

June 25, 2023

REGULARIZED METHODS VIA CUBIC SUBSPACE MINIMIZATION FOR NONCONVEX OPTIMIZATION

STEFANIA BELLAVIA*, DAVIDE PALITTA†, MARGHERITA PORCELLI†,‡, AND
VALERIA SIMONCINI†,§

Abstract. The main computational cost per iteration of adaptive cubic regularization methods for solving large-scale nonconvex problems is the computation of the step s_k , which requires an approximate minimizer of the cubic model. We propose a new approach in which this minimizer is sought in a low dimensional subspace that, in contrast to classical approaches, is reused for a number of iterations. A regularized Newton step to correct s_k is also incorporated whenever needed. We show that our method increases efficiency while preserving the worst-case complexity of classical cubic regularized methods. We also explore the use of rational Krylov subspaces for the subspace minimization, to overcome some of the issues encountered when using polynomial Krylov subspaces. We provide several experimental results illustrating the gains of the new approach when compared to classic implementations.

Key words. Adaptive regularization methods, nonconvex optimization, Krylov subspaces, worst-case iteration complexity

AMS subject classifications. 49M37, 65K05, 68W40

1. Introduction. We address the numerical solution of possibly nonconvex, unconstrained optimization problems of the form

$$(1.1) \quad \min_{x \in \mathbb{R}^n} f(x),$$

where the objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is supposed to be twice-continuously differentiable and bounded from below. To attack (1.1) we consider second order adaptive regularization methods (AR2). These are well established, globally convergent variants of the Newton method for (1.1), where the step length is implicitly controlled. This feature is achieved by adding a cubic term in the classic quadratic Taylor model to penalize long steps [9, Section 3.3]. More precisely, at a generic AR2 iteration k , the trial step is computed by (approximately) solving the subproblem

$$(1.2) \quad \min_{s \in \mathbb{R}^n} m_k(s),$$

where $m_k(s)$ is the following *cubic regularized model* of the objective function f around the current point x_k

$$(1.3) \quad m_k(s) \stackrel{\text{def}}{=} f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T \nabla^2 f(x_k) s + \frac{1}{3} \sigma_k \|s\|_2^3 = T_k(s) + \frac{1}{3} \sigma_k \|s\|_2^3.$$

The scalar $\sigma_k > 0$ is the regularization parameter and $T_k(s)$ is the second-order Taylor-series expansion of f at x_k .

For nonconvex problems, AR2 exhibits a worst-case $\mathcal{O}(\epsilon^{-3/2})$ iteration-complexity to find an ϵ -approximate first-order minimizer, i.e., a point x_ϵ such that

$$(1.4) \quad \|\nabla f(x_\epsilon)\| \leq \epsilon,$$

*Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Viale Morgagni, 40/44, 50134 Florence, Italy. Member of the INdAM Research Group GNCS. Email: stefania.bellavia@unifi.it

†Dipartimento di Matematica, (AM)², Alma Mater Studiorum - Università di Bologna, Piazza di Porta San Donato 5, 40126 Bologna, Italia. Members of the INdAM Research Group GNCS. Emails: {davide.palitta,margherita.porcelli,valeria.simoncini}@unibo.it

‡ISTI-CNR, Via Moruzzi 1, Pisa, Italia

§IMATI-CNR, Via Ferrata 5/A, Pavia, Italia

and a worst-case $\mathcal{O}(\max(\epsilon^{-3/2}, \epsilon_H^{-2}))$ iteration-complexity to find an (ϵ, ϵ_H) -approximate second-order minimizer, i.e., a point x_{ϵ, ϵ_H} such that

$$\|\nabla f(x_{\epsilon, \epsilon_H})\| \leq \epsilon \quad \text{and} \quad \lambda_{\min}(\nabla^2 f(x_{\epsilon, \epsilon_H})) \geq -\epsilon_H.$$

AR2 is optimal in terms of complexity when compared to standard second-order methods, such as trust-region and Newton methods, whose iteration-complexity to compute an ϵ -approximate first-order minimizer amounts to $\mathcal{O}(\epsilon^{-2})$ [9, Chapter 3].

The computational bottleneck of AR2 procedures is the construction of an approximate minimizer to the cubic model $m_k(s)$. In the current literature this task is performed either through the so-called “secular” equation, solved by root-finding methods [9, Chapter 9], or by approximately minimizing the cubic model over a sequence of nested Krylov subspaces generated at each nonlinear iteration, onto which the secular equation is projected and solved [9, Chapter 10].

In this latter context, we propose a new algorithm that minimizes the cubic model m_k over a low-dimensional subspace that, in contrast to classic approaches, is not recomputed at each nonlinear iteration, but it is rather *frozen* along the iterations, namely is kept as long as it provides sufficiently good outcomes. In particular, the minimization of m_k over the frozen subspace is not primarily aimed at computing an approximate minimizer to the cubic model but rather at providing an adaptive regularization parameter for the Newton step in the full space; see section 3. The effectiveness of the freezing technique relies on the quality of the approximation space, in terms of spectral information.

Our approach can be interpreted as a predictor-corrector scheme, where the prediction step is given by the minimization of the cubic model in a (low) dimensional subspace and the corrector step is a regularized Newton step where, again, the regularization parameter is provided by the subspace minimization.

We prove that our novel algorithm maintains the complexity of the original AR2, regardless of the selected subspace. To keep memory requirements under control, we suggest the use of rational Krylov subspaces; see, e.g., [33]. In addition to featuring significantly smaller dimensions than polynomial Krylov subspaces to achieve a prescribed level of accuracy, our computational experiments suggest that rational spaces are particularly well suited for sequences of slowly varying subproblems such as those arising in optimization methods.

The main practical advantage of our new scheme when using rational Krylov subspaces is the remarkable reduction in the overall number of factorizations of the Hessian matrix required by the computation of the step when compared to standard adaptive-regularization algorithms employing the secular approach such as those in [21]. In particular, for many problems, a sizable number of nonlinear iterations sees the computation of a single factorization if not even none; see section 6. Furthermore, when looking for first-order optimality points, the computation of the smallest eigenvalue of the Hessian matrix is never required in our scheme in contrast to what is asked in different routines recently proposed in the literature; see, e.g., [25]. Our experimental results also illustrate that the use of rational Krylov methods is a key ingredient for the success of the overall method, as polynomial Krylov spaces may require significantly larger dimensions, especially for ill-conditioned problems; see Remark 5.1 and section 6.

Related works. The efficient solution of the subproblem (1.2) in AR2-like methods have been the subject of intense research in the last 20 years; see [9, Chapters 9-10] and references therein. State-of-the-art methods can be mainly divided into two classes: Approaches based on the approximate minimization of the regularized model using a Lanczos process [7, 23, 24, 29, 31] and schemes based on the secular formulation exploiting matrix factorizations [21]. In [7, 23, 24] the cubic model is minimized over a sequence of nested Krylov subspaces generated by

the Lanczos procedure. In [29] it is observed that this approach may produce a large dimensional subspace when applied to ill-conditioned problems and a nested restarting version for the large scale and ill-conditioned case is proposed. Both [29] and [31] reduce the minimization subproblem to a suitable generalized eigenvalue problem.

The algorithm proposed in [20] concurrently solves a set of systems with the shifted Hessian matrix for a predefined large number of shifts by means of a hybrid procedure employing the secular equation combined with the Lanczos method for indefinite linear systems.

A different approach is taken in [6], where the gradient descent is used to approximate the global minimizer of the cubic model; the worst-case iteration complexity is studied.

More recently, several works have focused on the development of modified Newton methods which exhibit the AR2-like worst-case complexity (or nearly so); see e.g. [4, 14, 25]. In [14] a general framework covering several algorithms, like AR2 and the modified trust-region algorithm named TRACE [13], is given. At each iteration the trial step and the multiplier estimate are calculated as an approximate solution of a constrained optimization subproblem whose objective function is a regularized Newton model. A second-order algorithm alternating between a regularized Newton and negative curvature steps is given in [25], while [4] incorporates cubic descent into a quadratic regularization framework.

Similarly to the methods in [4, 14, 25], also our novel scheme employs quadratic-regularization variants preserving the complexity results of the cubic regularization. On the other hand, in contrast to what is done in, e.g., [4, 14], we employ the regularized Newton step only when the step provided by the subspace minimization is not satisfactory. Moreover, the multiplier estimate is a byproduct of the subspace minimization process. We also aim at reducing the computational cost associated to the step computation. To this end, we make use of rational Krylov subspaces whose expensive basis construction is carried out only a handful of times throughout all the nonlinear iterations. Indeed, once a rational Krylov subspace is constructed at a certain nonlinear iteration k , we reused it in the subspace minimization as long as possible thus fully capitalizing on the computational efforts performed for its construction. Such a strategy significantly differs from the ones proposed in, e.g., [20, 29, 31].

Outline. Here is a synopsis of the paper. In section 2 we present some preliminary background about the secular formulation of (1.2) (section 2.1) and the AR2 method (section 2.2). Section 3 is devoted to present the main contribution of this paper, namely the novel solution process for (1.2). In section 4 we study the complexity of our algorithm, and show that the optimal worst-case iteration complexity of AR2 is maintained. In section 5 we describe the rational Krylov subspace and its use within our solution scheme. A panel of diverse numerical results displaying the potential of our novel approach is presented in section 6 whereas in section 7 we draw some conclusions.

Notation. Throughout the paper we adopt the following notation. At the k -th nonlinear iteration, the gradient and the Hessian of f evaluated at the current iterate x_k are denoted as $g_k = \nabla f(x_k)$ and $H_k = \nabla^2 f(x_k)$, respectively. The symbol I is used for the $n \times n$ identity matrix, and $\|\cdot\|$ denotes the Euclidean norm of a vector.

2. Preliminaries. This section provides a brief description of the features of the cubic regularized minimization problem (1.2) in terms of its secular formulation and a quick review of the main steps of the AR2 method.

2.1. The “secular” formulation. The secular formulation of the subproblem (1.2) is defined by exploiting the characterization of the global minimizer of $m_k(s)$ reported below. For the sake of simplicity, in this section we omit the subscript k .

THEOREM 2.1. [9, Theorem 8.2.8] *Any global minimizer s^* of (1.2) satisfies*

$$(2.1) \quad (H + \lambda^* I)s^* = -g,$$

where $H + \lambda^* I$ is positive semidefinite and

$$\lambda^* = \sigma \|s^*\|.$$

If $H + \lambda^* I$ is positive definite, then s^* is unique.

Let λ_1 be the leftmost eigenvalue of H . In the so-called “easy case”, $\lambda^* > \lambda_S \stackrel{\text{def}}{=} \max(0, -\lambda_1)$ and a solution to (1.2) can be computed by solving the scalar *secular equation*

$$(2.2) \quad \phi_R(\lambda; g, H, \sigma) \stackrel{\text{def}}{=} \|(H + \lambda I)^{-1}g\| - \frac{\lambda}{\sigma} = 0.$$

It can be proved that the Newton and the secant methods applied to (2.2) rapidly converge to λ^* starting from any approximation in $(\lambda_S, +\infty)$ [7, 9, 21]. Whenever the computation of λ_1 is prohibitive and the computation of appropriate starting guesses is thus not guaranteed, suitable estimations to λ_1 can be derived as done in [21].

As opposite to the “easy case”, the “hard case” takes place when $\lambda_1 < 0$ and g is orthogonal to the λ_1 -eigenspace of H , namely $v^T g = 0$ for all v in $\{v : (H - \lambda_1 I)v = 0\}$. In this case, $\lambda^* = -\lambda_1$ and there is no solution to the secular equation in $(\lambda_S, +\infty)$. Sophisticated numerical strategies have to be employed to compute the global minimizer s^* in this case; see, e.g., [9, Section 9.3.4].

The theorem below summarizes the form of the global minimizer of (1.2) in both the easy and the hard cases.

THEOREM 2.2. [9, Corollary 8.3.1] *Any global minimizer of (1.2) can be expressed as*

$$(2.3) \quad s^* = \begin{cases} -(H + \lambda^* I)^{-1}g \text{ (uniquely)} & \text{if } \lambda^* > -\lambda_1, \\ -(H + \lambda^* I)^\dagger g + \alpha v_1 & \text{if } \lambda^* = -\lambda_1, \end{cases}$$

where $(H + \lambda^* I)^\dagger$ denotes the pseudoinverse of $H + \lambda^* I$, $\lambda^* = \sigma \|s^*\|$, λ_1 is the leftmost eigenvalue of H , v_1 is any corresponding eigenvector, and α is one of the two roots of $\|-(H + \lambda^* I)^\dagger g + \alpha v_1\| = \lambda^* / \sigma$.

2.2. The AR2 method. The AR2 method is an iterative procedure where, at iteration k , a trial step s_k is computed by approximately minimizing the cubic model (1.3). Theorem 2.2 shows the exact minimizer of $m_k(s)$ but, in practice, only an approximate minimizer s_k satisfying conditions (2.5) and (2.6) is indeed necessary; see, e.g., [9, p.65].

Given the trial step s_k , the trial point $x_k + s_k$ is then used to compute the ratio

$$(2.4) \quad \rho_k = \frac{f(x_k) - f(x_k + s_k)}{T_k(0) - T_k(s_k)},$$

with T_k as in (1.3). If $\rho_k \geq \eta_1$, with $\eta_1 \in (0, 1)$, then the trial point is accepted, the iteration is declared successful and the regularization parameter σ_k is possibly decreased. Otherwise, an unsuccessful iteration occurs: The point $x_k + s_k$ is rejected and the regularisation parameter is increased. Algorithm 2.1 summarizes a possible implementation of the AR2 algorithm; see also [9, Algorithm 3.3.1].

Algorithm 2.1 The Adaptive-regularization algorithm with a second order model (AR2) algorithm [9]

Require: An initial point x_0 ; accuracy threshold $\epsilon \in (0, 1)$; an initial regularization parameter $\sigma > 0$ are given as well as constants $\eta_1, \eta_2, \gamma_1, \gamma_2, \theta_1, \sigma_{\min}$ that satisfy

$$\sigma_{\min} \in (0, \sigma_0], \quad \theta_1 > 0, \quad 0 < \eta_1 \leq \eta_2 < 1, \quad 0 < \gamma_1 < 1 < \gamma_2.$$

- 1: Compute $f(x_0), g_0 = \nabla f(x_0), H_0 = \nabla^2 f(x_0)$, and set $k = 0$.
- 2: **Step 1: Test for termination.** If $\|g_k\| \leq \epsilon$, terminate.
- 3: **Step 2: Step computation.** Compute a step s_k such that

$$(2.5) \quad m_k(s_k) < m_k(0)$$

and

$$(2.6) \quad \|\nabla m_k(s_k)\| \leq \frac{1}{2}\theta_1\|s_k\|^2$$

- 4: **Step 3: Acceptance of the trial point.** Compute $f(x_k + s_k)$ and the ratio ρ_k given in (2.4).
- 5: If $\rho_k \geq \eta_1$, then define $x_{k+1} = x_k + s_k$ and compute $g_{k+1} = \nabla f(x_{k+1})$ and $H_{k+1} = \nabla^2 f(x_{k+1})$. Otherwise define $x_{k+1} = x_k$.
- 6: **Step 4: Regularization parameter update.** Compute

$$(2.7) \quad \sigma_{k+1} \in \begin{cases} [\max(\sigma_{\min}, \gamma_1\sigma_k), \sigma_k] & \text{if } \rho_k \geq \eta_2 \\ \sigma_k & \text{if } \rho_k \in [\eta_1, \eta_2) \\ \gamma_2\sigma_k & \text{otherwise} \end{cases}$$

- 7: Increment k by one and **goto** Step 1.
-

3. The new FAR2 algorithm. In this section we introduce our subspace cubic approach that we name the Frozen AR2 (FAR2) algorithm. The main idea behind FAR2 is to construct a “good” subspace \mathcal{K} at some nonlinear iteration, say at the initial iteration, compute a minimizer to the cubic model projected onto this subspace satisfying (2.6)¹ and then keep using such subspace also in the subsequent nonlinear iterations. If (2.6) is not satisfied by minimizing (1.3) over the current \mathcal{K} , a further regularized Newton-like step is performed. The employed regularization parameter $\widehat{\lambda}_k$ is a by-product of the minimization process over the current subspace and it is cheaply computed by solving the *projected* secular equation, that is equation (2.2) where the quantities involved H and g consist of the projection of H_k and g_k onto \mathcal{K} , respectively. The current subspace \mathcal{K} is discarded and a new, more informative approximation space is generated from scratch whenever either the computed $\widehat{\lambda}_k$ does not provide a positive definite matrix $H_k + \widehat{\lambda}_k I$, or the regularized Newton step s_k fails to satisfy the following condition:

$$(3.1) \quad C_{\text{low}}\|\widehat{s}_k\| \leq \|s_k\| \leq C_{\text{up}}\|\widehat{s}_k\|,$$

where \widehat{s}_k is the minimizer of the cubic model onto the current subspace and C_{low} and C_{up} are given positive constants. Condition (3.1) is crucial to analyze the complexity of FAR2 in section 4 and is similar to the one adopted in [14, Eq. (2.1)]. Note that condition (3.1) is not restrictive as it only requires that the ratio $\|s_k\|/\|\widehat{s}_k\|$ is lower bounded away from zero and upper bounded.

The overall FAR2 procedure is described in Algorithm 3.1 and Algorithm 3.2. The outcomes of Algorithm 3.2 are the regularization parameter $\widehat{\lambda}_k$, the step \widehat{s}_k

¹Condition (2.5) is naturally satisfied as the null vector lies in such subspace.

and the matrix W_k storing the basis of the subspace used to carry out the minimization process needed to generate $\widehat{\lambda}_k$ and \widehat{s}_k . The matrix W_k is such that $\text{Range}(W_k) = \text{Range}([g_k, V_{k+1}])$. The columns of the matrix V_{k+1} represent an orthonormal basis of the current subspace \mathcal{K} and they are computed from scratch whenever the input parameter **refresh** is equal to one. The minimization is carried out over the augmented subspace spanned by W_k to exactly project the current gradient g_k .

The reduced step \widehat{s}_k is the exact global minimizer of $m_k(s)$ onto the subspace generated by W_k . Moreover, the regularization parameter has the following form:

$$\widehat{\lambda}_k = \sigma_k \|\widehat{s}_k\|.$$

In Line 4 of Algorithm 3.1 the minimizer of the cubic model onto the current subspace \widehat{s}_k and the regularization parameter $\widehat{\lambda}_k$ are computed by invoking Algorithm 3.2. In Line 9, if the matrix $H_k + \widehat{\lambda}_k I$ is positive definite and the approximation vector $W_k \widehat{s}_k$ does not satisfy the condition (2.6), the regularized Newton step $s_k = -(H_k + \widehat{\lambda}_k I)^{-1} g_k$ is computed.

The flag **refresh** rules the computation of a new subspace. A new subspace is computed whenever **refresh** is 1. **refresh** is set to 1 at the first iteration and whenever one of the following two situations occurs:

- i) The minimization of the model in the current subspace does not satisfy (2.6) and, in addition, the computed shift $\widehat{\lambda}_k$ does not provide a positive definite shifted matrix $H_k + \widehat{\lambda}_k I$ (Line 11 of Algorithm 3.1).
- ii) $H_k + \widehat{\lambda}_k I$ is positive definite but condition (3.1) is not met (Line 14 of Algorithm 3.1).

One iteration is declared unsuccessful when **i**) or **ii**) occurs or, as in classic adaptive methods, when the ratio ρ_k in (2.4) is smaller than the input parameter $\eta_1 \in (0, 1)$. Note that in case **i**) or **ii**) the parameter σ_k is left unchanged and a new subspace is computed at the subsequent iteration. In case the unsuccessful iteration is due to $\rho_k < \eta_1$ the parameter σ_k is increased. We refrain from increasing σ_k in cases **i**) and **ii**) as the failure is not ascribed to an unsatisfactory model.

We observe that when the condition in Line 5 of Algorithm 3.2 is met, that is when the subspace minimization provides a reduced step \widehat{s}_k such that the pullback step $W_k \widehat{s}_k$ satisfies (2.6), then the computation of the regularized Newton step and the consequent factorization of the shifted Hessian matrix in the full space is not needed.

We would like to point out that the nature of the step s_k in the successful iterations is strictly related to our refresh strategy. In particular, whenever the refresh takes place, the step s_k computed in Line 6 of Algorithm 3.1 is such that (2.6) holds and $\|s_k\| = \|\widehat{s}_k\|$ since W_k has orthonormal columns. On the other hand, the step s_k computed in Line 9 of Algorithm 3.1 (**refresh** = 0) is the exact global minimizer of the quadratic regularized model

$$(3.2) \quad m_k^Q(s) \stackrel{\text{def}}{=} T_k(s) + \frac{1}{2} \widehat{\lambda}_k \|s\|^2,$$

with $\widehat{\lambda}_k = \sigma_k \|\widehat{s}_k\|$, since $H_k + \widehat{\lambda}_k I$ is positive definite whenever **refresh** = 0. Then, in this latter case, we have

$$(3.3) \quad m_k^Q(s_k) = T_k(s_k) + \frac{1}{2} \sigma_k \|\widehat{s}_k\| \|s_k\|^2 \leq m_k^Q(0).$$

In our method we use second derivatives. Therefore, given $\epsilon > 0, \epsilon_H > 0$, it is possible to seek an (ϵ, ϵ_H) -approximately second-order minimizer. To this end, Algorithm 3.1 needs to be equipped with a further stopping criterion, and an

Algorithm 3.1 The Frozen AR2 (FAR2) algorithm

Require: An initial point x_0 ; accuracy threshold $\epsilon \in (0, 1)$; an initial regularization parameter $\sigma_0 > 0$ are given, and constants $\eta_1, \eta_2, \gamma_1, \gamma_2, \theta_1, \sigma_{\min}$ s.t.

$$\sigma_{\min} \in (0, \sigma_0], \theta_1 > 0, 0 < \eta_1 \leq \eta_2 < 1, 0 < \gamma_1 < 1 < \gamma_2, 0 < C_{\text{low}} < C_{\text{up}}$$

An integer j_{\max} .

- 1: Compute $f(x_0), g_0 = \nabla f(x_0), H_0 = \nabla^2 f(x_0)$, and set $k = 0$, **refresh** = 1 and $V_0 = []$.
- 2: **Step 1: Test for termination.** If $\|g_k\| \leq \epsilon$, terminate.
- 3: **Step 2: Step computation (reduction of secular equation).**
- 4: Compute the basis W_k and V_{k+1} , the scalar $\hat{\lambda}_k$, and the reduced step \hat{s}_k by Algorithm 3.2
- 5: **if refresh or** $W_k \hat{s}_k$ satisfies (2.6)
- 6: set $s_k = W_k \hat{s}_k$, **refresh** = 0, and **goto** Step 3
- 7: **end if**
- 8: **if** $H_k + \hat{\lambda}_k I > 0$
- 9: Compute $s_k = -(H_k + \hat{\lambda}_k I)^{-1} g_k$.
- 10: **else** % unsuccessful iteration, step rejection
- 11: Define $x_{k+1} = x_k$, $\sigma_{k+1} = \sigma_k$, set **refresh** = 1, increment k by one and **goto** Step 1
- 12: **end if**
- 13: **if** $\|s_k\|/\|\hat{s}_k\| < C_{\text{low}}$ or $\|s_k\|/\|\hat{s}_k\| > C_{\text{up}}$ % unsuccessful iteration, step rejection
- 14: Define $x_{k+1} = x_k$, $\sigma_{k+1} = \sigma_k$ set **refresh** = 1, increment k by one and **goto** Step 1
- 15: **Step 3** Compute

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{T_k(0) - T_k(s_k)}.$$

- 16: **Step 4**
- 17: **if** $\rho_k \geq \eta_1$ **then** % Successful iteration
- 18: set $x_{k+1} = x_k + s_k$, compute $g_{k+1} = \nabla f(x_{k+1})$, $H_{k+1} = \nabla^2 f(x_{k+1})$
- 19: **else** % unsuccessful iteration, step rejection
- 20: set $x_{k+1} = x_k$
- 21: **end if**
- 22: **Step 5: Regularization parameter update.** Compute

$$(3.4) \quad \sigma_{k+1} \in \begin{cases} [\max\{\sigma_{\min}, \gamma_1 \sigma_k\}, \sigma_k] & \text{if } \rho_k \geq \eta_2 \\ \sigma_k & \text{if } \rho_k \in [\eta_1, \eta_2) \\ \gamma_2 \sigma_k & \text{otherwise} \end{cases}$$

- 23: Increment k by one and **goto** Step 1.
-

additional condition on the step has to be imposed. More precisely, the method is stopped whenever the following two conditions hold:

$$(3.5) \quad \|g_k\| \leq \epsilon \quad \text{and} \quad \lambda_{\min}(H_k) \geq -\epsilon_H.$$

Secondly, when **refresh** = 1, the trial step s_k computed in Step 2 is required to satisfy the following condition in addition to (2.6):

$$(3.6) \quad \lambda_{\min}(\nabla^2 m_k(s_k)) \geq -\theta_2 \|s_k\|, \quad \theta_2 > 0.$$

We will refer to this variant of FAR2 as FAR2-SO (Frozen AR2-Second Order). The corresponding algorithm is detailed in the Appendix.

Algorithm 3.2 Refresh and solution of the projected secular equation

Require: The matrix H_k ; the vector g_k ; an initial basis V_k ; accuracy threshold $\theta_1 > 0$; the parameter σ_k ; an integer $j_{\max} \ll n$; **refresh**.

```

1:   if refresh                                     % Generate new proj space
2:       Set  $V_{k+1} = []$ 
3:       for  $j = 1, \dots, j_{\max} - 1$ 
4:           Set2  $W^{(j)} = \text{orth}([V_{k+1}, g_k])$ 
5:           Compute projections  $g^{(j)} = (W^{(j)})^T g_k$ ,  $H^{(j)} = (W^{(j)})^T H_k W^{(j)}$ 
6:           Find  $\hat{\lambda}$  s.t.  $\phi_R(\hat{\lambda}; g^{(j)}, H^{(j)}, \sigma_k) = 0$ , i.e.
                                     
$$\hat{\lambda} = \sigma_k \|(H^{(j)} + \hat{\lambda}I)^{-1} g^{(j)}\|$$

7:           Set  $\hat{s} = -(H^{(j)} + \hat{\lambda}I)^{-1} g^{(j)}$ 
8:           if  $\|\nabla m_k(W^{(j)} \hat{s})\| \leq \frac{1}{2} \theta_1 \|\hat{s}\|^2$ 
9:               Set  $\hat{\lambda}_k = \hat{\lambda}$ ,  $\hat{s}_k = \hat{s}$ ,  $W_k = W^{(j)}$  and return
10:          end if
11:          Expand  $V_{k+1}$  with new basis vector
12:       end for
13:   else                                           % Project onto the old space
14:       Set  $W^{(\hat{j})} = \text{orth}([V_k, g_k])$ 
15:       Compute projections  $g^{(\hat{j})} = (W^{(\hat{j})})^T g_k$ ,  $H^{(\hat{j})} = (W^{(\hat{j})})^T H_k W^{(\hat{j})}$ 
16:       Find  $\hat{\lambda}$  s.t.  $\phi_R(\hat{\lambda}; g^{(\hat{j})}, H^{(\hat{j})}, \sigma_k) = 0$ , i.e.
                                     
$$\hat{\lambda} = \sigma_k \|(H^{(\hat{j})} + \hat{\lambda}I)^{-1} g^{(\hat{j})}\|$$

17:       Set  $\hat{s}_k = -(H^{(\hat{j})} + \hat{\lambda}I)^{-1} g^{(\hat{j})}$ ,  $\hat{\lambda}_k = \hat{\lambda}$ 
18:       Set  $V_{k+1} = V_k$  and  $W_k = W^{(\hat{j})}$ 
19:   end if

```

We stress that H_k is expected to be positive definite during the final convergence phase of the FAR2-SO method and this ensures that condition (3.1) holds eventually. In fact, (3.1) is satisfied whenever there exist $0 < \theta_{\min} < \theta_{\max}$ such that $\lambda_i(H_k + \hat{\lambda}_k I)^{-2} \in [\theta_{\min}, \theta_{\max}]$, for $i = 1, \dots, n$. Indeed, we have

$$\|s_k\|^2 = g_k^T (H_k + \hat{\lambda}_k I)^{-2} g_k,$$

so that

$$\theta_{\min} \|g_k\|^2 \leq \|s_k\|^2 \leq \theta_{\max} \|g_k\|^2.$$

Similarly, since g_k is always included in the subspace employed to compute \hat{s}_k , it holds

$$\hat{\theta}_{\min} \|g_k\|^2 \leq \|\hat{s}_k\|^2 \leq \hat{\theta}_{\max} \|g_k\|^2,$$

where $\hat{\theta}_{\min}$ and $\hat{\theta}_{\max}$ denote the smallest and largest eigenvalues of $(\hat{H}_k + \hat{\lambda}_k I)^{-2}$, respectively. Notice that $\hat{\theta}_{\min}$ and $\hat{\theta}_{\max}$ are guaranteed to be strictly positive as $\hat{\lambda}_k$ solves the reduced secular equation so that $\hat{H}_k + \hat{\lambda}_k I$ is certainly positive definite. Therefore,

$$\|g_k\|^2 \leq \frac{\|\hat{s}_k\|^2}{\hat{\theta}_{\min}}, \quad \frac{\|\hat{s}_k\|^2}{\hat{\theta}_{\max}} \leq \|g_k\|^2.$$

By putting everything together, we have

$$\frac{\theta_{\min}}{\hat{\theta}_{\max}} \|\hat{s}_k\|^2 \leq \|s_k\|^2 \leq \frac{\theta_{\max}}{\hat{\theta}_{\min}} \|\hat{s}_k\|^2.$$

4. Complexity analysis of finding first-order optimality points. In this section, given $\epsilon > 0$ we are going to provide an upper bound on the number of iterations needed by FAR2 to compute an ϵ -approximately first-order optimality point (see (1.4)). To this end, we make the following assumptions on the optimization problem (1.1).

AS.1 $f(x)$ is twice continuously differentiable in \mathbb{R}^n , and its gradient $\nabla f(x)$ and its Hessian $\nabla^2 f(x)$ are Lipschitz continuous on \mathbb{R}^n with Lipschitz constants L_1 and L_2 , respectively.

AS.2 f is bounded from below in \mathbb{R}^n , that is there exists a constant f_{low} such that $f(x) \geq f_{\text{low}}$ for all $x \in \mathbb{R}^n$.

We recall that, given $x_k, s_k \in \mathbb{R}^n$ and by letting $T_k(s_k)$ be the second-order Taylor approximation of $f(x_k + s_k)$ around x_k , Assumption **AS.1** implies the following

$$(4.1) \quad |f(x_k + s_k) - T_k(s_k)| \leq \frac{L_2}{6} \|s_k\|^3,$$

see, e.g., [9, Corollary A.8.4].

Let us denote by \mathcal{S}_k the set of indexes of successful iterations up to iteration k generated by the FAR2 method detailed in Algorithms 3.1-3.2. Then, the following result allows us to bound the number of iterations in terms of the number of successful iterations under the assumption that the regularization parameter is bounded from above.

LEMMA 4.1. *Suppose that the sequence $\{x_k\}$ is generated by the FAR2 algorithm. Assume that $\sigma_k \leq \sigma_{\max}$ for some $\sigma_{\max} > 0$. Then,*

$$k \leq |\mathcal{S}_k| \left(1 + \frac{|\log \gamma_1|}{\log \gamma_2} \right) + \frac{1}{\log \gamma_2} \log \left(\frac{\sigma_{\max}}{\sigma_0} \right).$$

Proof. The proof follows proceeding as in [9, Lemma 2.4.1]. □

Let us now denote by \mathcal{I}_{RN} the set of indexes of iterations such that the step s_k used at Step 3 has been computed in Line 9 of Algorithm 3.1, i.e., it is a regularized Newton step. We observe that at any successful iteration $k \notin \mathcal{I}_{RN}$ the step s_k is the step used in the classical adaptive regularization method with second order model. Then, the theoretical results given in [9, Section 3.3] can be used for these iterations. We also observe that condition (3.1) holds at any successful iteration $k \in \mathcal{I}_{RN}$ and that, as already noticed, σ_k is left unchanged at any unsuccessful iteration in \mathcal{I}_{RN} where (3.1) does not hold.

LEMMA 4.2. (Decrease in the Taylor-series model) *At every successful iteration k of the FAR2 algorithm, it holds that*

$$(4.2) \quad T_k(0) - T_k(s_k) > \frac{1}{2} \sigma_k \|\widehat{s}_k\| \|s_k\|^2, \quad \text{for } k \in \mathcal{I}_{RN},$$

and

$$(4.3) \quad T_k(0) - T_k(s_k) > \frac{1}{3} \sigma_k \|s_k\|^3, \quad \text{for } k \notin \mathcal{I}_{RN}.$$

Proof. Let us consider the case $k \in \mathcal{I}_{RN}$. Inequality (3.3) yields

$$T_k(0) - T_k(s_k) = m_k^Q(0) - m_k^Q(s_k) + \frac{1}{2} \widehat{\lambda}_k \|s_k\|^2 \geq \frac{1}{2} \widehat{\lambda}_k \|\widehat{s}_k\|^2 = \frac{1}{2} \sigma_k \|\widehat{s}_k\| \|s_k\|^2.$$

In case $k \notin \mathcal{I}_{RN}$, the result follows from [9, Lemma 3.3.1]. □

²Here and in the following, orth defines a procedure that updates a matrix having orthonormal columns by adding a new column and orthogonalizing it with respect to the previous ones.

LEMMA 4.3. (**Upper bound on the regularization parameter**) *Suppose that Assumption **AS.1** holds. At each iteration of the FAR2 algorithm it holds*

$$(4.4) \quad \sigma_k \leq \sigma_{\max} \stackrel{\text{def}}{=} \max \left\{ \sigma_0, \gamma^2 \frac{L_2 C_{\text{up}}}{3(1-\eta_1)}, \gamma^2 \frac{L_2}{2(1-\eta_1)} \right\},$$

where γ_2 and η_1 are given in the algorithm.

Proof. For any $k \notin \mathcal{I}_{RN}$, exploiting (4.3) and proceeding as in [9, Lemma 3.3.2] we can show that the iteration is successful whenever

$$\sigma_k \geq \frac{L_2}{2(1-\eta_1)}.$$

In case $k \in \mathcal{I}_{RN}$ the following two situations can occur:

- a) The matrix $H_k + \hat{\lambda}_k I$ is positive definite and condition (3.1) holds.
- b) The iteration is declared unsuccessful either in Line 11 or in Line 14 of Algorithm 3.1. In both cases **refresh** is set to 1, σ_k is left unchanged, and $k+1 \notin \mathcal{I}_{RN}$.

Let us consider case a). The ratio ρ_k is evaluated in Step 5 and from its definition we have

$$\begin{aligned} |\rho_k - 1| &= \frac{|f(x_k) - f(x_k + s_k) - T_k(0) + T_k(s_k)|}{T_k(0) - T_k(s_k)} \\ &= \frac{|f(x_k + s_k) - T_k(s_k)|}{T_k(0) - T_k(s_k)} \\ &\leq \frac{L_2 \|s_k\|^3}{3\sigma_k \|\hat{s}_k\| \|s_k\|^2} \\ &= \frac{L_2 \|s_k\|}{3\sigma_k \|\hat{s}_k\|}, \end{aligned}$$

where we used the inequalities (4.1) and (4.2). Therefore, by using (3.1), we can conclude that if $k \in \mathcal{I}_{RN}$ and case **a)** occurs, then

$$\sigma_k \geq \frac{L_2 C_{\text{up}}}{3(1-\eta_1)},$$

implies $\rho_k > \eta_1$, i.e. the iteration is successful. Therefore, thanks to the mechanism of the algorithm we have

$$\sigma_k \leq \gamma^2 \frac{L_2 C_{\text{up}}}{3(1-\eta_1)},$$

for any $k \in \mathcal{I}_{RN}$ and case **a)** holds.

Finally, for unsuccessful iterations $k \in \mathcal{I}_{RN}$, case **b)** does not yield an increase in σ_k . Therefore, σ_k cannot be larger than σ_{\max} in (4.4) and the thesis follows. \square

We are now going to show two crucial properties that provide the optimal complexity result.

LEMMA 4.4. *Suppose that Assumption **AS.1** holds. Then, at any successful iteration k of the FAR2 algorithm, if $k \in \mathcal{I}_{RN}$ then*

$$\|\hat{s}_k\|^2 \geq \frac{2}{C_{\text{up}}(L_1 C_{\text{up}} + \sigma_{\max})} \|g_{k+1}\|.$$

Moreover, at any successful iteration, the following inequality holds

$$(4.5) \quad f(x_k) - f(x_{k+1}) \geq \eta_1 \sigma_{\min} \min \left\{ C_{\text{low}}^2 \kappa_0, \frac{1}{3} \kappa_1 \right\} \|g_{k+1}\|^{3/2},$$

where $\kappa_0 = \left(\frac{2}{C_{\text{up}}(L_1 C_{\text{up}} + \sigma_{\max})} \right)^{3/2}$, $\kappa_1 = \left(\frac{2}{L_1 + \theta_1 + \sigma_{\max}} \right)^{3/2}$.

Proof. Let us consider the case $k \in \mathcal{I}_{RN}$. By using the definition of s_k in Line 9 of Algorithm 3.1, (3.1), and (4.4) it follows that

$$\begin{aligned}
\|g_{k+1}\| &\leq \|g_{k+1} - (H_k s_k + g_k)\| + \|H_k s_k + g_k\| \\
&\leq \frac{L_1}{2} \|s_k\|^2 + \frac{1}{2} \sigma_k \|\widehat{s}_k\| \|s_k\| \\
(4.6) \quad &\leq \frac{C_{\text{up}}}{2} (L_1 C_{\text{up}} + \sigma_{\text{max}}) \|\widehat{s}_k\|^2.
\end{aligned}$$

Moreover, (4.2) yields

$$\begin{aligned}
f(x_k) - f(x_{k+1}) &\geq \eta_1 (T_k(0) - T_k(s_k)) \\
&\geq \eta_1 \sigma_k \|\widehat{s}_k\| \|s_k\|^2 \geq \eta_1 \sigma_{\text{min}} C_{\text{low}}^2 \|\widehat{s}_k\|^3 \geq \eta_1 \sigma_{\text{min}} C_{\text{low}}^2 \kappa_0 \|g_{k+1}\|^{3/2},
\end{aligned}$$

where $\kappa_0 \stackrel{\text{def}}{=} \left(\frac{2}{C_{\text{up}}(L_1 C_{\text{up}} + \sigma_{\text{max}})} \right)^{3/2}$.

In case $k \notin \mathcal{I}_{RN}$, Lemma 3.3.3 in [9] ensures that the step s_k computed in Line 6 of Algorithm 3.1 satisfies:

$$\|g_{k+1}\| \leq \frac{L_1 + \theta_1 + \sigma_{\text{max}}}{2} \|s_k\|^2,$$

and from (4.3) it follows that

$$f(x_k) - f(x_{k+1}) \geq \eta_1 (T_k(0) - T_k(s_k)) \geq \eta_1 \frac{1}{3} \sigma_k \|s_k\|^3 \geq \eta_1 \frac{1}{3} \sigma_{\text{min}} \kappa_1 \|g_{k+1}\|^{3/2},$$

where $\kappa_1 \stackrel{\text{def}}{=} \left(\frac{2}{L_1 + \theta_1 + \sigma_{\text{max}}} \right)^{3/2}$. \square

Proceeding as in Lemma 3.3.4 and Theorem 3.3.5 in [9] we prove the following iteration complexity result showing that FAR2 preserves the optimal complexity of AR2.

THEOREM 4.5. *Suppose that **AS.1** and **AS.2** hold. Given $\epsilon \in (0, 1)$, there exists a constant κ_p such that the FAR2 algorithm needs at most*

$$\left\lceil \frac{f(x_0) - f_{\text{low}}}{\kappa_p} \epsilon^{-\frac{3}{2}} + 1 \right\rceil,$$

successful iterations and at most

$$\left\lceil \frac{f(x_0) - f_{\text{low}}}{\kappa_p} \epsilon^{-\frac{3}{2}} + 1 \right\rceil \left(1 + \frac{|\log \gamma_1|}{\log \gamma_2} \right) + \frac{1}{\log \gamma_2} \log \left(\frac{\sigma_{\text{max}}}{\sigma_0} \right),$$

iterations to compute an iterate x_k such that $\|g_k\| \leq \epsilon$.

Proof. At each successful iteration k the algorithm guarantees the decrease in (4.5). Then,

$$(4.7) \quad f(x_k) - f(x_{k+1}) \geq \eta_1 \sigma_{\text{min}} \min \left\{ C_{\text{low}}^2 \kappa_0, \frac{1}{3} \kappa_1 \right\} \epsilon^{3/2},$$

whenever $\|g_{k+1}\| > \epsilon$. Thus, letting \bar{k} such that $\bar{k} + 1$ is the first iteration where $\|g_{\bar{k}+1}\| \leq \epsilon$ and summing up from $k = 0$ to $\bar{k} - 1$, since $f(x_k) = f(x_{k+1})$ along unsuccessful iterations, and using the telescopic sum property, we obtain

$$f(x_0) - f(x_{\bar{k}}) \geq \kappa_p \epsilon^{\frac{3}{2}} |\mathcal{S}_{\bar{k}-1}|,$$

where $\kappa_p = \eta_1 \sigma_{\text{min}} \min \left\{ C_{\text{low}}^2 \kappa_0, \frac{1}{3} \kappa_1 \right\}$.

Since f is bounded from below by f_{low} , we conclude that

$$(4.8) \quad |\mathcal{S}_{\bar{k}}| \leq \frac{f(x_0) - f_{\text{low}}}{\kappa_p} \epsilon^{-\frac{3}{2}} + 1.$$

Lemma 4.1 then yields the upper bound on the total number of iterations. \square

4.1. Complexity of finding second-order optimality points. We now provide an upper bound on the number of iterations needed by FAR2-SO to compute an (ϵ, ϵ_H) -approximately second-order minimizer.

THEOREM 4.6. *Suppose that **AS.1** and **AS.2** hold. Given $\epsilon, \epsilon_H \in (0, 1)$, there exist positive constants $\kappa_{H,1}$ and $\kappa_{H,2}$ such that the FAR2-SO algorithm needs at most*

$$\left\lceil \frac{f(x_0) - f_{\text{low}}}{\kappa_{H,1}} \max\{\epsilon^{-\frac{3}{2}}, \epsilon_H^{-2}\} + 1 \right\rceil,$$

successful iterations and at most

$$\left\lceil \frac{f(x_0) - f_{\text{low}}}{\kappa_p} \max\{\epsilon^{-\frac{3}{2}}, \epsilon_H^{-2}\} + 1 \right\rceil \left(1 + \frac{|\log \gamma_1|}{\log \gamma_2} \right) + \frac{1}{\log \gamma_2} \log \left(\frac{\sigma_{\max}}{\sigma_0} \right),$$

iterations to compute an iterate x_k such that $\|g_k\| \leq \epsilon$ and $\lambda_{\min}(H_k) > -\epsilon_H$.

Proof. For any successful iteration k with $k \in \mathcal{I}_{RN}$ it holds that s_k has been computed as in Line 9 of Algorithm 3.1. Then,

$$\lambda_{\min}(H_k) > -\hat{\lambda}.$$

Therefore, by using **AS.1** and Lemma 3.3, for any successful iteration k with $k \in \mathcal{I}_{RN}$ and for any $d \in \mathbb{R}^n$, $\|d\|_2 = 1$, we obtain

$$\begin{aligned} d^T H_{k+1} d &= d^T (H_{k+1} - H_k) d + d^T H_k d \geq -L_H \|s_k\| - \hat{\lambda} \\ &= -L_H C_{\text{low}} \|\hat{s}_k\| - \sigma_k \|\hat{s}_k\| \geq -(L_H C_{\text{low}} + \sigma_{\max}) \|\hat{s}_k\|. \end{aligned}$$

This latter inequality along with Lemma 4.4 yield

$$(4.9) \quad \|\hat{s}_k\| \geq \max\{\kappa_0^{1/3} \|g_{k+1}\|^{1/2}, -\kappa_{H,0}^{1/3} \lambda_{\min}(H_{k+1})\},$$

where $\kappa_{H,0} = \left(\frac{1}{L_H C_{\text{low}} + \sigma_{\max}} \right)^3$ and κ_0 has been defined as in Lemma 4.4. Then, by

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \eta_1 (T_k(0) - T_k(s_k)) \\ &\geq \eta_1 \sigma_k \|\hat{s}_k\| \|s_k\|^2 \geq \eta_1 \sigma_{\min} C_{\text{low}}^2 \|\hat{s}_k\|^3, \end{aligned}$$

we get that for any successful iteration k with $k \in \mathcal{I}_{RN}$ it holds

$$f(x_k) - f(x_{k+1}) \geq \eta_1 \sigma_{\min} C_{\text{low}}^2 \max\{\kappa_0 \|g_{k+1}\|^{3/2}, -\kappa_{H,0} \lambda_{\min}(H_{k+1})^3\}.$$

In case $k \notin \mathcal{I}_{RN}$, Lemma 3.3.3 in [9] ensures that the step s_k computed in Line 6 of Algorithm 3.1 satisfies:

$$\|s_k\| \geq -\frac{1}{L_H + \theta_2 + \sigma_{\max}} \lambda_{\min}(H_{k+1}).$$

Then, from (4.3) and (4.6) it follows that

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \eta_1 (T_k(0) - T_k(s_k)) \\ &\geq \eta_1 \sigma_{\min} \min \left\{ \kappa_1 \|g_{k+1}\|^{3/2}, -\kappa_{H,1} \lambda_{\min}(H_{k+1})^3 \right\}, \end{aligned}$$

where $\kappa_{H,1} = \left(\frac{1}{L_H + \theta_2 + \sigma_{\max}} \right)^{1/3}$.

Finally, for any iteration k such that the FAR2-SO algorithm has not terminated before or at iteration $k+1$, we must have that either $\|g_{k+1}\| > \epsilon$ or $\lambda_{\min}(H_{k+1}) < -\epsilon_H$. Therefore, at any successful iteration k we have

$$f(x_k) - f(x_{k+1}) \geq \eta_1 \sigma_{\min} \min \left\{ \kappa_p \epsilon^{3/2}, -\kappa_H \epsilon_H^3 \right\},$$

with $\kappa_p = \min\{\kappa_0, \kappa_1\}$ and $\kappa_H = \min\{\kappa_{H,0}, \kappa_{H,1}\}$. The thesis then follows proceeding as in the proof of Theorem 4.5. \square

5. Choice of the approximation space. The performance of Algorithm 3.1 strongly depends on the chosen approximation space $\mathcal{K}_{d_k} = \text{Range}(V_k)$ used, along with g_k , for projecting the problem. For given H_k and g_k , Krylov subspaces have proven to be well suited dimension reduction tools for eigenvalue problems [35, Chapter 9], matrix-function evaluations [28, Chapter 13], matrix equations [33], and, more generally, in model order reduction techniques [1]. Our context is more involved. Indeed, both the matrix H_k and the generating vector g_k vary as the outer nonlinear iterations proceed.

We aim at achieving two main goals: i) satisfy (2.6) with a projection space as small as possible; ii) generate a space that can be effectively used in later outer iterations, with different but slowly varying H_k and g_k . To this end, we propose to employ *rational* Krylov subspaces. These spaces are generally able to capture good spectral information on a given problem already for small dimensions, compared to classic polynomial Krylov subspaces, thus significantly reducing the original problem size. Hence, our technique will likely be able to attain (2.6) in few iterations for given H_k and g_k . Moreover, we expect the rational Krylov subspace generated by H_k and g_k to still be a successful approximation space for successive nonlinear iterations, especially if the spectral properties of H_k slowly change with k .

Given a suitable set of shifts $z_{d_k} = (\xi_1, \dots, \xi_{d_k}) \in \mathbb{R}^{d_k}$, the rational Krylov subspace generated by the symmetric matrix H_k and the vector g_k is defined as follows³

$$(5.1) \quad \mathcal{K}_{d_k}(H_k, g_k, z_{d_k}) = \text{span} \left\{ (H_k + \xi_1 I)^{-1} g_k, \dots, \prod_{j=1}^{d_k} (H_k + \xi_j I)^{-1} g_k \right\}.$$

An orthonormal basis $V_k \in \mathbb{R}^{n \times d_k}$ of \mathcal{K}_{d_k} can be constructed by means of the rational Arnoldi method [32]. Starting with $v_1 = (H_k + \xi_1 I)^{-1} g_k$, normalized to have unit norm, the next basis vector is obtained by first computing $\tilde{v}_2 = (H_k + \xi_2 I)^{-1} v_1$, orthogonalizing \tilde{v}_2 with respect to the previous vectors, in this case only v_1 , and finally normalizing to have unit norm. The procedure thus continues iteratively.

The major computational cost per iteration of the rational Krylov subspace basis construction amounts to the solution of a shifted linear system with H_k . Therefore, it is crucial that the number of performed iterations $d_k \leq j_{\max}$ remains small to limit computational costs. To this end, the choice of the shifts is crucial. Quasi-optimal and adaptive procedures have been proposed, where the shifts are computed automatically; see, e.g., [27, 19, 17] and the references therein. In case H_k is semi-definite, as it is expected to occur for large enough k , then the nonzero shifts are chosen to have the same signature of H_k so that $H_k + \xi_j I$ results in a positive definite matrix. On the other hand, if H_k is slightly indefinite as it may happen at the beginning of the outer nonlinear procedure, the shifts can be selected to ensure that $H_k + \xi_j I$ is still nonsingular. For instance, whenever second order optimality points are sought and Algorithm A.1 is employed, $\lambda_{\min}(H_k)$ is computed. Such a piece of information can be exploited also in the basis construction by, e.g., using shifts of the form $-\lambda_{\min}(H_k) + \xi$, $\xi > 0$.

The adaptive shift selection procedure described in [19] has been adopted to obtain all the numerical results reported in section 6. Notice that the shifts ξ_j are not necessarily related to the parameters $\hat{\lambda}_k$. In particular, the former values must be such that $H^{(j)} = W^{(j)T} H_k W^{(j)}$, $W^{(j)} = \text{orth}([V_k, g_k])$, is a small, yet meaningful, representation of H_k .

At the first nonlinear iteration of Algorithm 3.1 we thus construct $V_1 \in \mathbb{R}^{n \times d_1}$

³For later convenience, we do not include g_k in the space definition, following the typical setting in model order reduction. Nonetheless, the vector g_k is included in the projection phase, see Algorithm 3.2.

such that $\text{Range}(V_1) = \mathcal{K}_{d_1}(H_1, g_1, z_{d_1})$. As highlighted in Algorithm 3.2, this space is expanded until (2.6) is satisfied. At this point we proceed with Step 3 in Algorithm 3.1. At the next nonlinear iteration, namely for $k = 2$, we set $V_2 = V_1$ and project the current secular equation onto $\text{Range}([V_2, g_2])$. If the computed $\widehat{\lambda}_2$ is such that $H_2 + \widehat{\lambda}_2 I$ is not positive definite or (3.1) does not hold, we discard V_2 and construct a new rational Krylov subspace $\mathcal{K}_{d_3}(H_3, g_3, z_{d_3}) = \text{Range}(V_3)$. Otherwise, we set $V_3 = V_2$. The scheme then continues in the same fashion.

We would like to mention that the performance of Algorithm 3.1 largely benefits from the inclusion of the current g_k in the approximation space $\text{Range}([V_k, g_k])$ as g_k is exactly represented during the projection phase. In particular, this exact representation is a crucial hypothesis for obtaining the bounds on $\|s_k\|$ reported in (3.1).

REMARK 5.1. *The action of the rational Krylov reduction on the approximate solution of the given linear system at each nonlinear iteration can be appreciated by resorting to general results on rational Krylov subspaces⁴. To this end, we assume:*

H1: there exists λ_{th} such that $H + \lambda_{\text{th}} I > 0$, and the parameter λ_{j+1}^ such that $\phi_R(\lambda_{j+1}^*; g^{(j)}, H^{(j)}, \sigma_k) = 0$ satisfies $\lambda_{j+1}^* \geq \lambda_{\text{th}}$; $H^{(j)}$ and $g^{(j)}$ denote the projection of H and g onto the current subspace, respectively.*

To emphasize the role of the parameter λ_{j+1}^ , we explicitly report the procedure in Algorithm 3.2 (`refresh` = 1) in the case of the rational Krylov subspace (5.1).*

Given $V_1 = v_1 = (H + \xi_1^ I)^{-1} g$ with $H + \xi_1^* I > 0$, λ_1^* , and ξ_{\min}, ξ_{\max} ,*

For $j = 1, 2, \dots$,

- 1. Compute projection: $H^{(j)} = V_j^T H V_j$, $g^{(j)} = V_j^T g$*
- 2. Find λ_{j+1}^* such that $\|(H^{(j)} + \lambda_{j+1}^* I)^{-1} g^{(j)}\| = \lambda_{j+1}^* / \sigma$*
- 3. Check stopping criterion*
- 4. Compute \mathcal{I}_{j+1} smallest interval containing ξ_{\min}, ξ_{\max} and $\{\theta_1^{(j)}, \dots, \theta_j^{(j)}\} = \text{eig}(H^{(j)} + \lambda_{j+1}^* I)$*
- 5. For $\psi_j(\xi) = \prod_{\ell=1}^j (\xi - \theta_\ell^{(j)}) / (\xi + \xi_\ell)$, find next $\xi_{j+1} \in \mathcal{I}_{j+1}$ such that $\xi_{j+1} = \arg \max_\xi |1/\psi_j(-\xi)|$*
- 6. Compute next basis vector $v_{j+1} = (H + \xi_{j+1} I)^{-1} v_j$ and update $V_{j+1} = \text{orth}([V_j, v_{j+1}])$*

This procedure is a variant of the greedy method in [18] for adaptively selecting the shifts $\{\xi_j\}$ during the generation of the rational Krylov subspace. The two quantities ξ_{\min}, ξ_{\max} are chosen so that the interval $[\xi_{\min}, \xi_{\max}]$ encloses the spectra of all shifted matrices $H^{(j)} + \lambda_{j+1}^ I$.*

Let $\psi_j(\xi) = \prod_{\ell=1}^j (\xi - \theta_\ell^{(j)}) / (\xi + \xi_\ell)$ as in the procedure above. Let $y_j = V_j (H^{(j)} + \lambda_{j+1}^ I)^{-1} g^{(j)}$ be the approximate solution at the end of iteration j . The associated linear system residual is given by $r_{j+1} = g - (H + \lambda_{j+1}^* I) y_j = \psi_j (H + \lambda_{j+1}^* I) g$ and $\nabla m_k(y_j) = r_{j+1}$. Since our aim is to satisfy (2.6), we are indeed interested in the behaviour of the residual norm.*

According to the theoretical analysis in [16], and to the discussion in [18], under H1 we can expect that the norm of the residual r_{j+1} satisfies

$$\|r_{j+1}\|^{1/j} \approx \exp \left(-\frac{\pi^2}{2} \left(\log \frac{4\lambda_{\max}}{\lambda_{\min}} \right)^{-1} \right),$$

where $\lambda_{\max}, \lambda_{\min}$ are the extreme eigenvalues of $H + \lambda_{\text{th}} I > 0$, where we note that $\kappa := \lambda_{\max}/\lambda_{\min}$ is larger than the corresponding ratio of the extreme eigenvalues of

⁴We drop the nonlinear iteration index k for better readability.

$H + \lambda I$ with $\lambda > \lambda_{\text{th}}$. The asymptotic estimate above should be compared with that obtained with the conjugate gradient method for the same matrix, namely the ratio $(\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$; see, e.g., [24, Corollary 3.7]. The following table reports both convergence rates as κ increases.

κ	10^2	10^3	10^4	10^5	10^6	10^7	10^8	10^9	10^{10}
CG	0.818	0.938	0.980	0.993	0.9980	0.9994	0.9998	0.9999	1.0000
RatKr	0.438	0.551	0.627	0.682	0.7228	0.7543	0.7795	0.8000	0.8170

While the conjugate gradient rate rapidly reaches 1, thus predicting complete stagnation of the method, the rate associated with the ideal rational Krylov subspace remains very moderate, allowing a residual decay also for very large condition numbers.

In terms of our frozen technique, the arguments above show why we expect to construct a low-dimensional subspace \mathcal{K}_{d_k} whenever a refresh in Algorithm 3.1 is performed. In particular, the constructed rational Krylov subspace is likely to be much smaller than its polynomial counterpart, especially for ill-conditioned problems. See section 6.

REMARK 5.2. As outlined in [9, Section 10.1.2], (polynomial) Krylov methods are not able to easily handle the “hard case” mentioned in section 2.1. Rational Krylov subspace methods are no exception. Indeed, if the leftmost eigenvalue λ_1 of H_k is negative and g_k is orthogonal to the λ_1 -eigenspace of H_k , then, for any scalar ξ , the vector $(H_k + \xi I)^{-1}g_k$ is also orthogonal to the λ_1 -eigenspace. This means that the vector v_1 in Theorem 2.2 cannot have components in the rational Krylov subspace $\mathcal{K}_{d_k}(H_k, g_k, z_{d_k})$. Even though the occurrence of the hard case is very unlikely, adaptations of the strategies developed for polynomial spaces [9, Section 10.1.2] should be considered also in our case.

For nonlinear iterations in which the space $\text{Range}(V_k)$ is frozen and a refresh is not performed, the hard case is even less likely to take place, although still possible. Indeed, the hard-case occurs when both g_k and the entire $\text{Range}(V_k) = \mathcal{K}_{d_\ell}(H_\ell, g_\ell, z_{d_\ell})$, $\ell < k$, are orthogonal to the λ_1 -eigenspace of H_k . Since $\text{span}\{g_k\} \not\subset \text{Range}(V_k)$, it can still happen that $\text{Range}(W_k) = \text{Range}([V_k, g_k])$ is not orthogonal to the λ_1 -eigenspace of H_k , even when g_k is so.

6. Computational experiments. This section is devoted to the illustration of the computational performance of our new algorithm. First we introduce some implementation details and the set of test problems. Secondly, we compare FAR2 against the standard implementation of AR2 based on direct solvers. At this point, we adopt the rational Krylov subspace as approximation space as suggested in section 5 (FAR2-RK). However, the FAR2 framework is designed for general approximation spaces. As a sample, we also display the performance of FAR2 by adopting polynomial Krylov subspaces (FAR2-PK). To conclude, we compare the results attained by the latter routines to the ones achieved by classic Lanczos-based versions of the AR2 method.

6.1. Implementation details. We implemented the FAR2 method detailed in Algorithms 3.1-3.2 in Matlab⁵ [34] and we used the following standard values for the parameters

$$(6.1) \quad \eta_1 = 0.1, \quad \eta_2 = 0.8, \quad \gamma_1 = 0.1, \quad \gamma_2 = 2, \quad \theta_1 = 0.1, \quad \sigma_{\min} = 10^{-8},$$

and

$$C_{\text{low}} = 10^{-20}, \quad C_{\text{up}} = 10^{20}.$$

The solution of the secular equation projected onto $\text{Range}(W_k)$ (Lines 6 and 16 of Algorithm 3.2) is performed by using the Matlab function `fzero` providing

⁵MATLAB R2019b on a Intel Core i7-9700K CPU @ 3.60GHz x 8, 16 GB RAM, 64-bit.

$[\min(-\lambda_{\min}(H^{(j)}), 0), \min(-\lambda_{\min}(H^{(j)}), 0) + 1000]$ as starting interval for the bisection method.

Regarding the implementation of FAR2-RK and FAR2-PK, we set a maximum subspace dimension $j_{\max} = 50$ for the former and $j_{\max} = n$ for the latter. The linear system solutions involved in the rational Krylov basis construction are carried out by means of the Matlab backslash operator. Moreover, the positive definiteness of $H_k + \hat{\lambda}_k I$ is checked by attempting its Cholesky factorization in Line 8 of Algorithm 3.1. In case of success, such factorization is exploited for computing s_k in Line 9. We remark that the Cholesky factorization might be replaced by the modified Cholesky decomposition without additional cost, see e.g. [12]. This would ensure that the corresponding regularized Newton step s_k is a descent direction thus avoiding the possible step rejection in Line 11 of Algorithm 3.1 (and subsequent subspace refresh). Unfortunately, this strategy would deteriorate the optimal worst-case complexity bound of FAR2 and therefore we decided not to employ it.

For an indefinite Hessian matrix H_k , the shifted matrix $H_k + \xi_j I$ employed in the rational Krylov subspace basis construction may be singular. Although in some of the instances $H_k + \xi_j I$ turned out to be ill-conditioned, in our extensive experimental testing this fact did not seem to severely affect the quality of the computed subspace.

We compare the performance of our new method against classic implementations of Algorithm 2.1 that make use the parameter values in (6.1). In particular, in such implementations, Step 2 of Algorithm 2.1 has been implemented using the state-of-the-art solvers included in the GALAHAD optimization library (version 2.3)⁶: RQS and GLRT for the implementations based on direct and iterative solvers, respectively. We refer to the corresponding overall implementations as AR2-RQS and AR2-GLRT. We used the Matlab interface and default parameters except for the values `control.stop_normal` = 0.1 and `control.stop_relative` = 0.1, in RQS and GLRT, respectively.

All algorithms terminate when one of the following conditions holds: (i) $\|g_k\| \leq \tau \|g_0\|$, where $\tau > 0$ depends on the problem test set; (ii) 5000 iterations are performed; (iii) the computational time limit of 2 hours is reached.

6.2. Test problems. We consider the test problems in the OPM testing set [26]. The latter contains a small collection of CUTEst unconstrained and bound-constrained nonlinear optimization problems. From all problems listed in [26] we selected the unconstrained ones. For those problems allowing for a variable dimension n , we set $n = 1000$ (or nearby) as specified in Table 6.1.

As testing problems, we also considered binary classification problems. We suppose we have at our disposal a training set composed of pairs $\{(a_i, b_i)\}$ with $a_i \in \mathbb{R}^n$, $b_i \in \{-1, 1\}$ (or $b_i \in \{0, 1\}$) and $i = 1, \dots, N$, where b_i denotes the correct sample classification. We perform both a logistic regression and a sigmoidal regression, yielding convex and non convex problems, respectively.

In the first case, we consider as training objective function the logistic loss with ℓ_2 regularization [2, 5], i.e.

$$f(x) = \sum_{i=1}^N \log \left(1 + e^{-b_i a_i^T x} \right) + \lambda \|x\|_2^2,$$

with $b_i \in \{-1, 1\}$. In the second case the following sigmoid function and least-squares loss is employed [3, 36]

$$f(x) = \sum_{i=1}^N \left(b_i - \frac{1}{1 + e^{-a_i^T x}} \right)^2,$$

⁶Available from <http://galahad.rl.ac.uk/galahad-www/>.

Problem	n	Problem	n	Problem	n	Problem	n
ARGLINA	1000	DIXMAANC	3000	ENGVAL1	1000	PENALTY3	1000
ARGLINB	1000	DIXMAAND	3000	ENGVAL2	1000	POWELLSG	1000
ARGLINC	1000	DIXMAANE	3000	EXTROSNB	1000	POWR	1000
ARGTRIG	1000	DIXMAANF	3000	FMINSURF	900	ROSENBR	1000
ARWHEAD	1000	DIXMAANG	3000	FREUROTH	1000	SCOSINE	1000
BDARWHD	1000	DIXMAANH	3000	GENHUMPS	1000	SCURLY10	1000
BROWNAL	1000	DIXMAANI	3000	HELIX	1000	SCURLY20	1000
BROYDENBD	1000	DIXMAANJ	3000	HILBERT	1000	SCURLY30	1000
CHANDHEU	1000	DIXMAANK	3000	INDEF	1000	SENSORS	1000
CHEBYQAD	1000	DIXMAANL	3000	INTEGREQ	1000	SPMSQRT	1000
COSINE	1000	DIXON	1000	MANCINO	1000	TQUARTIC	1000
CRGLVY	1000	DQRTIC	1000	MSQRTALS	900	TRIDIA	1000
CUBE	1000	EDENSCH	1000	MSQRTBLS	900	VARDIM	1000
CURLY10	1000	EG2	1000	NONDIA	1000	WMSQRTALS	900
CURLY20	1000	EG2S	1000	NONDQUAR	1000	WMSQRTBLS	900
CURLY30	1000	EIGENALS	1056	NZF1	1300	WOODS	1000
DIXMAANA	3000	EIGENBLS	1056	PENALTY1	1000		
DIXMAANB	3000	EIGENCLS	1056	PENALTY2	1000		

TABLE 6.1
The OPM test problems and their dimension.

with $b_i \in \{0, 1\}$. We used the six data sets⁷ reported in Table 6.2 together with the dimension n of the parameter vector $x \in \mathbb{R}^n$.

For OPM and binary classification problems, we ran the algorithms until $\|g_k\| \leq \tau \|g_0\|$ with $\tau = 10^{-6}$ and $\tau = 10^{-3}$, respectively.

6.3. Numerical results. We now discuss the performance of the FAR2 method when the rational Krylov subspace are employed as approximation subspace as described in section 5. In particular, we compare FAR2-RK with AR2-RQS in terms of number of factorizations and nonlinear iterations. We have therefore generated the corresponding performance profiles in Figure 6.1 related to the OPM problems⁸ listed in Table 6.1. We remind the reader that a performance profile graph $p_A(\tau)$ of an algorithm A at point τ shows the fraction of the test set for which the algorithm is able to solve within a factor of τ of the best algorithm for the given measure [15].

The profiles in Figure 6.1 clearly show a great advantage in using FAR2-RK in terms of total number of matrix factorizations with respect to AR2-RQS, with FAR2-RK being the most efficient routine 84% of the runs. The behavior in terms of number of nonlinear iterations is similar, while AR2-RQS is slightly more robust solving 4 problems more than FAR2-RK⁹.

Similar results are obtained on the binary classification problems. Table 6.2 collects the results distinguishing between the logistic and the sigmoidal loss function cases for the 6 data sets we tested. In spite of the larger number of nonlinear iterations ($\#NLI$) performed by FAR2-RK in few cases, we remark that the overall number of factorizations always remains much in favour of FAR2-RK when compared to AR2-RQS.

The typical behavior of the two methods is illustrated in Figure 6.2 and Figure 6.3. In both cases, we observe that the FAR2-RK first iteration requires many factorizations as these are needed in the rational Krylov basis construction. However, our novel approach is able to fully capitalize on this initial computational efforts and the subspace-freezing strategy typically leads to a single factorization per nonlinear iteration (see Figure 6.2) or even no factorizations at all if the approximate solution $s_k = W_k \hat{s}_k$ to the secular equation satisfies (2.6) in Line 6 of Algorithm 3.1 (see Figure 6.3). This happens for several nonlinear iterations.

⁷A9A [11], CINA0 [11], GISETTE [11], MNIST [30], MUSH [11], REGERD0 [10]

⁸All the tested algorithms failed in solving the very ill-conditioned problems ARGLINB, ARGLINC, COSINE, GENHUMPS and SCOSINE. Therefore, these problems have been excluded from the testing set.

⁹All failures occurred because the maximum number of iteration limit was reached.

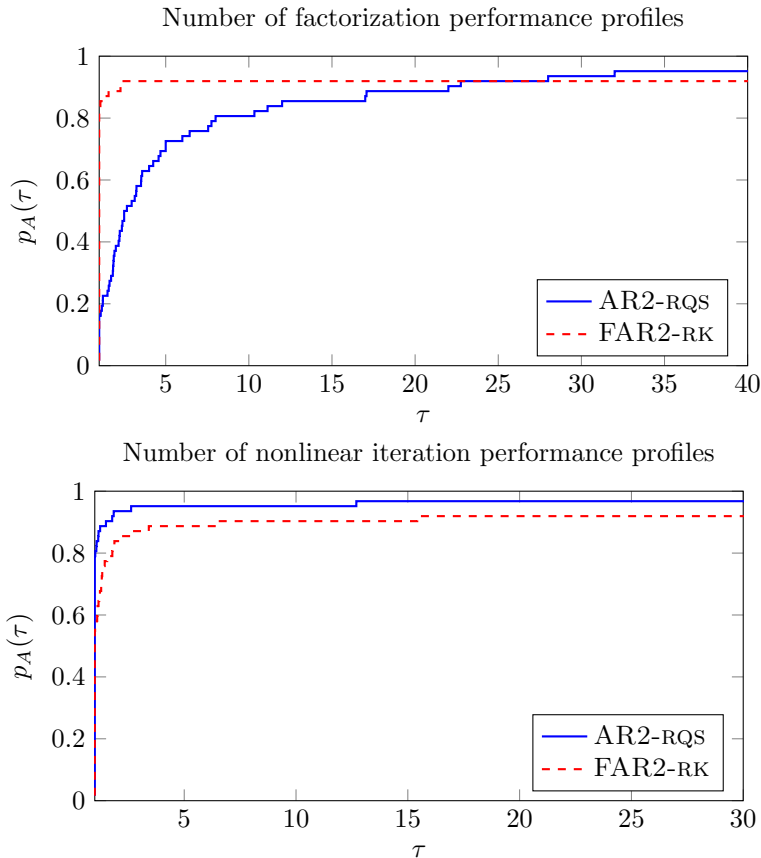


FIG. 6.1. Performance profiles of AR2-RQS and FAR2-RK for the OPM set.

The sigmoidal case in Figure 6.3 shows an unusual trend of FAR2-RK where many more nonlinear iterations are performed compared to what happens for AR2-RQS. It is clear from the plot that the refresh strategy is invoked repeatedly. On the other hand, a very low-dimensional subspace is constructed at each refresh thus keeping the number of factorizations low.

Finally, we observe that for strictly convex problems, as those obtained by using the logistic loss in classification problems, the refresh strategy is never activated as, in fact, $H_k + \lambda I$ is positive definite for any $\lambda \geq 0$ and the condition in Line 13 of Algorithm 3.1 is never met.

Leveraging on the discussion in section 5, we next computationally show the benefits of using the rational Krylov (RK) subspace as approximation space in Algorithm 3.2. To this purpose, we compare FAR2-RK with FAR2-PK where standard polynomial Krylov (PK) subspaces are used in Algorithm 3.2. We remark that FAR2-PK does not require any factorization in the construction of the subspace. On the other hand, since the computed orthonormal basis needs to be stored for later nonlinear iterations, PK cannot take advantage of the symmetry in the Hessian matrix and avoid the possibly large memory requirements.

Figure 6.4 displays the CPU time performance profile of both FAR2-RK and FAR2-PK for the OPM set, showing that the two implementations perform similarly for most of the problem but for few of them (about 10%) FAR2-RK is much faster. As a representative example of these cases, we report in Figure 6.5 the results obtained for the EIGENALS problem from OPM: for each nonlinear iteration the dimension of the generated Krylov subspace is plotted. We can observe that the

logistic loss		# factorizations		#NLI	
data set(n)	AR2-RQS	FAR2-RK	AR2-RQS	FAR2-RK	
A9A (123)	16	5	7	8	
CINA0 (132)	36	17	7	7	
GISETTE (5000)	24	8	7	8	
MNIST (784)	22	7	8	8	
MUSH (112)	20	3	9	19	
REGED0 (999)	17	12	8	7	

sigmoidal loss		# factorizations		#NLI	
data set(n)	AR2-RQS	FAR2-RK	AR2-RQS	FAR2-RK	
A9A (123)	19	7	8	18	
CINA0 (132)	49	16	7	8	
GISETTE (5000)	58	16	9	9	
MNIST (784)	116	80	16	57	
MUSH (112)	27	3	10	33	
REGED0 (999)	14	12	7	6	

TABLE 6.2

Results on binary classification problems: number of factorization (# factorizations) and number of nonlinear iterations (#NLI) for the AR2-RQS and the FAR2-RK methods.

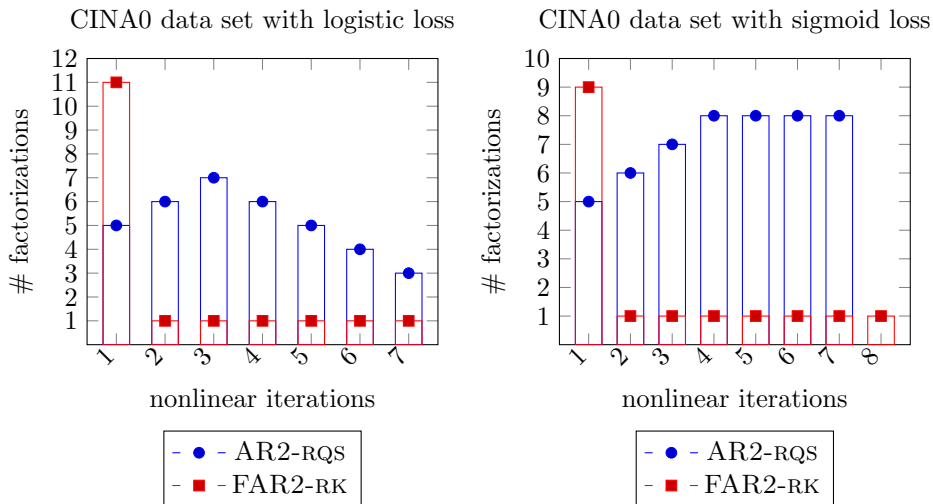


FIG. 6.2. Number of factorizations per nonlinear iteration for the CINA0 data set.

dimension of the PK subspace reaches 60% of the problem dimension n so that the cost of computing the projected step \hat{s}_k in Algorithm 3.2 considerably increases. On the contrary, the dimension of the RK subspace stays below 12% of n making the cost of computing \hat{s}_k negligible. In addition to leading to a more expensive computation of \hat{s}_k , the large space dimension reached in FAR2-PK has a remarkable impact on the memory requirements and on the cost of the orthogonalization step carried out during the PK basis construction. As a consequence, the overall running time of FAR2-RK and FAR2-PK is 5.1 and 1199.2 seconds, respectively, to solve the EIGENALS problem.

Finally, we have compared FAR2-RK with AR2-GLRT that is the most used implementation of AR2 based on the Lanczos iteration [8]. Considering the 67 OPM problems, for most runs the dimension of the PK generated by AR2-GLRT is rather small making the performance of FAR2-RK and AR2-GLRT comparable. Nevertheless, we report in Table 6.3 problems where the dimension of PK becomes

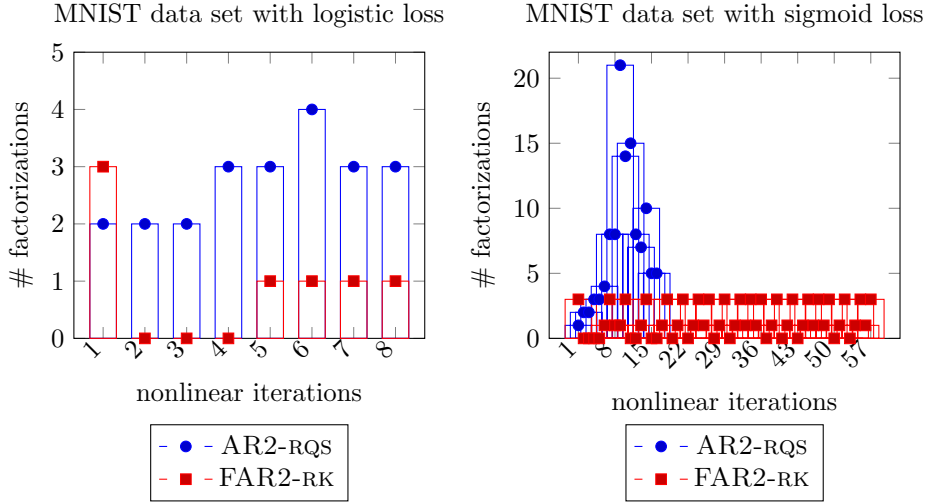


FIG. 6.3. Number of factorizations per nonlinear iteration for the MNIST data set.

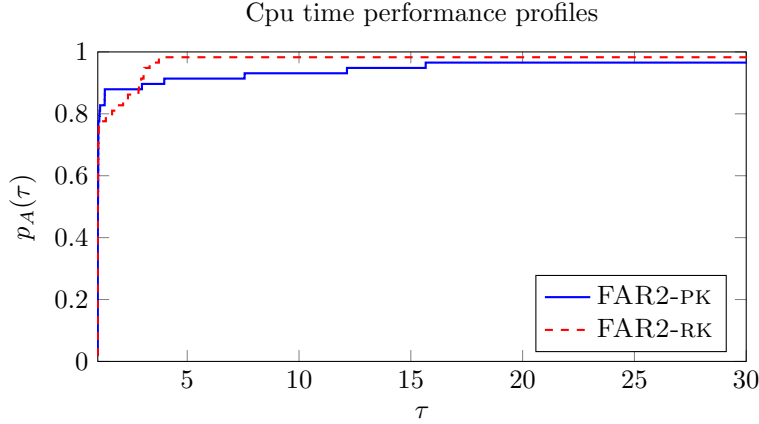


FIG. 6.4. Cpu time performance profiles of FAR2-RK and FAR2-PK for the OPM set.

larger than the 50% of n . The table reports the maximum dimension of the generated Krylov subspace max_K , the average value ave_K and the overall CPU time in seconds ($cpu*$)¹⁰. For FAR2-RK the value of ave_K is computed by taking into account the number of nonlinear iterations where the subspace is refreshed. The results confirm that, as expected from the discussion in Remark 5.1, FAR2-RK results in a valid option to solve problems for which AR2-GLRT struggles.

7. Discussion and conclusions. For the practical use of the adaptive regularization method, the efficient minimization of the cubic regularized model is a crucial issue. We have addressed this bottleneck by proposing a new two-step procedure, the FAR2 method. The main difference between FAR2 and standard adaptive-regularization techniques is in the minimization of the involved model. In our novel scheme, the latter is carried out over a possibly frozen subspace. Under mild conditions, we have proved that the optimal worst-case complexity of the original AR2 method is preserved, regardless of the adopted subspace. To steer computational effectiveness, we have used rational Krylov subspaces for the sub-

¹⁰We remark that we report the cpu time for illustrative purposes only, as AR2-GLRT uses optimized and compiled functions written in Fortran.

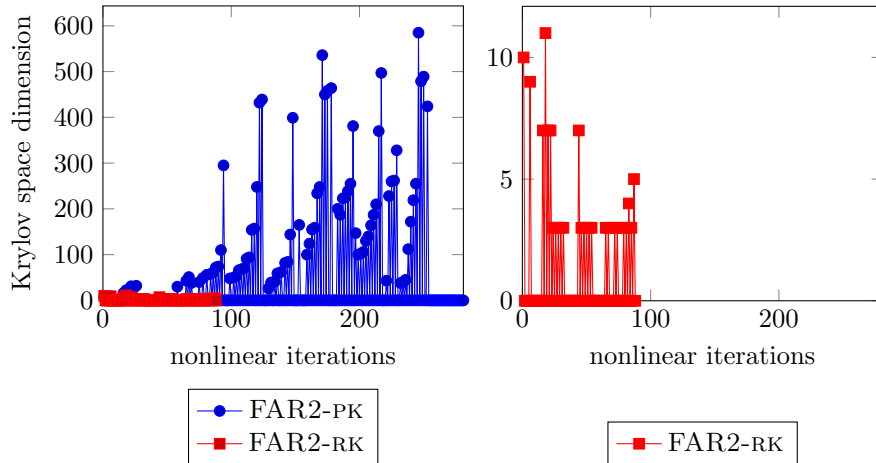


FIG. 6.5. Dimension of the Krylov subspace versus the nonlinear iterations for the EIGENALS problem in OPM: FAR2-RK and FAR2-PK (left), zoom of FAR2-RK (right).

Problem	AR2-GLRT			FAR2-RK		
	max_K	ave_K	cpu^*	max_K	ave_K	cpu^*
CURLY10	1000	400.9	2.8	50	25.3	2.3
CURLY20	1000	325.2	2.7	50	19.8	2.5
CURLY30	1000	273.9	3.1	50	30.5	3.7
DIXON	920	361.0	0.7	3	3.0	0.4
EIGENBLS	598	224.3	757.8	12	5.2	371.3
MSQRTALS	900	295.3	191.0	25	12.7	275.3
MSQRTBLS	900	435.7	472.7	22	10.9	372.0
WMSQRTALS	900	459.5	267.5	17	9.7	125.7
WMSQRTBLS	900	657.9	518.7	16	9.0	96.8

TABLE 6.3

Numerical results for selected OPM problems: maximum dimension of the generated Krylov subspace max_K , the average dimension ave_K and the overall cpu time in seconds (cpu^*).

space minimization step in place of the commonly employed polynomial Krylov subspaces. In the reported experimental results, our method is able to considerably reduce the overall number of matrix factorizations compared to classic procedures based on the solution of the full space secular equation.

We finally remark that the proposed approach could be easily adapted to trust-region frameworks with second order models; see, e.g. TR2 in [9, Section 3.2]. Our reduced approach could be appealing whenever the solution of the trust-region subproblem beyond the SteihaugToint point is sought; see, e.g., [22, 23]. Indeed, in this case, if the current step lies on the trust-region boundary, the following secular equation

$$\phi_T(\lambda; g_k, H_k, \Delta_k) \stackrel{\text{def}}{=} \|(H_k + \lambda I)^{-1} g_k\| - \Delta_k = 0,$$

has to be (approximately) solved; here $\Delta_k > 0$ is the so-called trust-region radius. Therefore, Algorithm 3.2 can be straightforwardly modified by using ϕ_T in place of ϕ_R in Lines 6 and 16.

Acknowledgments. The work of all the authors was partially supported by INdAM-GNCS under the INdAM-GNCS project CUP_E53C22001930001.

Appendix A. The Frozen AR2 algorithm for second order optimality points. We describe in Algorithms A.1-A.2 the details of a variant of the

Algorithm A.1 The Frozen AR2 algorithm for second order optimality points (FAR2-SO)

Require: An initial point x_0 ; accuracy thresholds $\epsilon, \epsilon_H \in (0, 1)$; an initial regularization parameter $\sigma_0 > 0$ are given, and constants $\eta_1, \eta_2, \gamma_1, \gamma_2, \theta_1, \theta_2, \sigma_{\min}$ s.t.

$$\sigma_{\min} \in (0, \sigma_0], \theta_1 > 0, \theta_2 > 0, 0 < \eta_1 \leq \eta_2 < 1, \\ 0 < \gamma_1 < 1 < \gamma_2, 0 < C_{\text{low}} < C_{\text{up}}$$

An integer j_{\max} .

- 1: Compute $f(x_0), g_0 = \nabla f(x_0), H_0 = \nabla^2 f(x_0)$, and set $k = 0$, **refresh** = 1 and $V_0 = []$.
- 2: **Step 1: Test for termination.** If $\|g_k\| \leq \epsilon$ and $\lambda_{\min}(H_k) \geq -\epsilon_H$, terminate.
- 3: **Step 2: Step computation (reduction of secular equation).**
- 4: Compute the basis W_k and V_{k+1} , the scalar $\hat{\lambda}_k$, and the reduced step \hat{s}_k by Algorithm A.2
- 5: **if refresh** or $W_k \hat{s}_k$ satisfies (2.6) and (3.6)
- 6: set $s_k = W_k \hat{s}_k$, **refresh** = 0, and **goto** Step 3
- 7: **end if**
- 8: **if** $H_k + \hat{\lambda}_k I \succ 0$
- 9: Compute $s_k = -(H_k + \hat{\lambda}_k I)^{-1} g_k$.
- 10: **else** % unsuccessful iteration, step rejection
- 11: Define $x_{k+1} = x_k, \sigma_{k+1} = \sigma_k$, set **refresh** = 1, increment k by one and **goto** Step 1
- 12: **end if**
- 13: **if** $\|s_k\|/\|\hat{s}_k\| < C_{\text{low}}$ or $\|s_k\|/\|\hat{s}_k\| > C_{\text{up}}$ % unsuccessful iteration, step rejection
- 14: Define $x_{k+1} = x_k, \sigma_{k+1} = \sigma_k$, set **refresh** = 1, increment k by one and **goto** Step 1
- 15: **Step 3** Compute

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{T_k(0) - T_k(s_k)}.$$

- 16: **Step 4**
- 17: **if** $\rho_k \geq \eta_1$ **then** % Successful iteration
- 18: set $x_{k+1} = x_k + s_k$, compute $g_{k+1} = \nabla f(x_{k+1}), H_{k+1} = \nabla^2 f(x_{k+1})$
- 19: **else** % unsuccessful iteration, step rejection
- 20: set $x_{k+1} = x_k$
- 21: **end if**
- 22: **Step 5: Regularization parameter update.** Compute

$$\sigma_{k+1} \in \begin{cases} [\max\{\sigma_{\min}, \gamma_1 \sigma_k\}, \sigma_k] & \text{if } \rho_k \geq \eta_2 \\ \sigma_k & \text{if } \rho_k \in [\eta_1, \eta_2) \\ \gamma_2 \sigma_k & \text{otherwise} \end{cases}$$

- 23: Increment k by one and **goto** Step 1.
-

FAR2 method (Algorithms 3.1-3.2) aimed at approximating second order optimality points.

REFERENCES

- [1] A. C. ANTOUNAS, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, 2005.
- [2] S. BELLAVIA, N. KREJI, AND N. KRKLEC JERINKI, *Subsampled inexact Newton methods for*

Algorithm A.2 Refresh and solution of the projected secular equation (second order optimality case)

Require: The matrix H_k ; the vector g_k ; an initial basis V_k ; accuracy thresholds $\theta_1 > 0, \theta_2 > 0$; the parameter σ_k ; an integer $j_{\max} \ll n$; **refresh**.

```

1:   if refresh                                     % Generate new proj space
2:       Set  $V_{k+1} = []$ 
3:       for  $j = 1, \dots, j_{\max} - 1$ 
4:           Set  $W^{(j)} = \text{orth}([V_{k+1}, g_k])$ 
5:           Compute projections  $g^{(j)} = (W^{(j)})^T g_k, \quad H^{(j)} = (W^{(j)})^T H_k W^{(j)}$ 
6:           Find  $\hat{\lambda}$  s.t.  $\phi_R(\hat{\lambda}; g^{(j)}, H^{(j)}, \sigma_k) = 0$ , i.e.

                                   
$$\hat{\lambda} = \sigma_k \|(H^{(j)} + \hat{\lambda}I)^{-1} g^{(j)}\|$$


7:           Set  $\hat{s} = -(H^{(j)} + \hat{\lambda}I)^{-1} g^{(j)}$ 
8:           if  $\|\nabla m_k(W^{(j)} \hat{s})\| \leq \frac{1}{2} \theta_1 \|\hat{s}\|^2$  and

                                   
$$\lambda_{\min}(\nabla^2 m_k(W^{(j)} \hat{s})) \geq -\theta_2 \|\hat{s}\|$$


9:               Set  $\hat{\lambda}_k = \hat{\lambda}, \hat{s}_k = \hat{s}, W_k = W^{(j)}$  and return
10:            end if
11:            Expand  $V_{k+1}$  with new Krylov direction
12:        end for
13:    else                                           % Project onto the old space
14:        Set  $W^{(\hat{j})} = \text{orth}([V_k, g_k])$ 
15:        Compute projections  $g^{(\hat{j})} = (W^{(\hat{j})})^T g_k, \quad H^{(\hat{j})} = (W^{(\hat{j})})^T H_k W^{(\hat{j})}$ 
16:        Find  $\hat{\lambda}$  s.t.  $\phi_R(\hat{\lambda}; g^{(\hat{j})}, H^{(\hat{j})}, \sigma_k) = 0$ , i.e.

                                   
$$\hat{\lambda} = \sigma_k \|(H^{(\hat{j})} + \hat{\lambda}I)^{-1} g^{(\hat{j})}\|$$


17:        Set  $\hat{s}_k = -(H^{(\hat{j})} + \hat{\lambda}I)^{-1} g^{(\hat{j})}, \quad \hat{\lambda}_k = \hat{\lambda}$ 
18:        Set  $V_{k+1} = V_k$  and  $W_k = W^{(\hat{j})}$ 
19:    end if

```

minimizing large sums of convex functions, IMA Journal of Numerical Analysis, 40 (2019), pp. 2309–2341.

- [3] S. BELLAVIA, N. KREJI, B. MORINI, AND S. REBEGOLDI, *A stochastic first-order trust-region method with inexact restoration for finite-sum minimization*, Computational Optimization and Applications, 84 (2023), pp. 53–84.
- [4] E. G. BIRGIN AND J. M. MARTÍNEZ, *The use of quadratic regularization with a cubic descent condition for unconstrained optimization*, SIAM Journal on Optimization, 27 (2017), pp. 1049–1074.
- [5] R. BOLLAPRAGADA, R. H. BYRD, AND J. NOCEDAL, *Exact and inexact subsampled Newton methods for optimization*, IMA Journal of Numerical Analysis, 39 (2018), pp. 545–578.
- [6] Y. CARMON AND J. DUCHI, *Gradient Descent Finds the Cubic-Regularized Nonconvex Newton Step*, SIAM Journal on Optimization, 29 (2019), pp. 2146–2178.
- [7] C. CARTIS, N. GOULD, AND P. TOINT, *Trust-region and other regularisations of linear least-squares problems*, BIT Numerical Mathematics, 49 (2009), pp. 21–53.
- [8] C. CARTIS, N. I. GOULD, AND P. L. TOINT, *Adaptive cubic regularisation methods for unconstrained optimization. part I: motivation, convergence and numerical results*, Mathematical Programming, 127 (2011), pp. 245–295.
- [9] C. CARTIS, N. I. M. GOULD, AND P. L. TOINT, *Evaluation complexity of algorithms for nonconvex optimization*, MOS-SIAM Series on Optimization, 2022.
- [10] C.-C. CHANG AND C.-J. LIN, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology, 2 (2011), pp. 27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [11] ———, *UCI machine learning repository*, 2013. <https://archive.ics.uci.edu/ml/index.php>.
- [12] S. H. CHENG AND N. J. HIGHAM, *A modified Cholesky algorithm based on a symmetric*

- indefinite factorization*, SIAM Journal on Matrix Analysis and Applications, 19 (1998), pp. 1097–1110.
- [13] F. E. CURTIS, D. P. ROBINSON, AND M. SAMADI, *A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization*, Mathematical Programming, 162 (2017), pp. 1–32.
- [14] ———, *An inexact regularized Newton framework with a worst-case iteration complexity of for nonconvex optimization*, IMA Journal of Numerical Analysis, 39 (2019), pp. 1296–1327.
- [15] E. D. DOLAN AND J. J. MORÉ, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91 (2002), pp. 201–213.
- [16] V. DRUSKIN, L. KNIZHNERMAN, AND M. ZASLAVSKY., *Solution of large scale evolutionary problems using rational Krylov subspaces with optimized shifts*, SIAM J. Sci. Comput., 31 (2009), pp. 3760–3780.
- [17] V. DRUSKIN, C. LIEBERMAN, AND M. ZASLAVSKY, *On Adaptive Choice of Shifts in Rational Krylov Subspace Reduction of Evolutionary Problems*, SIAM J. Sci. Comput., 32 (2010), pp. 2485–2496.
- [18] V. DRUSKIN, C. LIEBERMAN, AND M. ZASLAVSKY, *On adaptive choice of shifts in rational Krylov subspace reduction of evolutionary problems*, SIAM J. Sci. Comput., 32 (2010), pp. 2485–2496.
- [19] V. DRUSKIN AND V. SIMONCINI, *Adaptive Rational Krylov Subspaces for Large-scale Dynamical Systems*, Systems Control Lett., 60 (2011), pp. 546–560.
- [20] J.-P. DUSSAULT AND D. ORBAN, *Scalable adaptive cubic regularization methods*, arXiv preprint arXiv: 2103.16659, (2021).
- [21] N. GOULD, D. ROBINSON, AND H. THORNE, *On solving trust-region and other regularised subproblems in optimization*, Mathematical Programming Computation, 2 (2010), pp. 21–57.
- [22] N. I. GOULD, S. LUCIDI, M. ROMA, AND P. L. TOINT, *Solving the trust-region subproblem using the lanczos method*, SIAM Journal on Optimization, 9 (1999), pp. 504–525.
- [23] N. I. GOULD, M. PORCELLI, AND P. L. TOINT, *Updating the regularization parameter in the adaptive cubic regularization algorithm*, Computational Optimization and Applications, 53 (2012), pp. 1–22.
- [24] N. I. GOULD AND V. SIMONCINI, *Error estimates for iterative algorithms for minimizing regularized quadratic subproblems*, Optimization Methods and Software, 35 (2020), pp. 304–328.
- [25] S. GRATTON, S. JERAD, AND P. L. TOINT, *Yet another fast variant of Newton’s method for nonconvex optimization*, arXiv preprint arXiv: 2302.10065, (2023).
- [26] S. GRATTON AND P. L. TOINT, *OPM, a collection of Optimization Problems in Matlab*, arXiv preprint arXiv: 2112.05636, (2021).
- [27] S. GÜTTEL, *Rational Krylov Approximation of Matrix Functions: Numerical Methods and Optimal Pole Selection*, GAMM-Mitteilungen, 36 (2013), pp. 8–31.
- [28] N. J. HIGHAM, *Functions of Matrices*, Society for Industrial and Applied Mathematics, 2008.
- [29] X. JIA, X. LIANG, C. SHEN, AND L.-H. ZHANG, *Solving the cubic regularization model by a nested restarting Lanczos method*, SIAM Journal on Matrix Analysis and Applications, 43 (2022), pp. 812–839.
- [30] Y. LECUN, L. BOTTOU, Y. BENGIO, AND P. HAFFNER, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324. MNIST database available at <http://yann.lecun.com/exdb/mnist>.
- [31] F. LIEDER, *Solving large-scale cubic regularization by a generalized eigenvalue problem*, SIAM Journal on Optimization, 30 (2020), pp. 3345–3358.
- [32] A. RUHE, *The rational Krylov algorithm for nonsymmetric eigenvalue problems. III: Complex shifts for real matrices*, BIT, 34 (1994), pp. 165–176.
- [33] V. SIMONCINI, *Computational Methods for Linear Matrix Equations*, SIAM Review, 58 (2016), pp. 377–441.
- [34] THE MATHWORKS INC., *MATLAB version: 9.7.0 (R2019b)*, 2019.
- [35] D. S. WATKINS, *The Matrix Eigenvalue Problem*, Society for Industrial and Applied Mathematics, 2007.
- [36] Z. YAO, P. XU, F. ROOSTA, AND M. W. MAHONEY, *Inexact Nonconvex Newton-Type Methods*, INFORMS Journal on Optimization, 3 (2021), pp. 154–182.