

NONLINEAR ACCELERATION OF MOMENTUM AND PRIMAL-DUAL ALGORITHMS

RAGHU BOLLAPRAGADA

Corresponding author.

*Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL, USA.
The author was a PhD student in the department of Industrial Engineering and Management Sciences at
Northwestern University, IL, USA, when this work was done.*

DAMIEN SCIEUR

SAMSUNG SAIL, Montreal, Canada.

*This author was a PhD student at INRIA & D.I., UMR 8548, École Normale Supérieure, Paris, France,
when this work was done.*

ALEXANDRE D'ASPREMONT

CNRS & D.I., UMR 8548, École Normale Supérieure, Paris, France.

ABSTRACT. We describe convergence acceleration schemes for multistep optimization algorithms. The extrapolated solution is written as a nonlinear average of the iterates produced by the original optimization method. Our analysis does not need the underlying fixed-point operator to be symmetric, hence handles e.g. algorithms with momentum terms such as Nesterov's accelerated method, or primal-dual methods. The weights are computed via a simple linear system and we analyze performance in both online and offline modes. We use Crouzeix's conjecture to show that acceleration performance is controlled by the solution of a Chebyshev problem on the numerical range of a non-symmetric operator modeling the behavior of iterates near the optimum. Numerical experiments are detailed on logistic regression problems.

1. INTRODUCTION

Extrapolation techniques, such as Aitken's Δ^2 or Wynn's ε -algorithm, provide an improved estimate of the limit of a sequence using its last few iterates, and we refer the reader to [Brezinski and Zaglia, 2013] for a complete survey. These methods have been extended to vector sequences, where they are known under various names, e.g. Anderson acceleration [Anderson, 1965, Walker and Ni, 2011], minimal polynomial extrapolation [Cabay and Jackson, 1976] or reduced rank extrapolation [Eddy, 1979].

Classical optimization algorithms typically retain only the last iterate or the average of iterates [Polyak and Juditsky, 1992] as their best estimate of the optimum, throwing away all the information contained in the converging sequence. This is highly wasteful from a statistical perspective and extrapolation schemes estimate instead the optimum using a weighted average of the last iterates produced by the underlying algorithm, where the weights depend on the iterates (i.e. a *nonlinear* average). Overall, computing those weights means solving a small linear system so nonlinear acceleration has marginal computational complexity.

E-mail addresses: raghu.bollapragada@u.northwestern.edu, damien.scieur@gmail.com, aspremon@gmail.com.

Date: October 18, 2019.

Recent results by [Scieur et al., 2016] adapted classical extrapolation techniques related to Aitken’s Δ^2 , Anderson’s method and minimal polynomial extrapolation to design extrapolation schemes for accelerating the convergence of basic optimization methods such as gradient descent. They showed that using only iterates from fixed-step gradient descent, extrapolation algorithms achieve the optimal convergence rate of [Nesterov, 2013] *without any modification to the original algorithm*. However, these results are only applicable to iterates produced by single-step algorithms such as gradient descent, where the underlying operator is symmetric, thus excluding much faster momentum-based methods such as SGD with momentum or Nesterov’s algorithm. Our results here seek to extend those of [Scieur et al., 2016] to multistep methods, i.e. to accelerate accelerated methods.

Our contribution here is twofold. First, we show that the accelerated convergence bounds in [Scieur et al., 2016] can be directly extended to multistep methods when the operator describing convergence near the optimum has a particular block structure, by modifying the extrapolating sequence. This result applies in particular to Nesterov’s method and the stochastic gradient algorithms with a momentum term. Second, we use Crouzeix’s recent results [Crouzeix, 2007, Crouzeix and Palencia, 2017, Greenbaum et al., 2017] to show that, in the general non-symmetric case, acceleration performance is controlled by the solution of a Chebyshev problem on the numerical range of the linear, non-symmetric operator modelling the behavior of iterates near the optimum. We characterize the shape of this numerical range for various classical multistep algorithms such as Nesterov’s method [Nesterov, 1983], and Chambolle-Pock’s algorithm [Chambolle and Pock, 2011].

We then study the performance of our technique on a logistic regression problem. The online version (which modifies iterations) is competitive with L-BFGS in our experiments and significantly faster than classical accelerated algorithms. Furthermore, it is robust to miss-specified strong convexity parameters.

Organization of the paper. In Section 2, we describe the iteration schemes that we seek to accelerate, introduce the Regularized Nonlinear Acceleration (RNA) scheme, and show how to control its convergence rate for linear iterations (e.g. solving quadratic problems).

In Section 3 we show how to bound the convergence rate of acceleration schemes on generic nonsymmetric iterates using Crouzeix’s conjecture and bounds on the minimum of a Chebyshev problem written on the numerical range of the nonsymmetric operator. We apply these results to Nesterov’s method and the Chambolle-Pock primal-dual algorithm in Section 4.

We extend our results to generic nonlinear updates using a constrained formulation of RNA (called CNA) in Section 5. We show optimal convergence rates in the symmetric case for CNA on simple gradient descent with linear combination of previous iterates in Section 6, producing a much cleaner proof of the results in [Scieur et al., 2016] on RNA. In Section 7, we show that RNA can be applied online, i.e. that we can extrapolate iterates produced by an extrapolation scheme at each iteration (previous results only worked in batch mode) and apply this result to speed up Nesterov’s method.

2. NONLINEAR ACCELERATION

We begin by describing the iteration template for the algorithms to which we will apply acceleration schemes.

2.1. General setting. Consider 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. We consider the following class of algorithms, written

$$\begin{cases} x_i = g(y_{i-1}) \\ y_i = \sum_{j=1}^i \alpha_j^{(i)} x_j + \beta_j^{(i)} y_{j-1}, \end{cases} \tag{2}$$

where $x_i, y_i \in \mathbb{R}^d$ and $g : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is an iterative update, potentially stochastic. For example, $g(x)$ can be a gradient step with fixed stepsize, in which case $g(x) = x - h\nabla f(x)$. We assume the following condition on the coefficients α and β , to ensure consistency [Scieur et al., 2017b],

$$\mathbf{1}^T(\alpha + \beta) = 1, \quad \forall k, \alpha_j \neq 0.$$

We can write these updates in matrix format, with

$$X_i = [x_1, x_2, \dots, x_i], \quad Y_i = [y_0, y_1, \dots, y_{i-1}]. \quad (3)$$

Using this notation, (2) reads (assuming $x_0 = y_0$)

$$X_i = g(Y_{i-1}), \quad Y_i = [x_0, X_i]L_i, \quad (4)$$

where $g(Y)$ stands for $[g(y_0), g(y_1), \dots, g(y_{i-1})]$ and the matrix L_i is upper-triangular of size $i \times i$ with nonzero diagonal coefficients, with columns summing to one. The matrix L_i can be constructed iteratively, following the recurrence

$$L_i = \begin{bmatrix} L_{i-1} & \alpha_{[1:i-1]} + L_{i-1}\beta \\ 0_{1 \times i-1} & \alpha_i \end{bmatrix}, \quad L_0 = 1. \quad (5)$$

In short, L_i gathers coefficients from the linear combination in (2). This matrix, together with g , characterizes the algorithm.

The iterate update form (2) is generic and includes many classical algorithms such as the accelerated gradient method in [Nesterov, 2013], where

$$\begin{cases} x_i &= g(y_{i-1}) = y_{i-1} - \frac{1}{L} \nabla f(y_{i-1}) \\ y_i &= \left(1 + \frac{i-1}{i+2}\right) x_i - \frac{i-1}{i+2} x_{i-1}. \end{cases}$$

As in [Scieur et al., 2016] we will focus on improving our estimates of the solution to problem (1) by tracking only the sequence of iterates (x_i, y_i) produced by an optimization algorithm, without any further oracle calls to $g(x)$. The main difference with the work of [Scieur et al., 2016] is the presence of a linear combination of previous iterates in the definition of y in (2), so the mapping from x_i to x_{i+1} is usually *non-symmetric*. For instance, for Nesterov's algorithm, the Jacobian of x_{i+1} with respect to x_i, y_i reads

$$J_{x_{i+1}} = \begin{bmatrix} 0 & J_g \\ \left(1 + \frac{i-2}{i+1}\right) \mathbf{I} & -\frac{i-2}{i+1} \mathbf{I} \end{bmatrix} \neq J_{x_{i+1}}^T$$

where $J_{x_{i+1}}$ is the Jacobian of the function g evaluated at x_{i+1} . In what follows, we show that looking at the residuals

$$r(x) \triangleq g(x) - x, \quad r_i = r(y_{i-1}) = x_i - y_{i-1}, \quad R_i = [r_1 \dots r_i], \quad (6)$$

allows us to recover the convergence results from [Scieur et al., 2016] when the Jacobian of the function g , written J_g , is symmetric. Moreover, we extend the analysis for *non symmetric* Jacobians. This allows us to accelerate for instance accelerated methods or primal-dual methods. We now briefly recall the key ideas driving nonlinear acceleration schemes.

2.2. Linear Algorithms. In this section, we focus on iterative algorithms g that are linear, i.e., where

$$g(x) = G(x - x^*) + x^*. \quad (7)$$

The matrix G is of size $d \times d$, and, contrary to [Scieur et al., 2016], we do not assume symmetry. Here, x^* is a fixed point of g . In optimization problems where g is typically a gradient mapping x^* is the minimum of an objective function. Its worth mentioning that (7) is equivalent to $Ax + b$, thus we do not require x^* to evaluate the mapping $g(x)$. We first treat the case where $g(x)$ is linear, as the nonlinear case will then be handled as a perturbation of the linear one.

We introduce $\mathcal{P}_{[N]}^{(1)}$, the set of all polynomials p whose degree is *exactly* N (i.e., the leading coefficient is nonzero), and whose coefficients sum to one. More formally,

$$\mathcal{P}_{[N]}^{(1)} = \{p \in \mathbb{R}[x] : \deg(p) = N, p(1) = 1\}. \quad (8)$$

The following proposition extends a result by [Scieur et al. \[2016\]](#) showing that iterates in (2) can be written using polynomials in $\mathcal{P}_{[N]}^{(1)}$. This formulation is helpful to derive the rate of converge of the Nonlinear Acceleration algorithm.

Proposition 2.1. *Let g be the linear function (7). Then, the N -th iteration of (2) is equivalent to*

$$x_N = x^* + G(y_{N-1} - x^*), \quad y_N = x^* + p_N(G)(x_0 - x^*), \quad \text{for some } p_N \in \mathcal{P}_{[N]}^{(1)}. \quad (9)$$

Proof. We prove (9) iteratively. Of course, at iteration zero,

$$y_0 = x^* + 1 \cdot (x_0 - x^*),$$

and 1 is indeed polynomial of degree zero whose coefficient sum to one. Now, assume

$$y_{i-1} - x^* = p_{i-1}(G)(x_0 - x^*), \quad p_{i-1} \in \mathcal{P}_{[i-1]}^{(1)}.$$

We show that

$$y_i - x^* = p_i(G)(x_0 - x^*), \quad p_i \in \mathcal{P}_{[i]}^{(1)}.$$

By definition of y_i in (2),

$$y_i - x^* = \sum_{j=1}^i \alpha_j^{(i)} x_j + \beta_j^{(i)} y_{i-1} - x^*,$$

where $(\alpha + \beta)^T \mathbf{1} = 1$. This also means that

$$y_i - x^* = \sum_{j=1}^i \alpha_j^{(i)} (x_j - x^*) + \beta_j^{(i)} (y_{j-1} - x^*).$$

By definition, $x_j - x^* = G(y_{j-1} - x^*)$, so

$$y_i - x^* = \sum_{j=1}^i \left(\alpha_j^{(i)} G + \beta_j^{(i)} I \right) (y_{j-1} - x^*).$$

By the recurrence assumption,

$$y_i - x^* = \sum_{j=1}^i \left(\alpha_j^{(i)} G + \beta_j^{(i)} I \right) p_{j-1}(G)(x_0 - x^*),$$

which is a linear combination of polynomials, thus $y_i - x^* = p(G)(x_0 - x^*)$. It remains to show that $p \in \mathcal{P}_{[i]}^{(1)}$. Indeed,

$$\deg(p) = \max_j \max \left\{ (1 + \deg(p_{j-1}(G))) 1_{\alpha_j \neq 0}, \deg(p_{j-1}(G)) 1_{\beta_j \neq 0} \right\},$$

where $1_{\alpha_j \neq 0} = 1$ if $\alpha_j \neq 0$ and 0 otherwise. By assumption, $\alpha_i \neq 0$ thus

$$\deg(p) \geq 1 + \deg(p_{i-1}(G)) = i.$$

Since p is a linear combination of polynomials of degree at most i ,

$$\deg(p) = i.$$

It remains to show that $p(1) = 1$. Indeed,

$$p(1) = \sum_{j=1}^i \left(\alpha_j^{(i)} 1 + \beta_j^{(i)} \right) p_{j-1}(1).$$

Since $\left(\alpha_j^{(i)} 1 + \beta_j^{(i)} \right) = 1$ and $p_{j-1}(1) = 1$, $p(1) = 1$ and this proves the proposition. ■

2.3. Regularized Nonlinear Acceleration Scheme. We now propose a modification of RNA that can accelerate any algorithm of the form (2) by combining the approaches of [Anderson \[1965\]](#) and [Scieur et al. \[2016\]](#). We introduce a mixing parameter η , as in Anderson acceleration (which only impact the constant term in the rate of convergence). Throughout this paper, **RNA** will refer to Algorithm 1 below.

Algorithm 1 Regularized Nonlinear Acceleration (**RNA**)

- 1: **Data:** Matrices X and Y of size $d \times N$ constructed from the iterates as in (2) and (3).
- 2: **Parameters:** Mixing $\eta \neq 0$, regularization $\lambda \geq 0$.

- 3: **1.** Compute matrix of residuals $R = X - Y$.
- 4: **2.** Solve

$$c^\lambda = \frac{(R^T R + (\lambda \|R\|_2^2)I)^{-1} \mathbf{1}_N}{\mathbf{1}_N^T (R^T R + (\lambda \|R\|_2^2)I)^{-1} \mathbf{1}_N}. \quad (10)$$

- 5: **3.** Compute extrapolated solution $y^{\text{extr}} = (Y - \eta R)c^\lambda$.
-

2.4. Computational Complexity. Scieur et al. [2016] discuss the complexity of Algorithm 1 in the case where N is small (compared to d). When the algorithm is used once on X and Y (batch acceleration), the computational complexity is $O(N^2 d)$, because we have to multiply R^T and R . However, when Algorithm (1) accelerates iterates on-the-fly, the matrix $R^T R$ can be updated using only $O(Nd)$ operations. The complexity of solving the linear system is negligible as it takes only $O(N^3)$ operation. Even if the cubic dependence is bad for large N , in our experiment N is typically equal to 10, thus adding a negligible computational overhead compared to the computation of a gradient in large dimension which is higher by orders.

2.5. Convergence Rate. We now analyze the convergence rate of Algorithm 1 with $\lambda = 0$, which corresponds to Anderson acceleration [Anderson, 1965]. In particular, we show its optimal rate of convergence when g is a linear function. In the context of optimization, this is equivalent to the application of gradient descent for minimizing quadratics. Using this special structure, the iterations (9) produce a sequence of polynomials and the next theorem uses this special property to bound the convergence rate. Compared to previous work in this vein [Scieur et al., 2016, 2017a] where the results only apply to algorithm of the form $x_{i+1} = g(x_i)$, this theorem applies to *any* algorithm of the class (2) where in particular, we allow G to be nonsymmetric.

Theorem 2.2. *Let X, Y in (3) be formed using iterates from (2). Let g be defined in (7), where $G \in \mathbb{R}^{d \times d}$ does not have 1 as eigenvalue. The norm of the residual of the extrapolated solution y^{extr} , written*

$$r(y^{\text{extr}}) = g(y^{\text{extr}}) - y^{\text{extr}},$$

produced by Algorithm 1 with $\lambda = 0$, is bounded by

$$\|r(y^{\text{extr}})\|_2 \leq \|I - \eta(G - I)\|_2 \|p_{N-1}^*(G)r(x_0)\|_2,$$

where p_{N-1}^ solves*

$$p_{N-1}^* = \operatorname{argmin}_{p \in \mathcal{P}_{[N-1]}^{(1)}} \|p(G)r(x_0)\|_2. \quad (11)$$

Moreover, after at most d iterations, the algorithm converges to the exact solution, satisfying $\|r(y^{\text{extr}})\|_2 = 0$.

Proof. First, we write the definition of y^{extr} from Algorithm 1 when $\lambda = 0$,

$$y^{\text{extr}} - x^* = (Y - \eta R)c - x^*.$$

Since $c^T \mathbf{1} = 1$, we have $X^* c = x^*$, where $X^* = [x^*, x^*, \dots, x^*]$. Thus,

$$y^{\text{extr}} - x^* = (Y - X^* - \eta R)c.$$

Since $R = G(Y - X^*)$,

$$y^{\text{extr}} - x^* = (I - \eta(G - I))(Y - X^*)c.$$

We have seen that the columns of $Y - X^*$ are polynomials of different degrees, whose coefficients sums to one (9). This means

$$y^{\text{extr}} - x^* = (I - \eta(G - I)) \sum_{i=0}^{N-1} c_i p_i(G)(x_0 - x^*).$$

In addition, its residual reads

$$\begin{aligned}
r(y^{\text{extr}}) &= (G - I)(y^{\text{extr}} - x^*) \\
&= (G - I)(I - \eta(G - I)) \sum_{i=0}^{N-1} p_i(G)(x_0 - x^*) \\
&= (I - \eta(G - I)) \sum_{i=0}^{N-1} c_i p_i(G) r(x_0).
\end{aligned}$$

Its norm is thus bounded by

$$\|r(y^{\text{extr}})\| \leq \|I - \eta(G - I)\| \underbrace{\left\| \sum_{i=0}^{N-1} c_i p_i(G) r(x_0) \right\|}_{=Rc}.$$

By definition of c from Algorithm 1,

$$\|r(y^{\text{extr}})\| \leq \|I - \eta(G - I)\| \min_{c: c^T \mathbf{1} = 1} \left\| \sum_{i=0}^{N-1} c_i p_i(G) r(x_0) \right\|.$$

Because p_i are all of degree exactly equal to i , the p_i are a basis of the set of all polynomial of degree at most $N - 1$. In addition, because $p_i(1) = 1$, restricting the sum of coefficients c_i to 1 generates the set $\mathcal{P}_{[N-1]}^{(1)}$. We have thus

$$\|r(y^{\text{extr}})\| \leq \|I - \eta(G - I)\| \min_{p \in \mathcal{P}_{[N-1]}^{(1)}} \|p(G)r_0\|.$$

Finally, when $N > d$, it suffice to take the minimal polynomial of the matrix G named $p_{\min, G}$, whose coefficient are normalized by $p_{\min, G}(1)$. Since the eigenvalues of G are strictly inferior to 1, $p_{\min, G}(1)$ cannot be zero. ■

In optimization, the quantity $\|r(y^{\text{extr}})\|_2$ is proportional to the norm of the gradient of the objective function computed at y^{extr} . This last theorem reduces the analysis of the rate of convergence of RNA to the analysis of the quantity (11). In the symmetric case discussed in [Scieur et al., 2016], this bound recovers the optimal rate in [Nesterov, 2013] which also appears in the complexity analysis of Krylov methods (like GMRES or conjugate gradients [Golub and Varga, 1961, Golub and Van Loan, 2012]) for quadratic minimization.

3. CROUZEIX'S CONJECTURE & CHEBYSHEV POLYNOMIALS ON THE NUMERICAL RANGE

We have seen in (11) from Theorem 2.2 that the convergence rate of nonlinear acceleration is controlled by the norm of a matrix polynomial in the operator G , with

$$\|r(y^{\text{extr}})\|_2 \leq \|I - \eta(G - I)\|_2 \|p_{N-1}^*(G)r(x_0)\|_2,$$

where $r(y^{\text{extr}}) = y^{\text{extr}} - g(y^{\text{extr}})$ and p_{N-1}^* solves

$$p_{N-1}^* = \operatorname{argmin}_{p \in \mathcal{P}_{[N-1]}^{(1)}} \|p(G)r(x_0)\|_2.$$

The results in [Scieur et al., 2016] recalled above handle the case where the operator G is *symmetric*. Bounding $\|p(G)\|_2$ when G is non-symmetric is much more difficult. Fortunately, Crouzeix's conjecture [Crouzeix, 2004] allows us to bound $\|p(G)\|_2$ by solving a Chebyshev problem on the numerical range of G , in the complex plane.

Theorem 3.1 (Crouzeix [2004]). *Let $G \in \mathbb{C}^{n \times n}$, and $p(x) \in \mathbb{C}[x]$, we have*

$$\|p(G)\|_2 \leq c \max_{z \in W(G)} |p(z)|$$

for some absolute constant $c \geq 2$.

Here $W(G) \subset \mathbb{C}$ is the numerical range of the matrix $G \in \mathbb{R}^{n \times n}$, i.e. the range of the Rayleigh quotient

$$W(G) \triangleq \{x^* G x : \|x\|_2 = 1, x \in \mathbb{C}^n\}. \quad (12)$$

[Crouzeix, 2007] shows $c \leq 11.08$ and Crouzeix's conjecture states that this can be further improved to $c = 2$, which is tight. A more recent bound in [Crouzeix and Palencia, 2017] yields $c = 1 + \sqrt{2}$ and there is significant numerical evidence in support of the $c = 2$ conjecture [Greenbaum et al., 2017]. This conjecture has played a vital role in providing convergence results for e.g. the GMRES method [Saad and Schultz, 1986, Choi and Greenbaum, 2015].

Crouzeix's result allows us to turn the problem of finding uniform bounds for the norm of the matrix polynomial $\|p(G)\|_2$ to that of bounding $p(z)$ over the numerical range of G in the complex plane, an arguably much simpler two-dimensional Chebyshev problem.

3.1. Numerical Range Approximations. The previous result links the convergence rate of accelerated algorithms with the optimum value of a Chebyshev problem over the numerical range of the operator G and we now recall classical methods for computing the numerical range. There are no generic tractable methods for computing the exact numerical range of an operator G . However, efficient numerical methods approximate the numerical range based on key structural properties. The Toeplitz-Hausdorff theorem [Hausdorff, 1919, Toeplitz, 1918] in particular states that the numerical range $W(G)$ is a closed convex bounded set. Therefore, it suffices to characterize points on the boundary, the convex hull then yields the numerical range.

Johnson [1978] made the following observations using the properties of the numerical range,

$$\max_{z \in W(G)} \operatorname{Re}(z) = \max_{r \in W(H(G))} r = \lambda_{\max}(H(G)) \quad (13)$$

$$W(e^{i\theta} G) = e^{i\theta} W(G), \quad \forall \theta \in [0, 2\pi), \quad (14)$$

where $\operatorname{Re}(z)$ is the real part of complex number z , $H(G)$ is the Hermitian part of G , i.e. $H(G) = (G + G^*)/2$ and $\lambda_{\max}(H(G))$ is the maximum eigenvalue of $H(G)$. The first property implies that the line parallel to the imaginary axis is tangent to $W(G)$ at $\lambda_{\max}(H(G))$. The second property can be used to determine other tangents via rotations. Using these observations Johnson [1978] showed that the points on the boundary of the numerical range can be characterized as $p_\theta = \{v_\theta^* G v_\theta : \theta \in [0, 2\pi)\}$ where v_θ is the normalized eigenvector corresponding to the largest eigenvalue of the Hermitian matrix

$$H_\theta = \frac{1}{2}(e^{i\theta} G + e^{-i\theta} G^*) \quad (15)$$

The numerical range can thus be characterized as follows.

Theorem 3.2. [Johnson, 1978] *For any $G \in \mathbb{C}^{n \times n}$, we have*

$$W(G) = \operatorname{Co}\{p_\theta : 0 \leq \theta < 2\pi\}$$

where $\operatorname{Co}\{Z\}$ is the convex hull of the set Z .

Note that p_θ cannot be uniquely determined as the eigenvectors v_θ may not be unique but the convex hull above is uniquely determined.

3.2. Chebyshev Bounds & Convergence Rate. Crouzeix's result means that bounding the convergence rate of accelerated algorithms can be achieved by bounding the optimum of the Chebyshev problem

$$\min_{\substack{p \in \mathbb{C}[z] \\ p(1)=1}} \max_{z \in W(G)} |p(z)| \quad (16)$$

where $G \in \mathbb{C}^{n \times n}$. This problem has a trivial answer when the numerical range $W(G)$ is spherical, but the convergence rate can be significantly improved when $W(G)$ is less isotropic.

3.2.1. *Exact Bounds on Ellipsoids.* We can use an outer ellipsoidal approximation of $W(G)$, bounding the optimum value of the Chebyshev problem (16) by

$$\min_{\substack{p(z) \in \mathbb{C}[x] \\ p(1)=1}} \max_{z \in \mathcal{E}_r} |p(z)| \quad (17)$$

where

$$\mathcal{E}_r \triangleq \{z \in \mathbb{C} : |z - 1| + |z + 1| \leq r + 1/r\}. \quad (18)$$

This Chebyshev problem has an explicit solution in certain regimes. As in the real case, we will use $C_n(z)$, the Chebyshev polynomial of degree k . [Fischer and Freund \[1991\]](#) show the following result on the optimal solution to problem (17) on ellipsoids.

Theorem 3.3. [*Fischer and Freund, 1991, Th. 2*] *Let $k \geq 5$, $r > 1$ and $c \in \mathbb{R}$. The polynomial*

$$T_{k,\kappa}(z) = T_k(z)/T_k(1 - \kappa)$$

where

$$T_k(z) = \frac{1}{2} \left(v^k + \frac{1}{v^k} \right), \quad v = \frac{1}{2} \left(z + \frac{1}{z} \right)$$

is the unique solution of problem (17) if either

$$|1 - \kappa| \geq \frac{1}{2} (r^{\sqrt{2}} + r^{-\sqrt{2}})$$

or

$$|1 - \kappa| \geq \frac{1}{2a_r} \left(2a_r^2 - 1 + \sqrt{2a_r^4 - a_r^2 + 1} \right)$$

where $a_r = (r + 1/r)/2$.

The optimal polynomial for a general ellipse \mathcal{E} can be obtained by a simple change of variables. That is, the polynomial $\bar{T}_k(z) = T_k(\frac{c-z}{d})/T_k(\frac{c-1}{d})$ is optimal for the problem (17) over any ellipse \mathcal{E} with center c , focal distance d and semi-major axis a . It can be easily seen that the maximum value is achieved at the point a on the real axis. That is the solution to the min max problem is given by $\bar{T}_k(a)$. Figure 1 shows the surface of the optimal polynomial with degree 5 for $a = 0.8$, $d = 0.76$ and $c = 0$.

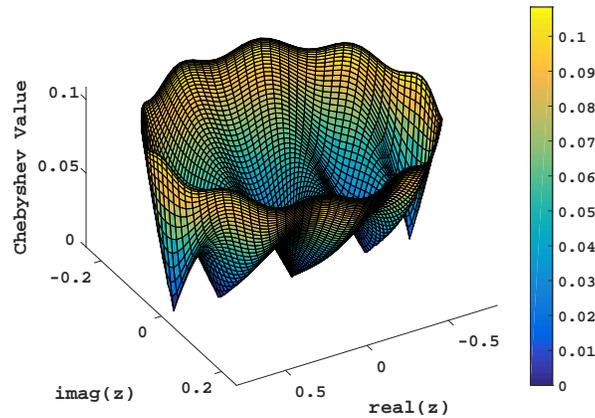


FIGURE 1. Surface of the optimal polynomial $\bar{T}_n(z)$ with degree 5 for $a = 0.8$, $d = 0.76$ and $c = 0$.

Figure 2 shows the solutions to the problem (17) with degree 5 for various ellipses with center at origin, various eccentricity values $e = d/a$ and semi-major axis a . Here, zero eccentricity corresponds to a sphere, while an eccentricity of one corresponds to a line.

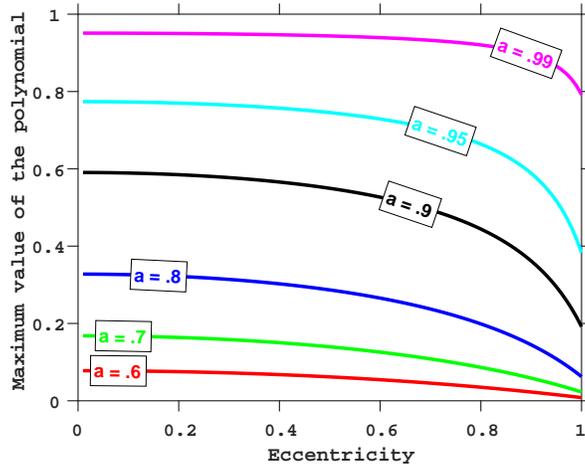


FIGURE 2. Optimal value of the Chebyshev problem (17) for ellipses with centers at origin. Lower values of the maximum of the Chebyshev problem mean faster convergence. The higher the eccentricity here, the faster the convergence.

4. ACCELERATING NON-SYMMETRIC ALGORITHMS

We have seen in the previous section that (asymptotically) controlling the convergence rate of the non-linear acceleration scheme in Algorithm 1 for generic operators G means bounding the optimal value of the Chebyshev optimization problem in (16) over the numerical range of the operator driving iterations near the optimum. In what follows, we explicitly detail this operator and approximate its numerical range for two classical algorithms, Nesterov’s accelerated method [Nesterov, 1983] and Chambolle-Pock’s Primal-Dual Algorithm [Chambolle and Pock, 2011]. We focus on quadratic optimization below. We will see later in Section 5 that, asymptotically at least, the behavior of acceleration on generic problems can be analyzed as a perturbation of the quadratic case.

4.1. Nesterov’s Accelerated Gradient Method. The iterates formed by Nesterov’s accelerated gradient descent method for minimizing smooth strongly convex functions with constant stepsize follow

$$\begin{cases} x_k = y_{k-1} - \alpha \nabla f(y_{k-1}) \\ y_k = x_k + \beta(x_k - x_{k-1}) \end{cases} \quad (19)$$

with $\beta = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}$, where L is the gradient’s Lipschitz continuity constant and μ is the strong convexity parameter. This algorithm is better handled using the results in previous sections, and we only use it here to better illustrate our results on non-symmetric operators.

4.1.1. Nesterov’s Operator in the quadratic case. When minimizing quadratic functions $f(x) = \frac{1}{2} \|Bx - b\|^2$, using constant stepsize $1/L$, these iterations become,

$$\begin{cases} x_k - x^* &= y_{k-1} - x^* - \frac{1}{L} B^T (By_{k-1} - b) \\ y_k - x^* &= x_k - x^* + \beta(x_k - x^* - x_{k-1} + x^*). \end{cases}$$

or again,

$$\begin{bmatrix} x_k - x^* \\ y_k - x^* \end{bmatrix} = \begin{bmatrix} 0 & A \\ -\beta I & (1 + \beta)A \end{bmatrix} \begin{bmatrix} x_{k-1} - x^* \\ y_{k-1} - x^* \end{bmatrix}$$

where $A = I - \frac{1}{L} B^T B$. We write G the *non-symmetric* linear operator in these iterations, i.e.

$$G = \begin{bmatrix} 0 & A \\ -\beta I & (1 + \beta)A \end{bmatrix} \quad (20)$$

The results in Section 2 show that we can accelerate the sequence $z_k = (x_k, y_k)$ if the solution to the minmax problem (16) defined over the numerical range of the operator G is bounded.

4.1.2. *Numerical Range.* We can compute the numerical range of the operator G using the techniques described in Section (2). In the particular case of Nesterov's accelerated gradient method, the numerical range is a convex hull of ellipsoids. We show this by considering the 2×2 operators obtained by replacing the symmetric positive matrix G with its eigenvalues, to form

$$G_j = \begin{bmatrix} 0 & \lambda_j \\ -\beta I & (1 + \beta)\lambda_j \end{bmatrix} \quad \text{for } j \in \{1, 2, \dots, n\} \quad (21)$$

where $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n < 1$ are the eigenvalues of the matrix A . We have the following result.

Theorem 4.1. *The numerical range of operator G is given as the convex hull of the numerical ranges of the operators G_j , i.e. $W(G) = \text{Co}\{W(G_1), W(G_2), \dots, W(G_n)\}$.*

Proof. Let v_1, v_2, \dots, v_n be eigen vectors associated with eigen values $\lambda_1, \lambda_2, \dots, \lambda_n$ of the matrix A . We can write

$$A = \sum_{j=0}^n \lambda_j v_j v_j^T \quad I = \sum_{j=0}^n v_j v_j^T$$

Let $t \in W(G) \subset \mathbb{C}$. By definition of the numerical range, there exists $z \in \mathbb{C}^{2n}$ with $z^* z = 1$ and

$$\begin{aligned} t &= z^* \begin{bmatrix} 0 & A \\ -\beta I & (1 + \beta)A \end{bmatrix} z \\ &= z^* \begin{bmatrix} 0 & \sum_{j=1}^n \lambda_j v_j v_j^T \\ -\beta \sum_{j=1}^n v_j v_j^T & (1 + \beta) \sum_{j=1}^n \lambda_j v_j v_j^T \end{bmatrix} z \\ &= \sum_{j=0}^n z^* \left(\begin{bmatrix} 0 & \lambda_j \\ -\beta & (1 + \beta)\lambda_j \end{bmatrix} \otimes v_j v_j^T \right) \text{vec}([z_1, z_2]) \\ &= \sum_{j=0}^n z^* \text{vec} \left(v_j v_j^T [z_1, z_2] \begin{bmatrix} 0 & \lambda_j \\ -\beta & (1 + \beta)\lambda_j \end{bmatrix}^T \right) \end{aligned}$$

and since $v_j v_j^T v_j v_j^T = v_j v_j^T$, this last term can be written

$$\begin{aligned} t &= \sum_{j=0}^n \text{Tr} \left(v_j v_j^T [z_1, z_2] \begin{bmatrix} 0 & \lambda_j \\ -\beta & (1 + \beta)\lambda_j \end{bmatrix}^T [z_1, z_2]^* v_j v_j^T \right) \\ &= \sum_{j=0}^n \text{Tr}(v_j v_j^T) \left([v_j^T z_1, v_j^T z_2] \begin{bmatrix} 0 & \lambda_j \\ -\beta & (1 + \beta)\lambda_j \end{bmatrix}^T [z_1^* v_j, z_2^* v_j]^T \right) \end{aligned}$$

Now, let $w_j = [z_1^* v_j, z_2^* v_j]^T$, and

$$y_j = \frac{w_j^T G_j w_j}{\|w_j\|_2^2}$$

and by the definition of the numerical range, we have $y_j \in W(G_j)$. Therefore,

$$t = \sum_{j=0}^n \left(\frac{w_j^T G_j w_j}{\|w_j\|_2^2} \right) \|w_j\|_2^2$$

hence

$$t \in \text{Co}(W(G_1), W(G_2), \dots, W(G_n)).$$

We have shown that if $t \in W(G)$ then $t \in \mathbf{Co}(W(G_1), W(G_2), \dots, W(G_n))$. We can show the converse by following the above steps backwards. That is, if $t \in \mathbf{Co}(W(G_1), W(G_2), \dots, W(G_n))$ then we have,

$$t = \sum_{j=0}^n \theta_j \left(\frac{w_j^T G_j w_j}{\|w_j\|_2^2} \right)$$

where $\theta_j > 0$, $\sum_{j=0}^n \theta_j = 1$ and $w_j \in \mathbb{C}^2$. Now, let

$$z = \sum_{j=0}^n \frac{\mathbf{vec}(v_j w_j^T) \theta_j^{1/2}}{\|w_j\|}$$

and we have,

$$t = \sum_{j=0}^n [z_1^* v_j z_2^* v_j] G_j \begin{bmatrix} v_j^T z_1 \\ v_j^T z_2 \end{bmatrix}$$

wherein we used the fact that $v_j^T v_k = 0$ for any $j \neq k$ and $v_j^T v_j = 1$ in computing $w_j^T = [z_1^* v_j z_2^* v_j]$. We also note that $z^* z = 1$ by the definition of z and rewriting the sum in the matrix form we can show that $t \in W(G)$ which completes the proof. ■

To minimize the solution of the Chebyshev problem in (16) and control convergence given the normalization constraint $p(1) = 1$, the point $(1, 0)$ should be outside the numerical range. Because the numerical range is convex and symmetric w.r.t. the real axis (the operator G is real), this means checking if the maximum real value of the numerical range is less than 1.

For 2×2 matrices, the boundary of the numerical range is given by an ellipse [Donoghue, 1957], so the numerical range of Nesterov's accelerated gradient method is the convex hull of ellipsoids. The ellipse in [Donoghue, 1957] can be determined directly from the entries of the matrix as in Johnson [1974], as follows.

Theorem 4.2. [Johnson, 1974] For any real 2 by 2 matrix

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

the boundary of the numerical range is an ellipse whose axes are the line segments joining the points x to y and w to z respectively where,

$$\begin{aligned} x &= \frac{1}{2}(a + d - ((a - d)^2 + (b + c)^2)^{1/2}) \\ w &= \frac{a + d}{2} - i \left| \frac{b - c}{2} \right| \\ y &= \frac{1}{2}(a + d + ((a - d)^2 + (b + c)^2)^{1/2}) \\ z &= \frac{a + d}{2} + i \left| \frac{b - c}{2} \right| \end{aligned}$$

are the points in the complex plane.

This allows us to compute the maximum real value of $W(G)$, as the point of intersection of $W(G)$ with the real line which can be computed explicitly as,

$$\begin{aligned} re(G) &= \max Re(W(G)) = \max_j Re(W(G_j)) \\ &= \frac{1}{2} \left((1 + \beta) \lambda_n + \sqrt{\lambda_n^2 (1 + \beta)^2 + (\lambda_n - \beta)^2} \right) \end{aligned}$$

where $\lambda_n = 1 - \frac{\mu}{L}$.

We observe that $re(G)$ is a function of the condition number of the problem and takes values in the interval $[0, 2]$. Therefore, RNA will only work on Nesterov's accelerated gradient method when $re(G) < 1$

holds, which implies that the condition number of the problem $\kappa = \frac{L}{\mu}$ should be less than around 2.5 which is highly restrictive.

An alternative approach is to use RNA on a sequence of iterates sampled every few iterations, which is equivalent to using powers of the operator G . We expect the numerical radius of some power of operator G to be less than 1 for any conditioning of the problem. This is because the iterates are converging at an R -linear rate and so the norm of the power of the operator is decreasing at an R -linear rate with the powers. Therefore, using the property that the numerical radius is bounded by the norm of the operator we have,

$$re(G^p) = \max Re(W(G^p)) \leq r_{G^p} \leq \|G^p\| \leq C_p \rho^p$$

where r_{G^p} is the numerical radius of G^p . Figure 3 shows the numerical range of the powers of the operator G for a random matrix $B^T B$ with dimension $d = 50$. We observe that after some threshold value for the power p , $(1, 0)$ lies outside the field values corresponding to G^p thus guaranteeing that the acceleration scheme will work. We also observe that the boundaries of the field values are almost circular for higher powers p , which is consistent with results on optimal matrices in [Lewis and Overton, 2018]. When the numerical range is circular, the solution of the Chebyshev problem is trivially equal to z^p so RNA simply picks the last iterate and does not accelerate convergence.

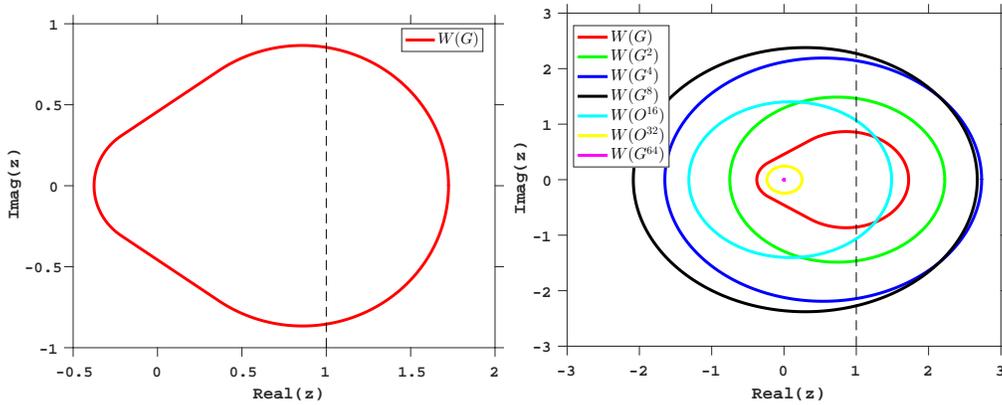


FIGURE 3. Numerical range for the linear operator in Nesterov’s method, on a random quadratic problem with dimension 50. Left: Operator G . Right: Various operator powers G^p . The RNA scheme will improve convergence whenever the point $(1, 0)$ lies outside of the numerical range of the operator.

The difficulty in performing RNA on Nesterov’s accelerated gradient method arises from the fact that the iterates can be non-monotonic. The restriction that 1 should be outside the numerical range is necessary for both non-symmetric and symmetric operators. In symmetric operators, the numerical range is a line segment on the real axis and the numerical radius and spectral radius are equal, so this restriction is equivalent to having spectral radius less than 1, i.e. having monotonically converging iterates.

4.2. Chambolle-Pock’s Primal-Dual Algorithm. Chambolle-Pock is a first-order primal-dual algorithm used for minimizing composite functions of the form

$$\min_x h_p(x) := f(Ax) + g(x) \tag{22}$$

where f and g are convex functions and A is a continuous linear map. Optimization problems of this form arise in e.g. imaging applications like total variation minimization (see Chambolle and Pock [2016]). The Fenchel dual of this problem is given by

$$\max_y h_d(y) := -f^*(-y) - g^*(A^*y) \tag{23}$$

where f^*, g^* are the convex conjugate functions of f, g respectively. These problems are primal dual formulations of the general saddle point problem,

$$\min_x \max_y \langle Ax, y \rangle + g(x) - f^*(y), \quad (24)$$

where f^*, g are closed proper functions. [Chambolle and Pock \[2011\]](#) designed a first-order primal-dual algorithm for solving these problems, where primal-dual iterates are given by

$$\begin{cases} y_{k+1} = \mathbf{Prox}_{f^*}^\sigma(y_k + \sigma A\bar{x}_k) \\ x_{k+1} = \mathbf{Prox}_g^\tau(x_k - \tau A^* y_{k+1}) \\ \bar{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k) \end{cases} \quad (25)$$

where σ, τ are the step length parameters, $\theta \in [0, 1]$ is the momentum parameter and the proximal mapping of a function f is defined as

$$\mathbf{Prox}_f^\tau(y) = \arg \min_x \{ \|y - x\|^2 / (2\tau) + f(x) \}$$

Note that if the proximal mapping of a function is available then the proximal mapping of the conjugate of the function can be easily computed using Moreau's identity, with

$$\mathbf{Prox}_f^\tau(y) + \mathbf{Prox}_{f^*}^{1/\tau}(y/\tau) = y$$

The optimal strategy for choosing the step length parameters σ, τ and the momentum parameter θ depend on the smoothness and strong convexity parameters of the problem. When f^* and g are strongly convex with strong convexity parameters δ and γ respectively then these parameters are chosen to be constant values given as

$$\sigma = \frac{1}{\|A\|} \sqrt{\frac{\gamma}{\delta}} \quad \tau = \frac{1}{\|A\|} \sqrt{\frac{\delta}{\gamma}} \quad \theta = \left(1 + \frac{2\sqrt{\gamma\delta}}{\|A\|} \right)^{-1} \quad (26)$$

to yield the optimal linear rate of convergence. When only one of f^* or g is strongly convex with strong convexity parameter γ , then these parameters are chosen adaptively at each iteration as

$$\theta_k = (1 + 2\gamma\tau_k)^{-1/2} \quad \sigma_{k+1} = \sigma_k / \theta_k \quad \tau_{k+1} = \tau_k \theta_k \quad (27)$$

to yield the optimal sublinear rate of convergence.

A special case of the primal-dual algorithm with no momentum term, i.e., $\theta = 0$ in (25) is also known as the Arrow-Hurwicz method ([Mizoguchi \[1960\]](#)). Although theoretical complexity bounds for this algorithm are worse compared to methods including a momentum term, it is observed experimentally that the performance is either on par or sometimes better, when step length parameters are chosen as above.

We first consider algorithms with no momentum term and apply RNA to the primal-dual sequence $z_k = (y_k, x_k)$. We note that, as observed in the Nesterov's case, RNA can only be applied on non-symmetric operators for which the normalization constant 1 is outside their numerical range. Therefore, the step length parameters τ, σ should be suitably chosen such that this condition is satisfied.

4.2.1. Chambolle-Pock's Operator in the Quadratic Case. When minimizing smooth strongly convex quadratic functions where $f(Ax) = \frac{1}{2}\|Ax - b\|^2$ and $g(x) = \frac{\mu}{2}\|x\|^2$, the proximal operators have closed form solutions. That is

$$\mathbf{Prox}_{f^*}^\sigma(y) = \frac{y - \sigma b}{1 + \sigma} \quad \text{and} \quad \mathbf{Prox}_g^\tau(x) = \frac{x}{1 + \tau\mu}.$$

Iterates of the primal-dual algorithm with no momentum term can be written as,

$$y_{k+1} = \frac{y_k + \sigma Ax_k - \sigma b}{1 + \sigma}, \quad x_{k+1} = \frac{x_k - \tau A^T y_{k+1}}{1 + \tau\mu}$$

Note that the optimal primal and dual solutions satisfy $y^* = Ax^* - b$ and $x^* = \frac{-1}{\mu}A^T y$. This yields the following operator for iterations

$$G = \begin{bmatrix} \frac{I}{1+\sigma} & \frac{\sigma A}{1+\sigma} \\ \frac{\tau A^T}{(1+\sigma)(1+\tau\mu)} & \frac{I}{1+\tau\mu} - \frac{\tau\sigma A^T A}{(1+\sigma)(1+\tau\mu)} \end{bmatrix} \quad (28)$$

Note that G is a non-symmetric operator except when $\sigma = \frac{\tau}{1+\tau\mu}$, in which case the numerical range is a line segment on the real axis and the spectral radius is equal to the numerical radius.

4.2.2. Numerical Range. The numerical range of the operator can be computed using the techniques described in Section 2. As mentioned earlier, the point 1 should be outside the numerical range for the Chebyshev polynomial to be bounded. Therefore, using (13), we have, $re(G) = \max Re(W(G)) = \lambda_{max}(\frac{G+G^*}{2})$. The step length parameters σ, τ should be chosen such that the above condition is satisfied. We observe empirically that there exists a range of values for the step length parameters such that $re(G) < 1$. Figure 4 shows the numerical range of operator G for $\sigma = 4, \tau = 1/\|A^T A\|$ with two different regularization constants and Figure 5 shows the regions for which $re(G^p) \leq 1$ (converging) for different values of σ and τ .

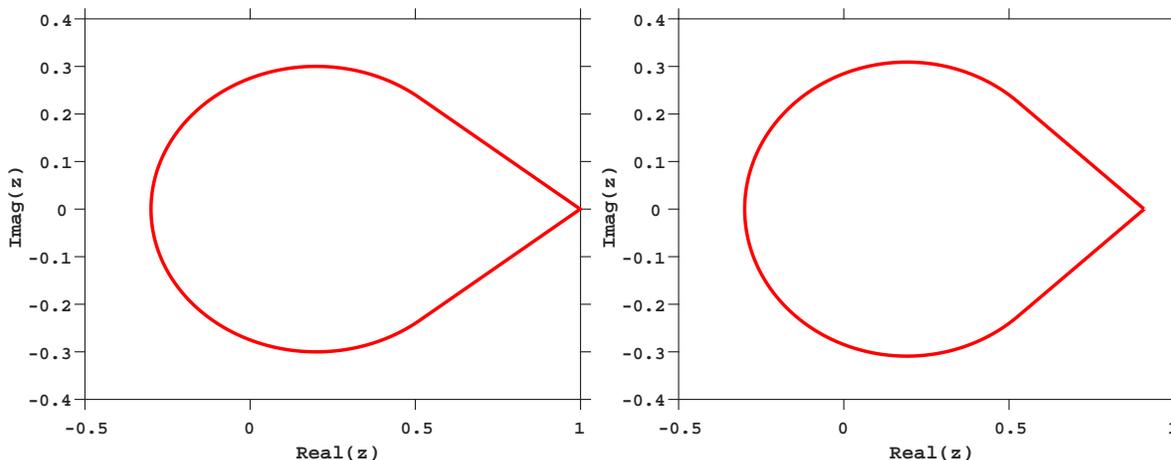


FIGURE 4. Field values for the Sonar dataset [Gorman and Sejnowski, 1988] with $\sigma = 4, \tau = 1/\|A^T A\|$. The dataset has been scaled such that $\|A^T A\| = 1$. Left: $\mu = 10^{-3}$, right: $\mu = 10^{-1}$. The smaller numerical range on the right means faster convergence.

5. RNA ON NONLINEAR ITERATIONS

In previous sections, we analyzed the rate of convergence of RNA on linear algorithms (or quadratic optimization problems). In practice however, the operator g is not linear, but can instead be nonlinear with potentially random perturbation. In this situation, regularizing parameter ensures RNA converges [Scieur et al., 2016].

In this section, we first introduce the CNA algorithm, a constrained version of RNA that explicitly bounds the norm of the coefficients c for the linear combinations. We show its equivalence with the RNA algorithm. Then, we analyze the rate of convergence of CNA when g is a linear function perturbed with arbitrary errors, whose origin can be nonlinearities and/or random noises.

5.1. Constrained Nonlinear Acceleration. We now introduce the constrained version of RNA, replacing the regularization term by the hard constraint

$$\|c\|_2 \leq \frac{1+\tau}{\sqrt{N}}.$$

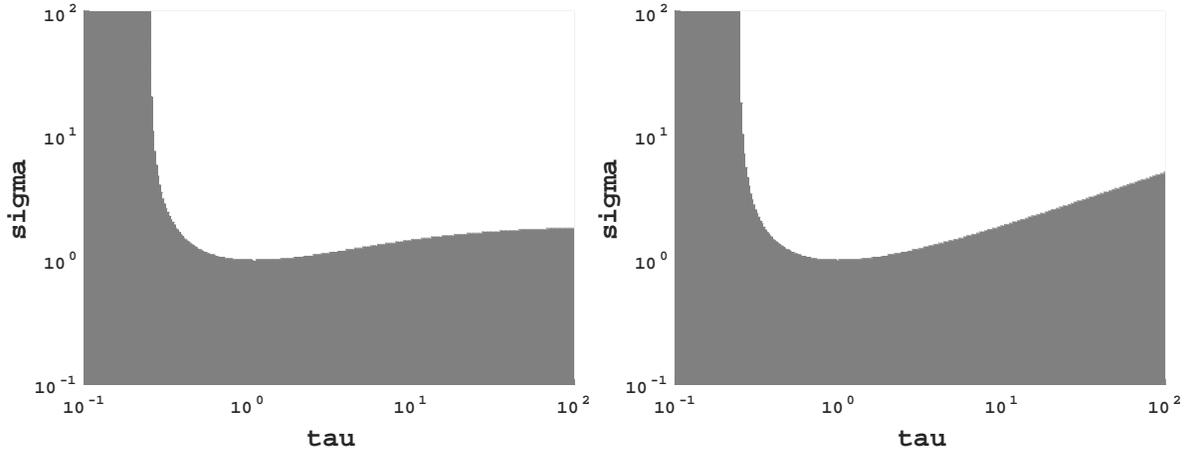


FIGURE 5. Plot of the $re(G^p) = 1$ frontier with degree $p = 5$ for the Sonar dataset [Gorman and Sejnowski, 1988] for different values of τ and σ . White color represents values for which $re(G^p) \leq 1$ (converging) and black color represents values $re(G^p) > 1$ (not converging). Left: $\mu = 10^{-3}$. Right: $\mu = 10^{-1}$.

In this algorithm, the parameter $\tau > 0$ controls the norm of the coefficients c . Of course, all the previous analysis applies to CNA, as RNA with $\lambda = 0$ is exactly CNA with $\tau = \infty$.

Algorithm 2 Constrained Nonlinear Acceleration (CNA)

Data: Matrices X and Y of size $d \times N$ constructed from the iterates as in (2) and (3).

Parameters: Mixing $\eta \neq 0$, constraint $\tau \geq 0$.

1. Compute matrix of residuals $R = X - Y$.
2. Solve

$$c^{(\tau)} = \operatorname{argmin}_{c: c^T \mathbf{1} = 1} \|Rc\|_2 \quad \text{s.t.} \quad \|c\|_2 \leq \frac{1+\tau}{\sqrt{N}} \quad (29)$$

3. Compute extrapolated solution $y^{\text{extr}} = (Y - \eta R)c^{(\tau)}$.
-

5.2. Equivalence Between Constrained & Regularized Nonlinear Acceleration. The parameters λ in Algorithm 1 and τ in Algorithm 2 play similar roles. High values of λ give coefficients close to simple averaging, and $\lambda = 0$ retrieves Anderson Acceleration. We have the same behavior when $\tau = 0$ or $\tau = \infty$. We can jump from one algorithm to the other using dual variables, since (10) is the Lagrangian relaxation of the convex problem (29). This means that, for all values of τ there exists $\lambda = \lambda(\tau)$ that achieves $c^\lambda = c^{(\tau)}$. In fact, we can retrieve τ from the solution c^λ by solving

$$\frac{1+\tau}{\sqrt{N}} = \|c^\lambda\|_2.$$

Conversely, to retrieve λ from $c^{(\tau)}$, it suffices to solve

$$\left\| \frac{(R^T R + (\lambda \|R\|_2^2) I)^{-1} \mathbf{1}_N}{\mathbf{1}_N^T (R^T R + (\lambda \|R\|_2^2) I)^{-1} \mathbf{1}_N} \right\|^2 = \frac{(1+\tau)^2}{N}, \quad (30)$$

assuming the constraint in (29) tight, otherwise $\lambda = 0$. Because the norm in (30) is increasing with λ , a binary search or one-dimensional Newton methods gives the solution in a few iterations.

The next proposition bounds the norm of the coefficients of Algorithm 1 with an expression similar to (29).

Proposition 5.1. *The norm of c^λ from (10) is bounded by*

$$\|c^\lambda\|_2 \leq \frac{1}{\sqrt{N}} \sqrt{1 + \frac{1}{\lambda}} \quad (31)$$

Proof. See Scieur et al. [2016], (Proposition 3.2). ■

Having established the equivalence between constrained and regularized nonlinear acceleration, the next section discusses the rate of convergence of CNA in the presence of perturbations.

5.3. Constrained Chebyshev Polynomial. The previous results consider the special cases where $\lambda = 0$ or $\tau = 0$, which means that $\|c\|$ is unbounded. However, Scieur et al. [2016] show instability issues when $\|c\|$ is not controlled. Regularization is thus required in practice to make the method more robust to perturbations, even in the quadratic case (e.g., round-off errors). Unfortunately, this section will show that robustness comes at the cost of a potentially slower rate of convergence.

We first introduce *constrained Chebyshev polynomials* for the range of a specific matrix. Earlier work in [Scieur et al., 2016] considered regularized Chebyshev polynomials, but using a constrained formulation significantly simplifies the convergence analysis here. This polynomial plays an important role in Section 5.4 in the convergence analysis.

Definition 5.2. *The Constrained Chebyshev Polynomial $\mathcal{T}_N^{\tau,G}(x)$ of degree N solves, for $\tau \geq 0$,*

$$\mathcal{T}_N^{\tau,G}(x) \triangleq \underset{p \in \mathcal{P}_{[N]}^{(1)}}{\operatorname{argmin}} \max_{x \in W(G)} p(x) \quad \text{s.t. } \|p\|_2 \leq \frac{1+\tau}{\sqrt{1+N}} \quad (32)$$

in the variable $p \in \mathcal{P}_{[N]}^{(1)}$, where $W(G)$ is the numerical range of G . We write $\mathcal{C}_N^{\tau,G} \triangleq \|\mathcal{T}_N^{\tau,G}(G)\|_2$ the norm of the polynomial $\mathcal{T}_N^{\tau,G}$ applied to the matrix G .

5.4. Convergence Rate of CNA without perturbations. The previous section introduced constrained Chebyshev polynomials, which play an essential role in our convergence results when g is nonlinear and/or iterates (2) are noisy. Instead of analyzing Algorithm 1 directly, we focus on its constrained counterpart, Algorithm 2.

Proposition 5.3. *Let X, Y (3) be build using iterates from (2) where g is linear (7) does not have 1 as eigenvalue. Then, the norm of the residual (6) of the extrapolation produced by Algorithm 2 is bounded by*

$$\|r(y^{\text{extr}})\|_2 \leq \|I - \eta(G - I)\|_2 \|r(x_0)\|_2 \mathcal{C}_{N-1}^{\tau,G}, \quad (33)$$

where $\tau \geq 0$ and $\mathcal{C}_N^{\tau,G}$ is defined in (32).

Proof. The proof is similar to the one of Theorem 2.2. It suffices to use the constrained Chebyshev polynomial rather than the rescaled Chebyshev polynomial from Golub and Varga [1961]. ■

Proposition 5.3 with $\tau = \infty$ gives the same result than Theorem 2.2. However, smaller values of τ give weaker results as $\mathcal{C}_{N-1}^{\tau,G}$ increases. However, smaller values of τ also reduce the norm of coefficients $c^{(\tau)}$ (29), which makes the algorithm more robust to noise.

Using the constrained algorithm in the context of non-perturbed linear function g yields no theoretical benefit, but the bounds on the extrapolated coefficients simplify the analysis of perturbed non-linear optimization schemes as we will see below.

In this section, we analyze the convergence rate of Algorithm 2 for simplicity, but the results also hold for Algorithm 1. We first introduce the concept of perturbed linear iteration, then we analyze the convergence rate of RNA in this setting.

Perturbed Linear Iterations. Consider the following perturbed scheme,

$$\tilde{X}_i = X^* + G(\tilde{Y}_{i-1} - X^*) + E_i, \quad \tilde{Y}_i = [x_0, \tilde{X}_i]L_i, \quad (34)$$

where \tilde{X}_i and \tilde{Y}_i are formed as in (3) using the perturbed iterates \tilde{x}_i and \tilde{y}_i , and L_i is constructed using (5), and we write $E_i = [e_1, e_2, \dots, e_i]$. For now, we do not assume anything on e_i or E_i . This class

contains many schemes such as gradient descent on nonlinear functions, stochastic gradient descent or even Nesterov’s fast gradient with backtracking line search for example.

The notation (34) makes the analysis simpler than in [Scieur et al., 2016, 2017a], as we have the explicit form of the error over time. Consider the perturbation matrix P_i ,

$$P_i \triangleq \tilde{R}_i - R_i, \quad (35)$$

Proposition 5.4 shows that the magnitude of the perturbations $\|P_i\|$ is proportional to the noise matrix $\|E_i\|$, i.e., $\|P_i\| = O(\|E_i\|)$.

Proposition 5.4. *Let P_i be defined in (35), where $(\tilde{X}_i, \tilde{Y}_i)$ and $(\tilde{X}_i, \tilde{Y}_i)$ are formed respectively by (4) and (34). Let $\bar{L}_j = \|L_1\|_2 \|L_2\|_2 \dots \|L_j\|_2$. Then, we have the following bound*

$$\|P_i\| \leq 2\|E_i\| \bar{L}_i \sum_{j=1}^i \|G\|^j.$$

Proof. First, we start with the definition of R and \tilde{R} . Indeed,

$$\tilde{R}_i - R_i = \tilde{X}_i - X_i - (\tilde{Y}_{i-1} - Y_{i-1}).$$

By definition,

$$\tilde{X}_i - X_i = G(\tilde{Y}_{i-1} - X^*) + X^* + E_i - G(Y_{i-1} - X^*) - X^* = G(\tilde{Y}_{i-1} - Y_{i-1}) + E_i$$

On the other side,

$$\tilde{Y}_{i-1} - Y_{i-1} = [0; \tilde{X}_{i-1} - X_{i-1}]L_{i-1}$$

We thus have

$$\begin{aligned} P_i &= \tilde{X}_i - X_i - (\tilde{Y}_{i-1} - Y_{i-1}), \\ &= G(\tilde{Y}_{i-1} - Y_{i-1}) + E_i - [0; \tilde{X}_{i-1} - X_{i-1}]L_{i-1}, \\ &= G([0; \tilde{X}_{i-1} - X_{i-1}]L_{i-1}) + E_i - [0; G(\tilde{Y}_{i-2} - Y_{i-2}) + E_{i-1}]L_{i-1}, \\ &= G[0; P_{i-1}]L_{i-1} + E_i - [0; E_{i-1}]L_{i-1}. \end{aligned}$$

Finally, knowing that $\|E_i\| \geq \|E_{i-1}\|$ and $\|L_i\| \geq 1$, we expand

$$\begin{aligned} \|P_i\| &= \|G\| \|P_{i-1}\| \|L_{i-1}\| + \|E_i\| + \|E_{i-1}\| \|L_{i-1}\| \\ &\leq \|G\| \|P_{i-1}\| \|L_{i-1}\| + 2\|E_i\| \|L_{i-1}\| \end{aligned}$$

to have the desired result. ■

We now analyze how close the output of Algorithm 2 is to x^* . To do so, we compare scheme (34) to its perturbation-free counterpart (4). Both schemes have the same starting point x_0 and “fixed point” x^* . It is important to note that scheme (34) may not converge due to noise. The next theorem bounds the accuracy of the output of CNA.

Theorem 5.5. *Let y^{extr} be the output of Algorithm (2) applied to (34). Its accuracy is bounded by*

$$\|(G - I)(y^{extr} - x^*)\| \leq \|I - \eta(G - I)\| \left(\underbrace{C_{N-1}^{\tau, G} \|(G - I)(x_0 - x^*)\|}_{\text{acceleration}} + \underbrace{\frac{1+\tau}{\sqrt{N}} (\|P_N\| + \|E_N\|)}_{\text{stability}} \right).$$

Proof. We start with the following expression for arbitrary coefficients c that sum to one,

$$(G - I) \left((\tilde{Y} - \eta\tilde{R})c - x^* \right).$$

Since

$$\tilde{R} = \tilde{X} - \tilde{Y} = (G - I)(\tilde{Y} - X^*) + E,$$

we have

$$(G - I)(\tilde{Y} - X^*) = (\tilde{R} - E).$$

So,

$$(G - I)(\tilde{Y} - X^* - \eta\tilde{R})c = (\tilde{R} - E)c - \eta(G - I)\tilde{R}c.$$

After rearranging the terms we get

$$(G - I) \left((\tilde{Y} - \eta\tilde{R})c - x^* \right) = (I - \eta(G - I))\tilde{R}c - Ec. \quad (36)$$

We bound (36) as follow, using coefficients from (29),

$$\|I - \eta(G - I)\| \|\tilde{R}c^{(\tau)}\| + \|E\| \|c^{(\tau)}\|.$$

Indeed,

$$\|\tilde{R}c^{(\tau)}\|^2 = \min_{c: c^T \mathbf{1} = 1, \|c\| \leq \frac{1+\tau}{\sqrt{N}}} \|\tilde{R}c\|^2.$$

We have

$$\begin{aligned} \min_{c: \mathbf{1}^T c = 1, \|c\| \leq \frac{1+\tau}{\sqrt{N}}} \|\tilde{R}c\|_2, &\leq \min_{c: \mathbf{1}^T c = 1, \|c\| \leq \frac{1+\tau}{\sqrt{N}}} \|Rc\|_2 + \|P_R c\|_2, \\ &\leq \left(\min_{c: \mathbf{1}^T c = 1, \|c\| \leq \frac{1+\tau}{\sqrt{N}}} \|Rc\|_2 \right) + \|P_R\|_2 \frac{1+\tau}{\sqrt{N}}, \\ &\leq \mathcal{C}_{N-1}^{\tau, G} \|r(x_0)\| + \frac{\|P_R\|(1+\tau)}{\sqrt{N}}. \end{aligned}$$

This prove the desired result. ■

This theorem shows that Algorithm 2 balances acceleration and robustness. The result bounds the accuracy by the sum of an *acceleration term* bounded using constrained Chebyshev polynomials, and a *stability term* proportional to the norm of perturbations. In the next section, we consider the particular case where g corresponds to a gradient step when the perturbations are Gaussian or due to non-linearities.

6. CONVERGENCE RATES FOR CNA ON GRADIENT DESCENT

We now apply our results when g in (4) corresponds to the gradient step

$$x - h\nabla f(x), \quad (37)$$

where f is the objective function and h a step size. We assume the function f twice differentiable, L -smooth and μ -strongly convex. This means

$$\mu I \leq \nabla^2 f(x) \leq LI. \quad (38)$$

Also, we assume $h = \frac{1}{L}$ for simplicity. Since we consider optimization of differentiable functions here, the matrix $\nabla^2 f(x^*)$ is symmetric.

When we apply the gradient method (37), we first consider its linear approximation

$$g(x) = x - h\nabla^2 f(x^*)(x - x^*). \quad (39)$$

with stepsize $h = 1/L$. We identify the matrix G in (7) to be

$$G = I - \frac{\nabla^2 f(x^*)}{L}.$$

In this case, and because the Hessian is now symmetric, the numerical range $W(G)$ simplifies into the line segment

$$W(G) = [0, 1 - \kappa],$$

where $\kappa = \frac{\mu}{L} < 1$ often refers to the inverse of the condition number of the matrix $\nabla^2 f(x^*)$.

In the next sections, we study two different cases. First, we assume the objective quadratic, but (39) is corrupted by a random noise. Then, we consider a general nonlinear function f , with the additional assumption that its Hessian is Lipchitz-continuous. This corresponds to a nonlinear, deterministic perturbation of (39), whose noise is bounded by $O(\|x - x^*\|^2)$.

6.1. Random Perturbations. We perform a gradient step on the quadratic form

$$f(x) = \frac{1}{2}(x - x^*)A(x - x^*), \quad \mu I \preceq A \preceq LI.$$

This corresponds to (39) with $\nabla f(x^*) = A$. However, each iteration is corrupted by e_i , where e_i is Gaussian with variance σ^2 . The next proposition is the application of Theorem 5.5 to our setting. To simplify results, we consider $\eta = 1$.

Proposition 6.1. *Assume we use Algorithm (2) with $\eta = 1$ on N iterates from (34), where g is the gradient step (37) and e_i are zero-mean independent random noise with variance bounded by σ^2 . Then,*

$$\mathbb{E}[\|\nabla f(y^{\text{extr}})\|] \leq (1 - \kappa) \mathcal{C}_{N-1}^{\tau, G} \|\nabla f(x_0)\| + \mathcal{E}, \quad (40)$$

where

$$\mathcal{E} \leq (1 - \kappa) \frac{1 + \tau}{\sqrt{N}} L \sigma \sum_{j=1}^N (1 - \kappa)^j \bar{L}_j.$$

In the simple case where we accelerate the gradient descent algorithm, all $L_i = I$ and thus

$$\mathcal{E} \leq \frac{1 + \tau}{\sqrt{N}} \frac{L \sigma}{\kappa}.$$

Proof. Since $\eta = 1$,

$$\|I - \eta(G - I)\| = \|G\| \leq 1 - \kappa.$$

Now, consider $\mathbb{E}[\|E\|]$. Because each e_i are independent Gaussian noise with variance bounded by σ , we have,

$$\mathbb{E}[\|E\|] \leq \sqrt{\mathbb{E}[\|E\|^2]} \leq \sigma.$$

Similarly, for P (35), we use Proposition (5.4) and we have

$$\begin{aligned} \mathbb{E}[\|P\|] &\leq \mathbb{E}[\|E_i\|] \left(1 + \sum_{j=1}^i (1 - \kappa)^j \bar{L}_j\right) \\ &\leq \sigma \left(1 + \sum_{j=1}^i (1 - \kappa)^j \bar{L}_j\right) \end{aligned}$$

Thus, $\mathcal{E}_N^{\kappa, \tau}$ in Theorem 5.5 becomes

$$\mathcal{E}_N^{\kappa, \tau} \leq \frac{\sigma(1 + \tau)}{\sqrt{N}} \left(2 + \sum_{j=1}^N (1 - \kappa)^j \bar{L}_j\right)$$

Finally, it suffice to see that

$$(G - I)(x - x^*) + x^* = (A/L)(x - x^*) + x^* = \frac{1}{L} \nabla f(x),$$

and we get the desired result. In the special case of plain gradient method, $L_i = I$ so $\bar{L}_i = 1$. We then get

$$\sum_{j=1}^N (1 - \kappa)^j \leq \sum_{j=1}^{\infty} (1 - \kappa)^j \leq \frac{1}{\kappa}.$$

which is the desired result. ■

This proposition also applies to gradient descent with momentum or with our online acceleration algorithm (48). We can distinguish two different regimes when accelerating gradient descent with noise. One when σ is small compared to $\|f(x_0)\|$, and one when σ is large. In the first case, the acceleration term dominates. In this case, Algorithm 2 with large τ produces output y^{extr} that converges with a near-optimal rate of convergence. In the second regime where the noise dominates, τ should be close to zero. In this case, using our extrapolation method when perturbation are high naturally gives the simple averaging scheme. We can thus see Algorithm (2) as a way to interpolate optimal acceleration with averaging.

6.2. Nonlinear Perturbations. Here, we study the general case where the perturbation e_i are bounded by a function of D , where D satisfies

$$\|\tilde{y}_i - x^*\|_2 \leq D \quad \forall i. \quad (41)$$

This assumption is usually met when we accelerate non-divergent algorithms. More precisely, we assume the perturbation are bounded by

$$(\|I - \eta(G - I)\| \|P_N\| + \|E\|) \leq \gamma \sqrt{N} D^\alpha. \quad (42)$$

where γ and α are scalar. Since $\|P_N\| = O(\|E\|)$ by proposition 5.4, we have that

$$\|e_i\| \leq O(D^\alpha) \Rightarrow (42). \quad (43)$$

We call these perturbations "nonlinear" because the error term typically corresponds to the difference between g and its linearization around x^* . For example, the optimization of smooth non-quadratic functions with gradient descent can be described using (42) with $\alpha = 1$ or $\alpha = 2$, as shown in Section 6.3. The next proposition bounds the accuracy of the extrapolation produced by Algorithm (2) in the presence of such perturbation.

Proposition 6.2. *Consider Algorithm (2) with $\eta = 1$ on N iterates from (34), where perturbations satisfy (41). Then,*

$$\|(G - I)(y^{\text{extr}} - x^*)\| \leq (1 - \kappa) \left(\mathcal{C}_{N-1}^{\tau, G} \|(G - I)(x_0 - x^*)\| + \mathcal{E} \right)$$

where $\mathcal{E} \leq (1 + \tau)\gamma D^\alpha$.

Proof. Combine Theorem 5.5 with assumption (42). ■

Here, $\|x_0 - x^*\|$ is of the order of D . This bound is generic as does not consider any strong structural assumption on g , only that its first-order approximation error is bounded by a power of D . We did not even assume that scheme (34) converges. This explains why Proposition 6.2 does not necessary give a convergent bound. Nevertheless, in the case of convergent scheme, Algorithm 2 with $\tau = 0$ output the average of previous iterates, that also converge to x^* .

However, Proposition 6.2 is interesting when perturbations are small compared to $\|x_0 - x^*\|$. In particular, it is possible to link τ and D^α so that Algorithm 2 asymptotically reach an optimal rate of convergence, when $D \rightarrow 0$.

Proposition 6.3. *If $\tau = O(D^{-s})$ with $0 < s < \alpha - 1$, then, when $D \rightarrow 0$, Proposition 6.2 becomes*

$$\|(G - I)(y^{\text{extr}} - x^*)\| \leq (1 - \kappa) \left(\frac{1 - \sqrt{\kappa}}{1 + \sqrt{\kappa}} \right)^{N-1} \|(G - I)(x_0 - x^*)\|$$

The same result holds with Algorithm 1 if $\lambda = O(D^r)$ with $0 < r < 2(\alpha - 1)$.

Proof. By assumption,

$$\|x_0 - x^*\| = O(D).$$

We thus have, by Proposition 6.2

$$\|(G - I)(y^{\text{extr}} - x^*)\| \leq (1 - \kappa) \left(\mathcal{C}_{N-1}^{\tau, G} O(D) + (1 + \tau) O(D^\alpha) \right).$$

τ will be a function of D , in particular $\tau = D^{-s}$. We want to have the following conditions,

$$\lim_{D \rightarrow 0} (1 + \tau(D)) D^{\alpha-1} = 0, \quad \lim_{D \rightarrow 0} \tau = \inf.$$

The first condition ensures that the perturbation converge faster to zero than the acceleration term. The second condition ask τ to grow as D decreases, so that CNA becomes unconstrained. Since $\tau = D^{-s}$, we have to solve

$$\lim_{D \rightarrow 0} D^{\alpha-1} + D^{\alpha-s-1} = 0, \quad \lim_{D \rightarrow 0} D^{-s} = \inf.$$

Clearly, $0 < s < \alpha - 1$ satisfies the two conditions. After taking the limit, we obtain

$$\|(G - I)(y^{\text{extr}} - x^*)\| \leq (1 - \kappa) \mathcal{C}_{N-1}^{\tau, G} \|(G - I)(x_0 - x^*)\|$$

Since $W(G)$ is the real line segment $[0, 1 - \kappa]$, and because $\tau \rightarrow \infty$, we end with an unconstrained minimax polynomial. Therefore, we can use the result from [Golub and Varga \[1961\]](#),

$$\min_{p \in \mathcal{P}_{[N]}^{(1)}} \max_{\lambda \in [0, 1 - \kappa]} |p(\lambda)| \leq \left(\frac{1 - \sqrt{\kappa}}{1 + \sqrt{\kappa}} \right)^{N-1}.$$

For the second result, by using (31),

$$\|c^\lambda\|_2 \leq \frac{1}{\sqrt{N}} \sqrt{1 + \frac{1}{\lambda}}.$$

Setting

$$\frac{1 + \tau}{\sqrt{N}} = \frac{1}{\sqrt{N}} \sqrt{1 + \frac{1}{\lambda}}$$

with $\tau = D^{-s}$ gives the conditions. ■

This proposition shows that, when perturbations are of the order of D^α with $\alpha > 1$, then our extrapolation algorithm converges optimally once the \tilde{y}_i are close to the solution x^* . The next section shows this holds, for example, when minimizing function with smooth gradients.

6.3. Optimization of Smooth Functions. Let the objective function f be a nonlinear function that follows (38), which also has a Lipchitz-continuous Hessian with constant M ,

$$\|\nabla^2 f(y) - \nabla^2 f(x)\| \leq M \|y - x\|. \quad (44)$$

This assumption is common in the convergence analysis of second-order methods. For the convergence analysis, we consider that $g(x)$ perform a gradient step on the quadratic function

$$\frac{1}{2}(x - x^*) \nabla^2 f(x^*) (x - x^*). \quad (45)$$

This is the quadratic approximation of f around x^* . The gradient step thus reads, if we set $h = 1/L$,

$$g(x) = \left(I - \frac{\nabla^2 f(x^*)}{L} \right) (x - x^*) + x^*. \quad (46)$$

The perturbed scheme corresponds to the application of (46) with a specific nonlinear perturbation,

$$\tilde{x}_{i+1} = g(\tilde{y}_i) - \underbrace{\frac{1}{L}(\nabla f(\tilde{y}_i) - \nabla^2 f(x^*)(\tilde{y}_i - x^*))}_{=e_i}. \quad (47)$$

This way, we recover the gradient step on the non-quadratic function f . The next Proposition shows that schemes (47) satisfies (42) with $\alpha = 1$ when D is big, or $\alpha = 2$ when D is small.

Proposition 6.4. *Consider the scheme (47), where f satisfies (44). If $\|y_i - x^*\| \leq D$, then (43) holds with $\alpha = 1$ for large D or $\alpha = 2$ for small D , i.e.,*

$$\|e_i\| = \left\| \frac{1}{L}(\nabla f(\tilde{y}_i) - \nabla^2 f(x^*)(\tilde{y}_i - x^*)) \right\| \leq \min\{\|y_i - x^*\|, \frac{M}{2L}\|y_i - x^*\|^2\} \leq \min\{D, \frac{M}{2L}D^2\}.$$

Proof. The proof of this statement can be found in [Nesterov and Polyak \[2006\]](#). ■

The combination of Proposition 6.3 with Proposition 6.4 means that RNA (or CNA) converges asymptotically when λ (or τ) are set properly. In other words, if λ decreases a little bit faster than the perturbations, the extrapolation on the perturbed iterations behave as if it was accelerating a perturbation-free scheme. Our result improves that in [Scieur et al. \[2016, 2017a\]](#), where $r \in]0, \frac{2(\alpha-1)}{3}[$.

7. ONLINE ACCELERATION

We now discuss the convergence of online acceleration, i.e. coupling iterates in g with the extrapolation Algorithm 1 at each iteration when $\lambda = 0$. The iterates are now given by

$$x_N = g(y_{N-1}), \quad y_N = \mathbf{RNA}(X, Y, \lambda, \eta), \quad (48)$$

where $\mathbf{RNA}(X, Y, \lambda, \eta) = y^{\text{extr}}$ with y^{extr} the output of Algorithm 1. By construction, y^{extr} is written

$$y^{\text{extr}} = \sum_{i=1}^N c_i^\lambda (y_{i-1} - \eta(x_i - y_{i-1})).$$

If $c_N^\lambda \neq 0$ then y^{extr} matches (2), thus online acceleration iterates in (48) belong to the class of algorithms in (2). If we can ensure $c_N^\lambda \neq 0$, applying Theorem 2.2 recursively will then show an optimal rate of convergence for online acceleration iterations in (48). We do this for linear iterations in what follows.

7.1. Linear Iterations. The next proposition shows that either $c_N^\lambda \neq 0$ holds, or otherwise $y^{\text{extr}} = x^*$ in the linear case.

Proposition 7.1. *Let X, Y (3) be built using iterates from (2). Let g be defined in (7), where the eigenvalues of G are different from one. Consider y^{extr} the output of Algorithm 1 with $\lambda = 0$ and $\eta \neq 0$. If $R = X - Y$ is full column rank, then $c_N^\lambda \neq 0$. Otherwise, $y^{\text{extr}} = x^*$.*

Proof. Since, by definition, $\mathbf{1}^T c^\lambda = 1$, it suffices to prove that the last coefficient $c_N^\lambda \neq 0$. For simplicity, in the scope of this proof we write $c = c^\lambda$. We prove it by contradiction. Let R_- be the matrix R without its last column, and c_- be the coefficients computed by RNA using R_- . Assume $c_N = 0$. In this case,

$$c = [c_-; 0] \quad \text{and} \quad Rc = R_-c_-.$$

This also means that, using the explicit formula for c in (10),

$$\frac{(R^T R)^{-1} \mathbf{1}}{\mathbf{1}(R^T R)^{-1} \mathbf{1}} = \left[\frac{(R_-^T R_-)^{-1} \mathbf{1}}{\mathbf{1}(R_-^T R_-)^{-1} \mathbf{1}}; 0 \right], \quad \Leftrightarrow \quad (R^T R)^{-1} \mathbf{1} = [(R_-^T R_-)^{-1} \mathbf{1}; 0].$$

The equivalence is obtained because

$$\mathbf{1}(R^T R)^{-1} \mathbf{1} = \mathbf{1}^T c = \mathbf{1}^T c_- = \mathbf{1}(R_-^T R_-)^{-1} \mathbf{1}.$$

We can write c and c_- under the form of a linear system,

$$R^T R c = \alpha \mathbf{1}_N, \quad (R_-^T R_-) c_- = \alpha \mathbf{1}_{N-1},$$

where $\alpha = \mathbf{1}(R^T R)^{-1} \mathbf{1} = \mathbf{1}(R_-^T R_-)^{-1} \mathbf{1}$, which is nonzero. We augment the system with c_- by concatenating zeros,

$$R^T R c = \alpha \mathbf{1}_N, \quad \begin{bmatrix} (R_-^T R_-) & 0_{N-1 \times 1} \\ 0_{1 \times N-1} & 0 \end{bmatrix} \begin{bmatrix} c_- \\ 0 \end{bmatrix} = \alpha \begin{bmatrix} \mathbf{1}_{N-1} \\ 0 \end{bmatrix}$$

Let r_+ be the residual at iteration N . This means $R = [R_-, r_+]$. We subtract the two linear systems,

$$\begin{bmatrix} 0 & R^T r_+ \\ r_+^T R & r_+^T r_+ \end{bmatrix} \begin{bmatrix} c_- \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \alpha \neq 0 \end{bmatrix}$$

The $N - 1$ first equations tells us that either $(R^T r_+)_i$ or $c_{-,i}$ are equal to zero. This implies

$$(R^T r_+)^T c = \sum_{i=1}^{N-1} (R^T r_+)_i^T c_i = 0.$$

However, the last equation reads

$$(R^T r_+)^T c + 0 \cdot r_+^T r_+ \neq 0.$$

This is a contradiction, since

$$(R^T r_+)^T c + 0 \cdot r_+^T r_+ = 0.$$

Now, assume R is not full rank. This means there exist a non-zero linear combination such that

$$Rc = 0.$$

However, due to its structure R is a Krylov basis of the Krylov subspace

$$\mathcal{K}_N = \text{span}[r_0, Gr_0, \dots, G^N]$$

If the rank of R is strictly less N (says $N - 1$), this means

$$\mathcal{K}_N = \mathcal{K}_{N-1}.$$

Due to properties of the Krylov subspace, this means that

$$r_0 = \sum_{i=1}^{N-1} \alpha_i \lambda_i v_i$$

where λ_i are distinct eigenvalues of G , and v_i the associated eigenvector. Thus, it suffices to take the polynomial p^* that interpolates the $N - 1$ distinct λ_i . In this case,

$$p^*(G)r_0 = 0.$$

Since $p(1) \neq 0$ because $\lambda_i \leq 1 - \kappa < 1$, we have

$$\min \|Rc\| = \min_{p \in \mathcal{P}_{[N-1]}^{(1)}} \|p(G)r_0\| = \frac{p^*(G)}{p(1)} r_0 = 0.$$

which is the desired result. ■

This shows that we can use *RNA to accelerate iterates coming from RNA*. In numerical experiments, we will see that this new approach significantly improves empirical performance.

7.2. RNA & Nesterov's Method. We now briefly discuss a strategy that combines Nesterov's acceleration with RNA. This means using RNA instead of the classical momentum term in Nesterov's original algorithm. Using RNA, we can produce iterates that are asymptotically adaptive to the problem constants, while ensuring an optimal upper bound if one provides constants L and μ . We show below how to design a condition that decides after each gradient steps if we should combine previous iterates using RNA or Nesterov coefficients.

Nesterov's algorithm first performs a gradient step, then combines the two previous iterates. A more generic version with a basic line search reads

$$\begin{cases} \text{Find } x_{i+1} : f(x_{i+1}) \leq f(y_i) - \frac{1}{2L} \|f(y_i)\|_2^2 \\ y_{i+1} = (1 + \beta)x_{i+1} - \beta x_i, \quad \beta = \frac{1 - \sqrt{\kappa}}{1 + \sqrt{\kappa}}. \end{cases} \quad (49)$$

The first condition is automatically met when we perform the gradient step $x_{i+1} = x_i - \nabla f(x_i)/L$. Based on this, we propose the following algorithm.

Algorithm 3 Optimal Adaptive Algorithm

Compute gradient step $x_{i+1} = y_i - \frac{1}{L} \nabla f(y_i)$.

Compute $y^{\text{extr}} = \text{RNA}(X, Y, \lambda, \eta)$.

Let

$$z = \frac{y^{\text{extr}} + \beta x_i}{1 + \beta}$$

Choose the next iterate, such that

$$y_{i+1} = \begin{cases} y^{\text{extr}} & \text{If } f(z) \leq f(x_i) - \frac{1}{2L} \|f(x_i)\|_2^2, \\ (1 + \eta)x_i - \eta x_{i-1} & \text{Otherwise.} \end{cases}$$

Algorithm 3 has an optimal rate of convergence, i.e., it preserves the worst case rate of the original Nesterov algorithm. The proof is straightforward: if we do not satisfy the condition, then we perform a standard Nesterov step ; otherwise, we pick z instead of the gradient step, and we combine

$$y_{i+1} = (1 + \eta)z - \eta x_{i-1} = y^{\text{extr}}.$$

By construction this satisfies (49), and inherits its properties, like an optimal rate of convergence.

8. NUMERICAL RESULTS

We now study the performance of our techniques on ℓ_2 -regularized logistic regression using acceleration on Nesterov’s accelerated method¹.

We solve a classical regression problem on the Madelon-UCI dataset [Guyon, 2003] using the logistic loss with ℓ_2 regularization. The regularization has been set such that the condition number of the function is equal to 10^6 . We compare to standard algorithms such as the simple gradient scheme, Nesterov’s method for smooth and strongly convex objectives [Nesterov, 2013] and L-BFGS. For the step length parameter, we used a backtracking line-search strategy. We compare these methods with their offline RNA accelerated counterparts, as well as with the online version of RNA described in (48).

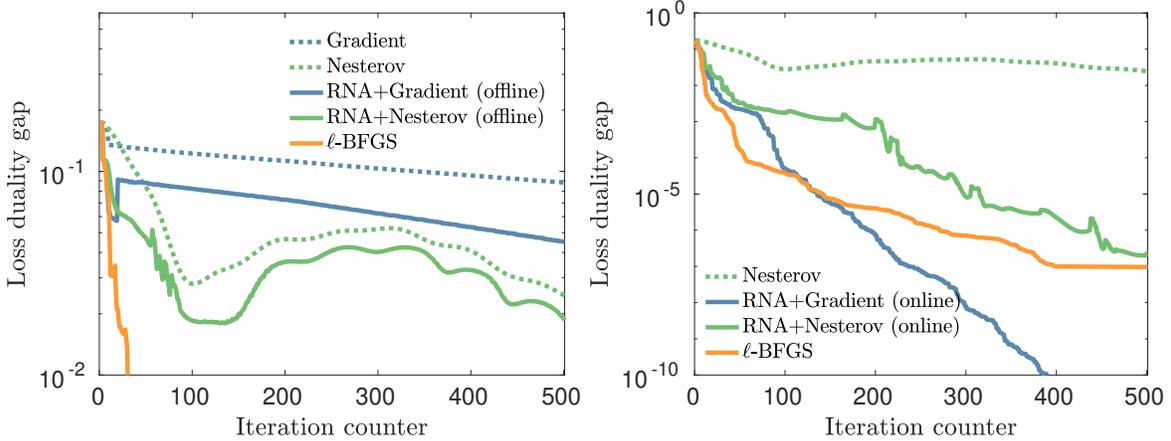


FIGURE 6. Logistic loss on the Madelon [Guyon, 2003]. Comparison between offline (*left*) and online (*right*) strategies for RNA on gradient and Nesterov’s method. We use ℓ -BFGS (with $\ell = 100$ gradients stored in memory) as baseline. Clearly, one step of acceleration improves the accuracy. The performance of online RNA, which applies the extrapolation at *each* step, is similar to that of L-BFGS methods, though RNA does not use line-search and requires 10 times less memory.

We observe in Figure 6 that offline RNA improves the convergence speed of gradient descent and Nesterov’s method. However, the improvement is only a constant factor: the curves are shifted but have the same slope. Meanwhile, the online version greatly improves the rate of convergence, transforming the basic gradient method into an optimal algorithm competitive with line-search L-BFGS.

In contrast to most quasi-Newton methods (such as L-BFGS), RNA does *not* require a Wolfe line-search to be convergent. This is because the algorithm is stabilized with a Tikhonov regularization. In addition, the regularization in a way controls the impact of the noise in the iterates, making the RNA algorithm suitable for stochastic iterations [Scieur et al., 2017a].

We also tested the performance of online RNA on general non-symmetric algorithm, Primal-Dual Gradient Method (PDGM) [Chambolle and Pock, 2011] defined in (25) with $\theta = 0$. We observe in Figure 7 that RNA has substantially improved the performance of the base algorithm.

¹The source code for the numerical experiments can be found on GitHub at <https://github.com/windows7lover/RegularizedNonlinearAcceleration>

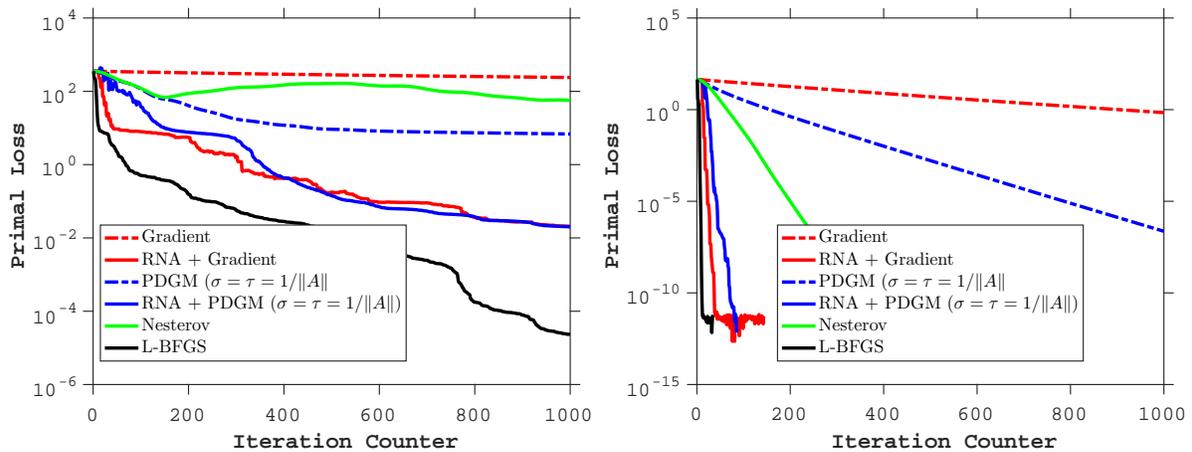


FIGURE 7. Logistic loss on the Madelon [Guyon, 2003]. Left : ℓ_2 regularization parameter $\mu = 10^{-2}$. Right : $\mu = 10^2$. Comparison of online RNA on primal-dual gradient methods with other first-order algorithms.

ACKNOWLEDGEMENTS

The authors are very grateful to Lorenzo Stella for fruitful discussions on acceleration and the Chambolle-Pock method. AA is at CNRS & département d’informatique, École normale supérieure, UMR CNRS 8548, 45 rue d’Ulm 75005 Paris, France, INRIA and PSL Research University. The authors would like to acknowledge support from the *ML & Optimisation* joint research initiative with the *fonds AXA pour la recherche* and Kamet Ventures, as well as a Google focused award. DS was supported by a European Union Seventh Framework Programme (FP7- PEOPLE-2013-ITN) under grant agreement n.607290 SpaRTaN. RB was a PhD student at Northwestern University at the time this work was completed and was supported by Department of Energy grant DE-FG02-87ER25047 and DARPA grant 650-4736000-60049398.

REFERENCES

- Donald G. Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4): 547–560, 1965.
- Claude Brezinski and M Redivo Zaglia. *Extrapolation methods: theory and practice*, volume 2. Elsevier, 2013.
- Stan Cabay and LW Jackson. A polynomial extrapolation method for finding limits and antilimits of vector sequences. *SIAM Journal on Numerical Analysis*, 13(5):734–752, 1976.
- Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.
- Antonin Chambolle and Thomas Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25: 161–319, 2016.
- Daeshik Choi and Anne Greenbaum. Roots of matrices in the study of gmres convergence and crouzeix’s conjecture. *SIAM Journal on Matrix Analysis and Applications*, 36(1):289–301, 2015.
- Michel Crouzeix. Bounds for analytical functions of matrices. *Integral Equations and Operator Theory*, 48(4):461–477, 2004.
- Michel Crouzeix. Numerical range and functional calculus in hilbert space. *Journal of Functional Analysis*, 244(2): 668–690, 2007.
- Michel Crouzeix and César Palencia. The numerical range as a spectral set. *arXiv preprint arXiv:1702.00668*, 2017.
- William F. Donoghue. On the numerical range of a bounded operator. *Michigan Math. J.*, 4(3):261–263, 1957. doi: 10.1307/mmj/1028997958. URL <https://doi.org/10.1307/mmj/1028997958>.

- RP Eddy. Extrapolating to the limit of a vector sequence. In *Information linkage between applied mathematics and industry*, pages 387–396. Elsevier, 1979.
- Bernd Fischer and Roland Freund. Chebyshev polynomials are not always optimal. *Journal of Approximation Theory*, 65(3):261–272, 1991.
- Gene H. Golub and Charles F. Van Loan. *Matrix Computations*, volume 3. JHU Press, 2012.
- Gene H Golub and Richard S Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order richardson iterative methods. *Numerische Mathematik*, 3(1):157–168, 1961.
- R Paul Gorman and Terrence J Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75, 1988.
- Anne Greenbaum, Adrian S Lewis, and Michael L Overton. Variational analysis of the crouzeix ratio. *Mathematical Programming*, 164(1-2):229–243, 2017.
- Isabelle Guyon. Design of experiments of the nips 2003 variable selection benchmark, 2003.
- Felix Hausdorff. Der wertvorrat einer bilinearform. *Mathematische Zeitschrift*, 3(1):314–316, 1919.
- Charles R Johnson. Computation of the field of values of a 2×2 matrix. *J. Res. Nat. Bur. Standards Sect. B*, 78:105, 1974.
- Charles R Johnson. Numerical determination of the field of values of a general complex matrix. *SIAM Journal on Numerical Analysis*, 15(3):595–602, 1978.
- A. Lewis and M. Overton. Partial smoothness of the numerical radius at matrices whose fields of values are disks. *Working paper (mimeo)*, 2018.
- Toshiyuki Mizoguchi. K.j. arrow, l. hurwicz and h. uzawa, studies in linear and non-linear programming. *Economic Review*, 11(3):349–351, 1960. URL <https://EconPapers.repec.org/RePEc:hit:ecorev:v:11:y:1960:i:3:p:349-351>.
- Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.
- Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.
- Yurii Nesterov and Boris T Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.
- Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.
- Yousef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.
- Damien Scieur, Alexandre d’Aspremont, and Francis Bach. Regularized nonlinear acceleration. In *Advances In Neural Information Processing Systems*, pages 712–720, 2016.
- Damien Scieur, Francis Bach, and Alexandre d’Aspremont. Nonlinear acceleration of stochastic algorithms. In *Advances in Neural Information Processing Systems*, pages 3985–3994, 2017a.
- Damien Scieur, Vincent Roulet, Francis Bach, and Alexandre d’Aspremont. Integration methods and optimization algorithms. In *Advances in Neural Information Processing Systems*, pages 1109–1118, 2017b.
- Otto Toeplitz. Das algebraische analogon zu einem satze von fejér. *Mathematische Zeitschrift*, 2(1-2):187–197, 1918.
- Homer F Walker and Peng Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, 2011.