} - \tilde{x}_{i+1}\|_2 &\leq (1 - \frac{\mu}{L})\|x_i - \tilde{x}_i\|_2 + \frac{M}{2L}\|\tilde{x}_i - x^*\|_2^2 \\
&\leq (1 - \frac{\mu}{L})\|x_i - \tilde{x}_i\|_2 + \frac{M}{2L}(r^2)^i\|x_0 - x^*\|_2^2 \\
&= \frac{M}{2L}\sum_{j=1}^{i}(1 - \frac{\mu}{L})^{i-j}(r^2)^j\|x_0 - x^*\|_2^2
\end{aligned}$$

The sum starts at $j = 1$ because, by definition, $\|e_0\|_2 = \|\tilde{x}_0 - x_0\|_2 = 0$. We can simplify this expression using the following property:

$$\frac{r^2}{1 - \frac{\mu}{L}} = \frac{1}{1 + \frac{\mu}{L}} < 1, \tag{25}$$

then bound $\|x_{i+1} - \tilde{x}_{i+1}\|_2$ as follows.

$$\begin{aligned}
\|x_{i+1} - \tilde{x}_{i+1}\|_2 &\leq \left(1 - \frac{\mu}{L}\right)^i \frac{M}{2L}\sum_{j=1}^{i}\left(\frac{r^2}{1 - \frac{\mu}{L}}\right)^j \|x_0 - x^*\|_2^2 \\
&\leq \left(1 - \frac{\mu}{L}\right)^i \frac{M}{2L}\left(\sum_{j=0}^{i}\left(\frac{r^2}{1 - \frac{\mu}{L}}\right)^j\right)\|x_0 - x^*\|_2^2.
\end{aligned}$$

By equation 25, we can simplify the bound:

$$\left(1 - \frac{\mu}{L}\right)^i \frac{M}{2L}\left(\sum_{j=0}^{i}\left(\frac{r^2}{1 - \frac{\mu}{L}}\right)^j\right)\|x_0 - x^*\|_2^2 = \left(1 - \frac{\mu}{L}\right)^i \frac{M}{2L}\left(\sum_{j=0}^{i}\left(\frac{1}{1 + \frac{\mu}{L}}\right)^j\right)\|x_0 - x^*\|_2^2$$

We have thus

$$\begin{aligned}
|x_{i+1} - \tilde{x}_{i+1}\|_2 &\leq \left(1 - \frac{\mu}{L}\right)^i \frac{M}{2L} \left(\sum_{j=0}^{i} \left(\frac{1}{1+\frac{\mu}{L}} \right)^j \right) \|x_0 - x^*\|_2^2 \\
&= \left(1 - \frac{\mu}{L}\right)^i \frac{M}{2L} \left(\frac{1 - \left(\frac{1}{1+\frac{\mu}{L}} \right)^i}{1 - \frac{1}{1+\frac{\mu}{L}}} \right) \|x_0 - x^*\|_2^2 \\
&= \frac{M}{2L} \left(\frac{(1-\frac{\mu}{L})^i - \left(\frac{1-\frac{\mu}{L}}{1+\frac{\mu}{L}} \right)^i}{1 - \frac{1}{1+\frac{\mu}{L}}} \right) \|x_0 - x^*\|_2^2 \\
&= \left(1 + \frac{L}{\mu}\right) \frac{M}{2L} \left((1 - \frac{\mu}{L})^i - \left(\frac{1-\frac{\mu}{L}}{1+\frac{\mu}{L}} \right)^i \right) \|x_0 - x^*\|_2^2.
\end{aligned}$$

By summing these error terms, we get

$$\begin{aligned}
\|\mathcal{E}\|_2 &\leq \sum_{i=0}^{k} \|x_i - \tilde{x}_i\|_2 \\
&\leq \left(1 + \frac{L}{\mu}\right) \frac{M}{2L} \left(\sum_{i=0}^{k} (1 - \frac{\mu}{L})^i - \sum_{i=0}^{k} \left(\frac{1-\frac{\mu}{L}}{1+\frac{\mu}{L}} \right)^i \right) \|x_0 - x^*\|_2^2 \\
&= \left(1 + \frac{L}{\mu}\right) \frac{M}{2L} \left(\frac{L}{\mu} \left(1 - \left(1 - \frac{\mu}{L} \right)^k \right) - \frac{L}{2\mu} \left(1 - \left(\frac{1-\frac{\mu}{L}}{1+\frac{\mu}{L}} \right)^k \right) \right) \|x_0 - x^*\|_2^2 \\
&\leq \left(1 + \frac{L}{\mu}\right)^2 \frac{M}{2L} \left(1 - \left(1 - \frac{\mu}{L} \right)^k - \frac{1}{2} \left(1 - \left(\frac{1-\frac{\mu}{L}}{1+\frac{\mu}{L}} \right)^k \right) \right) \|x_0 - x^*\|_2^2 \\
&= \left(1 + \frac{L}{\mu}\right)^2 \frac{M}{2L} \left(\frac{1}{2} - \left(1 - \frac{\mu}{L} \right)^k + \frac{1}{2} \left(\frac{1-\frac{\mu}{L}}{1+\frac{\mu}{L}} \right)^k \right) \|x_0 - x^*\|_2^2
\end{aligned}$$

We finally have

$$\|\tilde{U} - U\|_2 = \|E\|_2 \leq 2\|\mathcal{E}\|_2 = \|\tilde{X} - X\|_2.$$

APPENDIX B. TWO-TERMS RECURENCE FOR CHEBYSHEV'S ACCELERATION

In this section we will details all steps to go from the theoretical definition of Chebyshev's acceleration

$$y_k = T_k(A)(x_0 - x^*)$$

to the two-terms recurence

$$\begin{aligned}
\alpha_k &= t(1)\alpha_{k-1} - \alpha_{k-2} \\
z_{k-1} &= y_{k-1} - \frac{1}{L}(Ay_{k-1} - b) \\
y_k &= \frac{\alpha_{k-1}}{\alpha_k} \left(\frac{2z_{k-1}}{\sigma} - y_{k-1} \right) - \frac{\alpha_{k-2}}{\alpha_k} y_{k-2}
\end{aligned}$$

Define $\alpha_k = C_k(t(1))$. Then, by definition of $T_k(x)$,

$$\alpha_k T_k(x) = C_k(t(x))$$

So, we get easily a three-term recurrence for $T_k(x)$

$$\begin{aligned}\alpha_k T_k(x) &= C_k(t(x)) \\ &= t(x)C_{k-1}(t(x)) - C_{k-2}(t(x)) \\ &= t(x)\alpha_{k-1}T_{k-1}(x) - \alpha_{k-2}T_{k-2}(x)\end{aligned}$$

and also for α_k:

$$\begin{aligned}\alpha_k = C_k(t(1)) &= t(1)\alpha_{k-1}C_{k-1}(t(1)) - \alpha_{k-2}C_{k-2}(t(1)) \\ &= t(1)\alpha_{k-1} - \alpha_{k-2}\end{aligned}$$

We will now see that we can form $y_k = \sum_{i=0}^{k} c_i x_i$ using only y_{k-1} and y_{k-2}, where x_k comes from the gradient method and c_i are the coefficients of $T_k(x)$. We have

$$\begin{aligned}y_k - x^* &= T_k\left(A\right)\left(x_0 - x^*\right) \\ &= \frac{1}{\alpha_k}\left(t\left(A\right)\alpha_{k-1}T_{k-1}(A)(x_0 - x^*) - \alpha_{k-2}T_{k-2}(A)(x_0 - x^*)\right) \\ &= \frac{1}{\alpha_k}\left(\alpha_{k-1}t\left(A\right)\left(y_{k-1} - x^*\right) - \alpha_{k-2}(y_{k-2} - x^*)\right)\end{aligned}$$

We need now to compute $t\left(A\right)y_{k-1}$. Since $t(x) = \frac{2x-\sigma}{\sigma}$, we have

$$t\left(A\right)\left(y_{k-1} - x^*\right) = \frac{2\left(A\right)\left(y_{k-1} - x^*\right) - \sigma(y_{k-1} - x^*)}{\sigma}$$

However, we have access to $x^* + A(y_{k-1} - x^*)$, since it correspond to a gradient step. Define

$$z_{k-1} = x^* + A(y_{k-1} - x^*)$$

So,

$$t\left(A\right)\left(y_{k-1} - x^*\right) = \frac{2(z_{k-1} - x^*) - \sigma(y_{k-1} - x^*)}{\sigma} = \frac{2z_{k-1} - \sigma y_{k-1}}{\sigma} - t(1)x^*$$

If we plug this expression in the three-term recurrence,

$$y_k - x^* = \frac{1}{\alpha_k}\left(\alpha_{k-1}\left(\frac{2z_{k-1} - \sigma y_{k-1}}{\sigma} - t(1)x^*\right) - \alpha_{k-2}(y_{k-2} - x^*)\right)$$

Using the definition of α_k, we can eliminate x^* in both side. This leads to the following scheme.

$$\begin{aligned}z_{k-1} &= y_{k-1} - \frac{1}{L}(By_{k-1} - b) \\ y_k &= \frac{\alpha_{k-1}}{\alpha_k}\left(\frac{2z_{k-1}}{\sigma} - y_{k-1}\right) - \frac{\alpha_{k-2}}{\alpha_k}y_{k-2}\end{aligned}$$

INRIA & D.I.,
ÉCOLE NORMALE SUPÉRIEURE, PARIS, FRANCE.
E-mail address: damien.scieur@inria.fr

CNRS & D.I., UMR 8548,
ÉCOLE NORMALE SUPÉRIEURE, PARIS, FRANCE.
E-mail address: aspremon@ens.fr

INRIA & D.I.
ÉCOLE NORMALE SUPÉRIEURE, PARIS, FRANCE.
E-mail address: francis.bach@inria.fr