An optimally fast objective-function-free minimization algorithm using random subspaces

Stefania Bellavia[*], Serge Gratton[†], Benedetta Morini[*], Philippe L. Toint[‡]

19 January 2025

Abstract

An algorithm for unconstrained non-convex optimization is described, which does not evaluate the objective function and in which minimization is carried out, at each iteration, within a randomly selected subspace. It is shown that this random approximation technique does not affect the method's convergence nor its evaluation complexity for the search of an ϵ-approximate first-order critical point, which is $\mathcal{O}(\epsilon^{-(p+1)/p})$, where p is the order of derivatives used. A variant of the algorithm using approximate Hessian matrices is also analysed and shown to require at most $\mathcal{O}(\epsilon^{-2})$ evaluations. Preliminary numerical tests show that the random-subspace technique can significantly improve performance when used with $p = 2$ in the correct context, making it very competitive when compared to standard first-order algorithms.

Keywords: nonlinear optimization, stochastic adaptive regularisation methods, sketching, evaluation complexity, objective-function-free optimization (OFFO).

1 Introduction

Recent years have seen the emergence of random concepts in iterative algorithms for nonconvex optimization (see [13] and reference therein and [5, 1, 2, 3, 27]). In particular, several authors [30, 17, 28, 15, 6, 7][1] have suggested algorithms in which the search for a better iterate is carried out in random subspaces of the space of variables, instead of, as is more traditional and often more costly, in the complete space. In these proposals, the Johnson-Lindenstrauss embedding Lemma (see [14] for a simple exposition) is used to ensure that the relevant information can be found very efficiently in the selected subspace with high probability, and this leads to an elegant analysis yielding optimal complexity bounds for "random-subspace" variants of the standard trust-region and adaptive-regularisation methods for unconstrained minimization. In parallel with this interesting development, alternative non-standard optimization methods have also been introduced where the objective function of the problem is never computed (these algorithms use derivatives'

[1] Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Italy. Member of the INdAM Research Group GNCS. Email: stefania.bellavia@unifi.it, benedetta.morini@unifi.it. Work partially supported by INDAM-GNCS through Progetti di Ricerca 2024 and by PNRR - Missione 4 Istruzione e Ricerca - Componente C2 Investimento 1.1, Fondo per il Programma Nazionale di Ricerca e Progetti di Rilevante Interesse Nazionale (PRIN) funded by the European Commission under the NextGeneration EU programme, project "Numerical Optimization with Adaptive Accuracy and Applications to Machine Learning", code: 2022N3ZNAX MUR D.D. financing decree n. 973 of 30th June 2023 (CUP B53D23012670006), and by Partenariato esteso FAIR "Future Artificial Intelligence Research" SPOKE 1 Human-Centered AI. Obiettivo 4, Project "Mathematical and Physical approaches to innovative Machine Learning technologies (MaPLe)".

[2] Université de Toulouse, INP, IRIT, Toulouse, France. Work partially supported by 3IA Artificial and Natural Intelligence Toulouse Institute (ANITI), French "Investing for the Future - PIA3" program under the Grant agreement ANR-19-PI3A-0004. Email: serge.gratton@enseeiht.fr.

[3] Namur Center for Complex Systems (naXys), University of Namur, 61, rue de Bruxelles, B-5000 Namur, Belgium. Email: philippe.toint@unamur.be.

[1] [7] was posted on arXiv when the revision of the present paper was being finalized. It elaborates on the results of [28, 6] by using a variable dimension sketching strategy.

values only). The motivation for such methods originates in applications with noisy objective functions. Indeed, because differences of objective function's values are not used to accept or reject iterates, the methods' behaviour is much less sensitive to noise than that of the more standard algorithms using function values [23] This new class of "objective-function-free optimization" (OFFO) methods includes popular first-order algorithms as ADAM or ADAGRAD, and has been investigated, for instance, in [16, 25, 29, 19, 33].

The purpose of this paper is to discuss an algorithm which combines these two ideas for the first time while maintaining the desirable properties of both. More specifically, we describe an OFFO adaptive regularisation method using first- or higher-order models defined in random subspaces, and show that this algorithm still enjoys the optimal global rate of convergence known for comparable adaptive-regularisation methods. Independently of the practical interest for such a method, which, we argue below, can be substantial in the right context, our analysis is a new step in the "information thinning" question, which is to isolate what information is necessary for a minimization method to achieve optimal complexity. Indeed, while [20] proves that function values are unnecessary, the present paper further shows that this is also the case for "full space" information[2] under suitable probabilistic assumptions.

Our approach has a further advantage compared to existing proposals, like the random-subspace trust-region and random-subspace regularisation methods of [28] and [6]. Because no evaluation of the objective function is involved, the algorithm generates a much simpler random process (there is now only one random event per iteration), in turn considerably simplifying the proofs as the number of iteration types whose number must be estimated (in [28, Chapter 4]) is now reduced to only two. While our theory covers the general case where derivatives of higher order than one are estimated, our practical focus will be on the case where first and second derivatives are used.

The paper is organised as follows. The new algorithm is proposed in Section 2, while its evaluation complexity is analysed under general embedding conditions in Section 3. A brief discussion of a possible way to select the random subspaces are presented in Section 4. The numerical behaviour of the second-order variant is illustrated in Section 5. Some conclusions are finally presented in Section 6. A discussion of a further variant using quadratically regularised inexact quadratic models is proposed and analysed in appendix.

2 An OFFO adaptive regularisation algorithm using random subspaces

The problem of interest in what follows is the standard nonconvex unconstrained minimization of a (sufficiently) smooth objective function, that is

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

where $f:\mathbb{R}^n\to\mathbb{R}$. As indicated in the introduction, our aim is to design an adaptive regularisation algorithm in which *the objective function value is never computed*, and in which the step is obtained by approximately minimizing a suitable model of the objective function in a random subspace. To ensure that this approach is sensible, we make the following assumptions.

AS.1 f is p times continuously differentiable in $\mathbb{R}^n$.
AS.2 There exists a constant $f_{\rm low}$ such that $f(x)\geq f_{\rm low}$ for all $x\in\mathbb{R}^n$.
AS.3 The pth derivative of f is globally Lipschitz continuous, that is, there exists a non-negative constant L_p such that

$$\|\nabla_x^p f(x)-\nabla_x^p f(y)\|\leq L_p\|x-y\| \;\text{ for all } x,y\in\mathbb{R}^n,$$

[2] One might argue that it has long been known that information along the directions given by the gradient and the step suffices, but this requires the step to be known and thus amounts to an *a posteriori* observation instead of an *a priori* algorithmically exploitable strategy.

where $\|.\|$ denotes the Euclidean norm for vectors in $\mathbb{R}^n$ and the corresponding subordinate norm for tensors.
AS.4 The gradient of f is bounded, that is there exists a constant $\kappa_{\rm g} \geq 0$ such that, for all $x \in \mathbb{R}^n$,

$$\|\nabla_x^1 f(x)\| \leq \kappa_{\rm g}.$$

AS.5 If $p > 1$, there exists a constant $\kappa_{\rm high} \geq 0$ such that

$$\min_{\|d\|\leq 1} \nabla_x^i f(x)[d]^i \geq -\kappa_{\rm high} \;\text{ for all }\; x \in \mathbb{R}^n \;\text{ and }\; i \in \{2,\ldots,p\},$$

where $\nabla_x^i f(x)$ is the ith derivative tensor of f computed at x, and where $T[d]^i$ denotes the i-dimensional tensor T applied on i copies of the vector d. (For notational convenience, we set $\kappa_{\rm high} = 0$ if $p = 1$.)

We refer the reader to [11, Appendix 6] for details on derivative tensors. Observe that, given AS.1, AS.3 is automatically satisfied if the iterates generated by the algorithms remain in a bounded domain. This is in particular the case if a objective function's level set identified in Lemma 3.7 is bounded because, as we comment on after this lemma, it contains all generated iterates. Observe also that AS.5 is irrelevant in the case where $p = 1$. Should one be interested in higher-order methods, AS.5 is weaker than assuming uniform boundedness of the derivative tensors of degree two and above (there is no upper bound on the value of $\nabla_x^i f(x)[d]^i$), or, equivalently, Lipschitz continuity of derivatives of degree one to $p-1$.

2.1 The SKOFFARp algorithm

As suggested above, adaptive regularisation methods are iterative schemes which compute a step from an iterate x_k to the next by approximately minimizing a p-th degree regularised model $m_k(s)$ of $f(x_k+s)$ of the form

$$m_k(s) \stackrel{\rm def}{=} T_{f,p}(x_k,s) + \frac{\sigma_k}{(p+1)!}\|s\|^{p+1}, \tag{2}$$

where $T_{f,p}(x,s)$ is the pth order Taylor expansion of functional f at x truncated at order p, that is,

$$T_{f,p}(x,s) \stackrel{\rm def}{=} f(x) + \sum_{i=1}^p \frac{1}{i!}\nabla_x^i f(x)[s]^i. \tag{3}$$

To obtain the model (2), the p-th order Taylor series (3) is "regularised" by adding the term $\frac{\sigma_k}{(p+1)!}\|s\|^{p+1}$ (where σ_k is the iteration-dependent regularisation parameter), thereby ensuring that $m_k(s)$ is bounded below and that a step s_k (approximately) minimizing this model is well-defined.

Following [28], we propose to compute a random subspace step at iteration k as follows. Given an iteration-independent distribution $\mathcal{S}$ of $\ell \times n$ random matrices (with $\ell < n$), let S_k be drawn from this distribution and consider minimizing the sketched regularised model

$$\widehat{m}_k(\widehat{s}) \stackrel{\rm def}{=} \widehat{T}_{f,p}(x_k,\widehat{s}) + \frac{\sigma_k}{(p+1)!}\|S_k^T\widehat{s}\|^{p+1}, \tag{4}$$

as a function of $\widehat{s} \in \mathbb{R}^\ell$, where the sketched Taylor model $\widehat{T}_{f,p}(x,\widehat{s})$ is given by

$$\widehat{T}_{f,p}(x,\widehat{s}) \stackrel{\rm def}{=} f(x) + \sum_{i=1}^p \frac{1}{i!}\nabla_x^i f(x)[S_k^T\widehat{s}]^i.$$

Letting $\widehat{s}_k$ an approximate minimizer of $\widehat{m}_k(\widehat{s})$, the full dimensional step is then defined by $s_k = S_k^T\widehat{s}_k$. We note that $\widehat{T}_{f,p}(x_k,\widehat{s}_k) = T_{f,p}(x_k,s_k)$ and

$$\widehat{m}_k(\widehat{s}_k) = m_k(s_k). \tag{5}$$

Algorithm 2.1: Sketching OFFO adaptive regularisation of degree p (SKOFFARp)

Step 0: Initialization: An initial point $x_0 \in \mathbb{R}^n$, a regularisation parameter $\nu_0 > 0$ and a requested final gradient accuracy $\epsilon \in (0,1]$ are given, as well as the parameters

$$\theta > 1, \; \mu_{-1} \geq 0 \quad \text{and} \quad 0 < \vartheta < 1.$$

Set $k = 0$.

Step 1: Step calculation: If $k = 0$, set $\sigma_0 = \nu_0$. Otherwise, select

$$\sigma_k \in \Big[\vartheta\nu_k, \max[\nu_k, \mu_k]\Big], \tag{6}$$

where

$$\mu_k = \max\left[\mu_{k-1}, \frac{\|S_{k-1}g_k\| - \|\nabla^1_{\widehat{s}}\widehat{T}_{f,p}(x_{k-1}, \widehat{s}_{k-1})\|}{\kappa_{S,k-1} . \|s_{k-1}\|^p}\right], \tag{7}$$

with some $\kappa_{S,k-1}$ such that $\|S_{k-1}\| \leq \kappa_{S,k-1}$. Draw a random matrix $S_k \in \mathbb{R}^{\ell \times n}$ from $\mathcal{S}$ and compute a step $s_k = S_k^T\widehat{s}_k$ such that $\widehat{s}_k$ sufficiently reduces the model $\widehat{m}_k$ defined in (4) in the sense that

$$\widehat{m}_k(\widehat{s}_k) - \widehat{m}_k(0) < 0 \tag{8}$$

and

$$\|\nabla^1_{\widehat{s}}\widehat{T}_{f,p}(x_k, \widehat{s}_k)\| \leq \theta\frac{\sigma_k}{p!}\|S_k^T\widehat{s}_k\|^{p-1}\|S_kS_k^T\widehat{s}_k\|. \tag{9}$$

Step 2: Updates. Set

$$x_{k+1} = x_k + s_k$$

and

$$\nu_{k+1} = \nu_k + \nu_k\|s_k\|^{p+1}. \tag{10}$$

Increment k by one and go to Step 1.

Some comments on this algorithm are necessary.

1. It is crucial to observe that, while the definition of the model in (4) involves the function value $f(x_k)$ (in $\widehat{T}_{f,p}(x_k,\widehat{s})$), this function value is never needed in the algorithm (it cancels out in (8)) and therefore need not to be evaluated. The algorithm thus belong to the OFFO class. Of course, the minimization of the model may require the evaluation of the sketched derivatives $\{\nabla^j_{\widehat{s}} f(x_k)[S_k\cdot]^j\}_{j=1}^p$, at least along some directions[3]. This makes the use of derivatives of degree higher than two potentially useable in practice, especially if the objective function is partially separable [24, 12].

2. Since
$$\nabla^1_{\widehat{s}}\|S_k^T\widehat{s}\|^{p+1} = (p+1)\|S_k^T\widehat{s}\|^{p-1}\, S_kS_k^T\widehat{s},$$
one verifies that conditions (8) and (9) do hold at an exact minimizer of $\widehat{m}_k$ (the latter with $\theta = 1$). A step satisfying these conditions is therefore guaranteed to exist. Note that (9) is a condition on the norm of the gradient of the Taylor series for f, at variance with [11, 28] where the condition is on the gradient of the regularised model (2).

3. At variance with standard trust-region and adaptive-regularisation methods, the algorithm does not involve any (typically noise sensitive) test to accept or reject the trial iterate x_k+s_k, and every trial point is thus "accepted" as the new iterate. In the vocabulary used for trust-region and adaptive regularisation methods, every iteration is therefore "successful".

4. The value of μ_k in the definition (6) of σ_k is chosen to help the regularisation parameter σ_k to grow fast enough, given the knowledge at iteration k. We will show in Lemma 3.5 that μ_k is bounded above by $\max[\mu_{-1}, L_p]$ irrespective of the choice of $\kappa_{S,k-1}$. As a consequence, the specific values of $\kappa_{S,k-1}$ in (7) play no role in our complexity analysis, albeit they obviously affect the practical performance of the method. Finally, we stress that the knowledge of the constants L_p and κ_g, given in AS.3 and AS.4 respectively, is *not required* in the algorithm.

The SKOFFARp algorithm can be seen as a stochastic process because the selection of S_k is random and yields random realizations[4] of the iterates x_k and of the steps s_k. The objective of our forthcoming complexity analysis for this algorithm is to derive a probabilistic bound on the process hitting time
$$N_1(\epsilon) \stackrel{\rm def}{=} \min\{k \in \mathbb{N} \mid \|g_k\| \le \epsilon\}, \tag{11}$$
where we denote $g_k \stackrel{\rm def}{=} \nabla^1_x f(x_k)$ for all k. $N_1(\epsilon)$ is the number of iterations that a particular realization of the algorithm requires to obtain an ϵ-approximate first-order critical point.

3 Evaluation complexity for the SKOFFARp algorithm

Before discussing our analysis of evaluation complexity, we first restate some classical lemmas for ARp algorithms, starting with Lipschitz error bounds.

Lemma 3.1 Suppose that AS.1 and AS.3 hold. Then
$$f(x_{k+1}) - \widehat{T}_{f,p}(x_k,\widehat{s}_k) = f(x_{k+1}) - T_{f,p}(x_k,s_k) \le \frac{L_p}{(p+1)!}\|s_k\|^{p+1}, \tag{12}$$
and
$$\|g_{k+1} - \nabla^1_s T_{f,p}(x_k,s_k)\| \le \frac{L_p}{p!}\|s_k\|^p. \tag{13}$$

[3] In the course of a Krylov subproblem solver for $p = 2$, say.

[4] Formally, the iterates and steps are random variables on some implicitly defined probability space, and x_k and s_k are their realizations.

Proof. This is a standard result (see [10, Lemma 2.1] for instance). □

We next state a simple lower bound on the Taylor series' decrease.

Lemma 3.2

$$\Delta T_{f,p}(x_k, s_k) \stackrel{\rm def}{=} T_{f,p}(x_k, 0) - T_{f,p}(x_k, s_k) > \frac{\sigma_k}{(p+1)!}\|s_k\|^{p+1}. \tag{14}$$

Proof. The bound directly results from $\widehat{m}_k(\widehat{s}_k) = m_k(s_k)$, (8) and (2). □

This and AS.2 allow us to establish a lower bound on the decrease in the objective function (although it is never computed).

Lemma 3.3 Suppose that AS.1 and AS.3 hold and that $\sigma_k \geq 2L_p$. Then

$$f(x_k) - f(x_{k+1}) > \frac{\sigma_k}{2(p+1)!}\|s_k\|^{p+1}. \tag{15}$$

Proof. From (12) and (14), we obtain that

$$f(x_k) - f(x_{k+1}) > \frac{\sigma_k - L_p}{(p+1)!}\|s_k\|^{p+1}$$

and (15) immediately follows from our assumption on σ_k. □

We now recall an upper bound on $\|s_k\|$ generalizing those proposed in [8, 22] to the case where p is arbitrary.

Lemma 3.4 Suppose that AS.1 and AS.5 hold. At each iteration k, we have that

$$\|s_k\| \leq 2\eta + 2\left(\frac{(p+1)!\|g_k\|}{\sigma_k}\right)^{\frac{1}{p}}, \tag{16}$$

where

$$\eta = \sum_{i=2}^{p}\left[\frac{\kappa_{\rm high}(p+1)!}{i!\,\vartheta\nu_0}\right]^{\frac{1}{p-i+1}}. \tag{17}$$

Proof. See [20, Lemma 3.6]. Note that this result does not involve S_k as it is valid for any step which reduces m_k and, using (5) and (8), $m_k(s_k) = \widehat{m}_k(\widehat{s}_k) < \widehat{m}_k(0) = m_k(0)$. □

Our next step is to show that μ_k is bounded.

Lemma 3.5 Suppose that AS.1 and AS.3 hold. For all $k \geq 0$,

$$\mu_k \leq \max[\mu_{-1}, L_p]. \tag{18}$$

Proof. We have that $\nabla^1_{\widehat{s}}\widehat{T}_{f,p}(x_{k-1},\widehat{s}_{k-1}) = \nabla^1_{\widehat{s}}T_{f,p}(x_{k-1},S_{k-1}^T s_{k-1}) = S_{k-1}\nabla^1_s T_{f,p}(x_{k-1},s_{k-1})$, so that, using the triangular inequality, (13) and (9),

$$\begin{array}{rcl} \|S_{k-1}g_k\| & \leq & \|S_{k-1}(g_k - \nabla^1_x T_{f,p}(x_{k-1},s_{k-1}))\| + \|S_{k-1}\nabla^1_x T_{f,p}(x_{k-1},s_{k-1})\| \\ & \leq & \|S_{k-1}\| L_p \|s_{k-1}\|^p + \|\nabla^1_{\widehat{s}}\widehat{T}_{f,p}(x_{k-1},\widehat{s}_{k-1})\|, \end{array}$$

and thus

$$L_p \geq \frac{\|S_{k-1}g_k\| - \|\nabla^1_{\widehat{s}}\widehat{T}_{f,p}(x_{k-1},\widehat{s}_{k-1})\|}{\|S_{k-1}\|\|s_{k-1}\|^p}. \tag{19}$$

The inequality (18) then follows from (7) and $\|S_{k-1}\| \leq \kappa_{S,k-1}$. □

The proof of this lemma shows that a tighter lower bound on L_p (see (19)) is also available at the often significant cost of evaluating $\|S_{k-1}\|$, thus motivating the introduction of the (hopefully) cheaper $\kappa_{S,k-1}$.

Since our objective is to minimize f, obtaining a decrease as stated by Lemma 3.3 is important. The condition $\sigma_k \geq 2L_p$ in this lemma and (6) together suggest that the condition

$$\nu_k \geq \frac{2L_p}{\vartheta} \tag{20}$$

is important for our subsequent analysis. Remembering that ν_k is increasing with k, we therefore define

$$k_1 \stackrel{\rm def}{=} \inf\left\{k \geq 1 \mid \nu_k \geq \frac{2L_p}{\vartheta}\right\} \tag{21}$$

the index of the first iterate (in a given realization) such that significant objective function decrease is guaranteed by Lemma 3.3. Note that k_1 may be infinite, which is why we define the random event

$$\mathcal{K}_1 \stackrel{\rm def}{=} \{k_1 \text{ as defined by (21) is finite}\}. \tag{22}$$

We now pursue our analysis under the condition that $\mathcal{K}_1$ occurs. The next series of lemmas provides bounds, conditional on $\mathcal{K}_1$, on $f(x_{k_1})$ and ν_{k_1}, which in turn allows establishing an upper bound on the regularisation parameter, only depending on the problem and the fixed algorithmic parameters.

Lemma 3.6 Suppose that AS.1, AS.3, AS.4 and AS.5 hold and consider a realization of the SKOFFARp algorithm where $\mathcal{K}_1$ occurs. Then

$$\nu_{k_1} \leq \nu_{\max} \stackrel{\rm def}{=} \frac{2L_p}{\vartheta}\left[1 + \left(2\eta + 2\left(\frac{(p+1)!\kappa_{\rm g}}{\vartheta\nu_0}\right)^{\frac{1}{p}}\right)^{p+1}\right], \tag{23}$$

where η is defined in (17) and $\kappa_{\rm g}$ in AS.4.

Proof. Since $\mathcal{K}_1$ is assumed to occur, k_1 is well-defined and finite. Successively using Lemma 3.4 and the update rule for ν_k (10), we derive that

$$\nu_{k_1} \stackrel{(10)}{=} \nu_{k_1-1} + \nu_{k_1-1}\|s_{k_1-1}\|^{p+1} \stackrel{(16)}{\leq} \nu_{k_1-1} + \nu_{k_1-1}\left(2\left((p+1)!\frac{\|g_{k_1-1}\|}{\sigma_{k_1-1}}\right)^{\frac{1}{p}} + 2\eta\right)^{p+1}$$

and the desired result follows by using AS.4, the definition of k_1 in (21) and the inequalities $\sigma_{k_1-1} \geq \vartheta\nu_{k_1-1} \geq \vartheta\nu_0$. □

Lemma 3.6 allows us to establish an upper bound on $f(x_{k_1})$ as a function of $\nu_{\max}$.

Lemma 3.7 Suppose that AS.1, AS.3, AS.4 and AS.5 hold and consider a realization of the SKOFFARp algorithm where $\mathcal{K}_1$ occurs. Then

$$f(x_{k_1}) \le f_{\max} \stackrel{\rm def}{=} f(x_0) + \frac{1}{(p+1)!}\left(\frac{L_p}{\sigma_0}\nu_{\max} + \vartheta\sigma_0\right). \tag{24}$$

Proof. Lemma 3.8 in [20] shows that, for any $k \ge 0$

$$f(x_k) \le f(x_0) + \frac{1}{(p+1)!}\left(\frac{L_p\nu_k}{\sigma_0} + \vartheta\sigma_0)\right).$$

The desired bound then follows from Lemma 3.6. □

Observe that this result ensures that all iterates generated by the algorithm belong to the level set $\{x \in \mathbb{R}^n \mid f(x) \le f(x_{k_1})\}$. The two bounds stated in Lemmas 3.7 and 3.6 are also useful in that they now imply an upper bound on the regularisation parameter, an important step in standard theory for regularisation methods.

Lemma 3.8 Suppose that AS.1, AS.2, AS.3, AS.4 and AS.5 hold and consider a realization of the SKOFFARp algorithm. Then

$$\begin{aligned}\sigma_k &\le \sigma_{\max}\\ &\stackrel{\rm def}{=} \max\left[\frac{2(p+1)!}{\vartheta}\left[f(x_0) - f_{\rm low} + \frac{1}{(p+1)!}\left(\frac{L_p}{\sigma_0}\nu_{\max} + \vartheta\sigma_0\right)\right] + \nu_{\max}, \mu_{-1}, \frac{2L_p}{\vartheta}, \nu_0\right].\end{aligned} \tag{25}$$

Proof. We proceed as in Lemma [20, Lemma 3.9] and give the proof for sake of clarity. Suppose first that $\mathcal{K}_1$ occurs. From the definition of k_1 in (21), we deduce that $\sigma_j \ge 2L_p$. From Lemma 3.3, we then have that

$$f(x_j) - f(x_{j+1}) \ge \frac{\sigma_j}{2(p+1)!}\|s_j\|^{p+1} \ge \vartheta\frac{\nu_j}{2(p+1)!}\|s_j\|^{p+1}.$$

Summing the previous inequality from $j = k_1$ to $k-1$ and using the ν_j update rule (10) and AS.2, we deduce that

$$f(x_{k_1}) - f_{\rm low} \ge f(x_{k_1}) - f(x_k) \ge \frac{\vartheta}{2(p+1)!}(\nu_k - \nu_{k_1}).$$

Rearranging the previous inequality and using Lemma 3.6 then gives that

$$\nu_k \le \frac{2(p+1)!}{\vartheta}\left(f(x_{k_1}) - f_{\rm low}\right) + \nu_{\max}. \tag{26}$$

Combining now Lemma 3.7 (to bound $f(x_{k_1})$), (6) and (18) yields that

$$\sigma_k \le \sigma_{\max} \stackrel{\rm def}{=} \max\left[\frac{2(p+1)!}{\vartheta}\left[f(x_0) - f_{\rm low} + \frac{1}{(p+1)!}\left(\frac{L_p}{\sigma_0}\nu_{\max} + \vartheta\sigma_0\right)\right] + \nu_{\max}, \mu_{-1}, L_p, \nu_0\right].$$

If $\mathcal{K}_1$ does not occur, $\nu_k \le 2L_p/\vartheta$ for all k. Thus we obtain, using (6) and (18), that $\sigma_k \le \max[\frac{2L_p}{\theta}, \mu_{-1}]$ for all k, and (25) also holds. □

The theory of adaptive-regularisation methods crucially depends on the relation between the steplength $\|s_k\|$ and the norm of the gradient at the next iteration $\|g_{k+1}\|$ (see Lemmas 3.3.3 and 4.1.3 in [11], for instance), which is itself bounded below by ϵ before convergence. Here we choose to consider this dependence as a random event, depending on the choice of S_k. This is formalized in the following definition.

Definition 3.9 *Given some $\epsilon \in (0,1)$ independent of k, iteration $k \in \{0,\ldots,N_1(\epsilon)-2\}$ is said to be ω-true for some $\omega \in (0,1)$ independent of k whenever*

$$\|s_k\|^p \geq \omega\epsilon. \tag{27}$$

We discuss in Section 4 conditions which may enforce this property, but immediately note that it automatically holds if S_k is of rank n [20, Lemma 3.4]. We also define

$$\mathcal{T}_k^{(\omega)} \stackrel{\rm def}{=} \{j \in \{0,\ldots,k-1\} \mid \text{iteration } j \text{ is } \omega\text{-true}\}, \tag{28}$$

the index set $\mathcal{T}_k^{(\omega)}$ of all ω-true iterations in the first k.

Given these definitions, we now need to establish under which condition the event $\mathcal{K}_1$ occurs with high probability. Such a condition is obtained in two stages, the first follows arguments by [19, Lemma 7] and [20, Lemma 3.5] and investigates, in our probabilistic setting, the effect of accumulating ω-true iterations.

Lemma 3.10 Suppose that AS.1 and AS.3 hold and consider a particular realization of the SKOFFARp algorithm. Let $k_0 < N_1(\epsilon)$ be an iteration index (in this realization) such that k_* ω-true iterations have been performed among those of index 0 to k_0-1, where

$$k_* \stackrel{\rm def}{=} \left\lceil \frac{2L_p\epsilon^{-\frac{p+1}{p}}}{\vartheta\nu_0\,\omega^{\frac{p+1}{p}}} \right\rceil. \tag{29}$$

Then k_1 exists, $k_1 \leq k_0$ and, for all $k \geq k_1$,

$$\sigma_k \geq 2L_p. \tag{30}$$

Proof. First observe that (30) is a direct consequence of (6) if $\nu_k \geq 2L_p/\vartheta$. Suppose now that, for some $k \in \{k_0,\ldots,N_1(\epsilon)-1\}$, $\nu_k < 2L_p/\vartheta$. Since $\{\nu_k\}$ is a non-decreasing sequence, we deduce that this inequality holds for $j \in \{0,\ldots,k\}$. Successively using the form of the ν_k update rule (10), (27), (6) and the fact that $k < N_1(\epsilon)$, we obtain that

$$\begin{aligned}\nu_k &\overset{(10)}{>} \sum_{j=0}^{k-1}\nu_j\|s_j\|^{p+1} \overset{(28)}{>} \sum_{j\in\mathcal{T}_k^{(\omega)}}\nu_j\|s_j\|^{p+1} \overset{(27)}{\geq} \sum_{j\in\mathcal{T}_k^{(\omega)}}\nu_j(\omega\epsilon)^{\frac{p+1}{p}}\\ &\overset{(6)}{\geq} \sum_{j\in\mathcal{T}_k^{(\omega)}}\nu_0\,(\omega\epsilon)^{\frac{p+1}{p}} \overset{(28)}{\geq} k_*\,\nu_0(\omega\epsilon)^{\frac{p+1}{p}}.\end{aligned}$$

Substituting the definition of k_* in the last inequality, we obtain that

$$\frac{2L_p}{\vartheta} < \nu_k < \frac{2L_p}{\vartheta},$$

which is impossible. Hence no index $k \in \{k_0,\ldots,N_1(\epsilon)-1\}$ exists such that $\nu_k < 2L_p/\vartheta$. Thus, $k_1 \leq k_0$ exists by definition of k_1 in (21). By the same definition, we finally deduce that $\nu_k \geq 2L_p/\vartheta$ for all $k \geq k_1$, in turn implying (30) because of (6). □

Observe that (29) depends on the ratio L_p/ν_0 which is the fraction by which ν_0 underestimates the Lipschitz constant. This lemma thus implies that the probability of $\mathcal{K}_1$ is at least the probability that k_* ω-true iterations are performed, which we now investigate under the following assumption.

AS.6 There exists an $\omega \in (0,1)$ and a $\pi_S^{(1)} > 0$ such that for S_k drawn randomly,

$$\mathbb{P}\Big[\text{iteration } k \text{ is } \omega\text{-true} \mid x_k = \bar{x}_k,\, \sigma_k = \bar{\sigma}_k\Big] \geq \pi_S^{(1)},$$

for any $\bar{x}_k \in \mathbb{R}^n$, any $\bar{\sigma}_k \in [\vartheta\nu_0, \sigma_{\max}]$ and any $k \in \{0, \ldots, N_1(\epsilon) - 2\}$, where $\mathbb{P}[X]$ denotes the probability of the event X. Moreover, the occurrence of k-th iteration being ω-true is conditionally independent of the occurrence of iterations $0, \ldots, k-1$ being ω-true given $x_k = \bar{x}_k$ and $\sigma_k = \bar{\sigma}_k$.

This assumption differs from Assumption 1 in [28, page 71] in that it now it makes the probability of an ω-true iteration conditional not only on x_k but also on σ_k, which we feel is reasonable given the isotropic nature of the regularisation term in (2). Note that a suitable value for ω may depend on the bounds on σ_k (as we will see below in Lemmas 4.1, 4.2 and A.4). Assumption AS.6 can be ensured by suitably using Johnson-Lindenstrauss embeddings [14] and results are available in [28] for $p \in \{1, 2\}$. We will analyse such cases in Section 4.

Before using AS.6 and $\pi_S^{(1)}$ directly, we first recall a known probabilistic result.

Lemma 3.11 For all nonnegative i, let $\mathcal{A}_i$ be an event which can be true or false and is conditionally independent of $\mathcal{A}_0, \mathcal{A}_1, \ldots \mathcal{A}_{i-1}$. For any $\bar{x}_i \in \mathbb{R}^n$ and $\bar{\sigma}_i \in [\vartheta\nu_0, \sigma_{\max}]$, suppose that $\mathbb{P}\Big[\mathcal{A}_i \text{ is true } \mid x_i = \bar{x}_i, \sigma_i = \bar{\sigma}_i\Big] \geq \pi$, with $\pi \in (0,1)$. For $k \geq 0$, let $\mathcal{W}_k = \{i \in \{0, \ldots, k-1\} \mid \mathcal{A}_i \text{ is true}\}$. Then, for any given $\delta_1 \in (0,1)$,

$$\mathbb{P}\Big[|\mathcal{W}_k| > (1-\delta_1)\pi k\Big] \geq 1 - e^{-\frac{\delta_1^2}{2}\pi k}. \tag{31}$$

Proof. See [28, Lemma 4.3.1] where, as mentioned above, we now consider the "state" of the algorithm at iteration i to comprise both x_i and σ_i. □

We are now in position to use this result to obtain a lower bound on the probability that k_* ω-true iterations are performed, and that k_1 is well-defined.

Lemma 3.12 Suppose that AS.1, AS.3 and AS.6 hold and let $\delta_1 \in (0,1)$ be given. Let

$$k_\diamond \stackrel{\text{def}}{=} \left\lceil \frac{k_*}{(1-\delta_1)\pi_S^{(1)}} \right\rceil, \tag{32}$$

where k_* is given by (29). Then

$$\mathbb{P}\Big[\mathcal{K}_1 \mid N_1(\epsilon) > k_\diamond\Big] \geq 1 - e^{-\frac{\delta_1^2}{2}\pi_S^{(1)} k_\diamond} \stackrel{\text{def}}{=} \pi_1^{(1)}. \tag{33}$$

Proof. Identifying $\mathcal{A}_i = \{\text{iteration } i \text{ is } \omega\text{-true}\}$, Lemma 3.11 with $\pi = \pi_S^{(1)}$ and $k_0 = k_\diamond$ gives that the probability that at least k_* ω-true iterations have been performed during iterations 0 to $k_\diamond - 1$ is at least $\pi_1^{(1)}$. The desired conclusion then follows from Lemma 3.10. □

We finally propose a variant of the well-known "telescoping sum" argument adapted to our probabilistic setting to derive the desired evaluation complexity bound.

Theorem 3.13 Suppose that AS.1, AS.2, AS.3, AS.4, AS.5 and AS.6 hold, that $\delta_1 \in (0,1)$ is given and that the SKOFFARp algorithm is applied to problem (1). Define

$$\kappa_{\rm SKOFFARp} \stackrel{\rm def}{=} \frac{4\,[L_p + (p+1)!(f_{\max} - f_{\rm low})]}{\vartheta \nu_0 \omega^{\frac{p+1}{p}}(1-\delta_1)\pi_S^{(1)}} \tag{34}$$

where $f_{\max}$ is defined in (24). Then

$$\mathbb{P}\Big[N_1(\epsilon) \leq \kappa_{\rm SKOFFARp}\,\epsilon^{-\frac{p+1}{p}} + 4\Big] \geq \left(1 - e^{-\frac{\delta_1^2}{2}\pi_S^{(1)}k_\diamond}\right)^2 \tag{35}$$

where $k_\diamond$ is defined by (32).

Proof. First note that (29) and (32) imply that

$$k_\diamond \leq \frac{1}{(1-\delta_1)\pi_S^{(1)}}\left(\frac{2L_p}{\vartheta\nu_0\omega^{\frac{p+1}{p}}}\right)\epsilon^{-\frac{p+1}{p}} + 1. \tag{36}$$

Thus, given (34),

$$\mathbb{P}\Big[N_1(\epsilon) \leq \kappa_{\rm SKOFFARp}\,\epsilon^{-\frac{p+1}{p}} + 4 \;\mid\; N_1(\epsilon) \leq 2k_\diamond + 2\Big] = 1. \tag{37}$$

Suppose now that $N_1(\epsilon) > k_\diamond + 2 > k_\diamond$ and that $\mathcal{K}_1$ occurs. Consider an iteration $j > k_\diamond \geq k_1$ (note that k_1 is well-defined) such that $j+1 < N_1(\epsilon)$ and suppose furthermore that iteration j is ω-true, a situation which occurs with probability at least $\pi_S^{(1)}$ because of AS.6. From the fact that $\mathcal{K}_1$ occurs, $N_1(\epsilon) > k_\diamond$ and the definition of k_1 in (21), we have that $\sigma_j \geq 2L_p$ and we may apply Lemma 3.3, yielding (15) for iteration j. Since this iteration is also ω-true, (15) and inequality (27) also hold for iteration j. Moreover, the fact $\mathcal{K}_1$ occurs ensures (because of Lemma 3.8, (6), the non-decreasing nature of ν_k and the identity $\sigma_0 = \nu_0$) that $\sigma_j \in [\vartheta\sigma_0, \sigma_{\max}]$. Finally, $\|g_{j+1}\| \geq \epsilon$ because $j+1 < N_1(\epsilon)$. Combining these observations, we obtain that

$$f(x_j) - f(x_{j+1}) \geq \frac{\sigma_j\|s_j\|^{p+1}}{2(p+1)!} \geq \frac{\sigma_j\omega^{\frac{p+1}{p}}\|g_{j+1}\|^{\frac{p+1}{p}}}{2(p+1)!} \geq \frac{\vartheta\nu_0\omega^{\frac{p+1}{p}}\epsilon^{\frac{p+1}{p}}}{2(p+1)!} \tag{38}$$

with probability (conditional to $\mathcal{K}_1$ and $N_1(\epsilon) > k_\diamond$+2) at least $\pi_S^{(1)}$. Applying now Lemma 3.11 to iterations of index $k_\diamond + 1$ to j with

$$\mathcal{A}_{i-k_\diamond} = \{\text{ (38) holds at iteration } i - k_\diamond\ \}, \quad \pi = \pi_S^{(1)} \text{ and } k = j - k_\diamond,$$

we deduce that, for all $j \in \{k_\diamond + 1, \ldots, N_1(\epsilon) - 2\}$,

$$\mathbb{P}\left[|\mathcal{V}_j| \geq (j - k_\diamond)(1-\delta_1)\pi_S^{(1)} \,|\, \mathcal{K}_1 \text{ and } N_1(\epsilon) > k_\diamond\right] \geq 1 - e^{-\frac{\delta_1^2}{2}\pi_S^{(1)}(j-k_\diamond)}$$

where $\mathcal{V}_j \stackrel{\rm def}{=} \{i \in \{k_\diamond + 1, \ldots, j\} \mid \text{(38) holds at iteration } i\}$. In particular, we have that

$$\mathbb{P}\left[|\mathcal{V}_j| \geq (j - k_\diamond)(1-\delta_1)\pi_S^{(1)} \,|\, \mathcal{K}_1 \text{ and } N_1(\epsilon) > 2k_\diamond + 2\right] \geq \pi_1^{(1)}, \tag{39}$$

with $\pi_1^{(1)}$ defined in (33), for all $j \in \{2k_\diamond + 1, \ldots, N_1(\epsilon) - 2\}$. We also know from Lemma 3.3 and the definition of k_1 in (21) that the sequence $\{f(x_j)\}$ is non-increasing for $j \geq k_1$, and thus that

$$f(x_{k_1}) - f(x_{j+1}) = \sum_{i=k_1}^{j}[f(x_i) - f(x_{i+1})] \geq \sum_{i=k_\diamond+1}^{j}[f(x_i) - f(x_{i+1})] \geq |\mathcal{V}_j| \min_{i\in\mathcal{V}_j}[f(x_i) - f(x_{i+1})].$$

Combining this inequality with (38) and (39) then yields that

$$\mathbb{P}\left[f(x_{k_1}) - f(x_{j+1}) \geq (j-k_\diamond)(1-\delta_1)\pi_S^{(1)}\ \kappa_2^{-1}\epsilon^{\frac{p+1}{p}}\ \mid \mathcal{K}_1 \text{ and } N_1(\epsilon) > 2k_\diamond + 2\right] \geq \pi_1^{(1)}$$

where

$$\kappa_2 = \frac{2(p+1)!}{\vartheta\nu_0\omega^{\frac{p+1}{p}}}, \tag{40}$$

and thus, because of AS.3, that

$$\mathbb{P}\left[f(x_{k_1}) - f_{\rm low} \geq \kappa_2^{-1}(1-\delta_1)\pi_S^{(1)}(j-k_\diamond)\,\epsilon^{\frac{p+1}{p}}\ \mid \mathcal{K}_1 \text{ and } N_1(\epsilon) > 2k_\diamond + 2\right] \geq \pi_1^{(1)}.$$

Furthermore, (24) in Lemma 3.7 then implies that

$$\mathbb{P}\left[j - k_\diamond \leq \frac{\kappa_2}{(1-\delta_1)\pi_S^{(1)}}(f_{\max} - f_{\rm low})\,\epsilon^{-\frac{p+1}{p}}\ \mid \mathcal{K}_1 \text{ and } N_1(\epsilon) > 2k_\diamond + 2\right] \geq \pi_1^{(1)},$$

Since j is arbitrary between $2k_\diamond + 1$ and $N_1(\epsilon) - 2$, we obtain that

$$\mathbb{P}\left[N_1(\epsilon) \leq \frac{\kappa_2}{(1-\delta_1)\pi_S^{(1)}}(f_{\max} - f_{\rm low})\,\epsilon^{-\frac{p+1}{p}} + k_\diamond + 2\ \mid \mathcal{K}_1 \text{ and } N_1(\epsilon) > 2k_\diamond + 2\right] \geq \pi_1^{(1)},$$

which, given the definitions of κ_2 in (40), of $\kappa_{\rm SKOFFARp}$ in (34) and inequality (36), yields that

$$\mathbb{P}\left[N_1(\epsilon) \leq \kappa_{\rm SKOFFARp}\,\epsilon^{-\frac{p+1}{p}} + 4\ \mid \mathcal{K}_1 \text{ and } N_1(\epsilon) > 2k_\diamond + 2\right] \geq \pi_1^{(1)}.$$

Therefore, from (37), the fact that

$$\mathbb{P}\Big[\mathcal{K}_1 \mid N_1(\epsilon) > 2k_\diamond + 2\Big] \geq \mathbb{P}\Big[\mathcal{K}_1 \mid N_1(\epsilon) > k_\diamond\Big]$$

and Lemma 3.12, we finally obtain that

$$\begin{aligned}
&\mathbb{P}\left[N_1(\epsilon) \leq \kappa_{\rm SKOFFARp}\,\epsilon^{-\frac{p+1}{p}} + 4\right] \\
&\quad= \mathbb{P}\left[N_1(\epsilon) \leq \kappa_{\rm SKOFFARp}\,\epsilon^{-\frac{p+1}{p}} + 4\ \mid \mathcal{K}_1 \text{ and } N_1(\epsilon) > 2k_\diamond + 2\right] \\
&\qquad\quad \times \mathbb{P}\Big[\mathcal{K}_1 \mid N_1(\epsilon) > 2k_\diamond + 2\Big] \times \mathbb{P}\Big[N_1(\epsilon) > 2k_\diamond + 2\Big] + 1 \times \mathbb{P}\Big[N_1(\epsilon) \leq 2k_\diamond + 2\Big] \\
&\quad\geq \mathbb{P}\left[N_1(\epsilon) \leq \kappa_{\rm SKOFFARp}\,\epsilon^{-\frac{p+1}{p}} + 4\ \mid \mathcal{K}_1 \text{ and } N_1(\epsilon) > 2k_\diamond + 2\right] \times \mathbb{P}\Big[\mathcal{K}_1 \mid N_1(\epsilon) > k_\diamond\Big] \\
&\quad\geq (\pi_1^{(1)})^2.
\end{aligned}$$

Substituting the values of $\pi_1^{(1)}$ given by (33) in this inequality then yields (35). □

We now comment on this result.

1. As in the methods of [28] and [6], it is not necessary to evaluate the full-space derivatives $\{\nabla_x^j f(x_k)\}_{j=1}^p$ because only their sketched versions $\{\nabla_x^j f(x_k)[S_k\cdot]^j\}_{j=1}^p$ are used. As a consequence, the cost of evaluating the derivatives (not to mention that of computing the step) is potentially reduced typically by a significant factor ℓ/n. We discuss below whether this advantage may be offset by the choice of ω in AS.6.

2. Because it is proved in [20, Theorem 3.12] that the $\mathcal{O}(\epsilon^{-(p+1)/p})$ order bound for finding ϵ-approximate critical points is sharp for the OFFARp algorithm, the same is also true for Theorem 3.13 above, because SKOFFARp subsumes[5] OFFARp if $S_k = I$ for all k.

[5] The different conditions on the regularisation parameter σ_k only result in differences in the constants.

3. Considered as a worst-case evaluation complexity bound for $p = 2$, the order bound $\mathcal{O}(\epsilon^{-3/2})$ is known to be optimal for a large class of methods using first- and second-derivatives [9], justifying the title of this paper.

4. Note that (29) and (32) not only imply (36), but also that $k_\diamond$ is at least a (significant) fraction of $\epsilon^{-(p+1)/p}$, which, for meaningul values of ϵ, is a reasonably large number. Moreover, $(k_\epsilon - k_\diamond)$ is expected to be at least of the same order. Thus the factor

$$\left(1 - e^{-\frac{\delta_1^2}{2}\pi_S^{(1)}k_\diamond}\right)$$

   in the right-hand side of (35) is expected to be very close to 1.

5. The parameter δ_1, which we are still free to choose in (0,1) occurs in (34) and in the exponentials of (35). A quick calculation indicates that choosing δ_1 close to 1 can improves the bound on the right-hand side of (35) (although marginally because of our previous comment) while its possibly detrimental effect on (34) occurs because of the factor $1/(1-\delta_1)$ which must be kept bounded. Given the magnitude of the other factors in these formulae, values such as $\delta_1 = \frac{1}{2}$ or $\delta_1 = \frac{1}{10}$ could be considered acceptable.

6. As can be expected, the conditions for a random embedding given by (43) and AS.6 have a significant impact on the result, which significantly degrades if ω and/or $\pi_S^{(1)}$ tends to zero.

7. As we have mentioned above, the objective function is not evaluated by the SKOFFARp algorithm and the trial point $x_k + s_k$ is always accepted as the next iterate. Thus no distinction is necessary in the stochastic analysis between "successful" iterations (where the step is accepted because the objective function has decreased enough) and "unsuccessful" ones. This distinction had however to be taken into account in the analysis of [28] for more standard trust-region and adaptive-regularisation methods using functions values, leading to several different types of iterations whose numbers have to be bounded.

4 Selecting random subspaces

We now turn to ways in which ω-true iterations can be shown to happen with suitable probability $\pi_S^{(1)}$, thereby satisfying AS.6. A natural approach is to rely on Johnson-Lindenstrauss embeddings and results are available in the literature for $p \in \{1,2\}$. Restricting ourselves to such values of p and using [28, Definition 5.3.1] (see also [32], for instance), we say that, for some given "preservation parameter" $\alpha_S \in (0,1)$ and for some positive scalar $S_{\max}$ independent of k, iteration k is $(\alpha_S, S_{\max})$-embedded whenever

$$\|S_k\| \le S_{\max}, \tag{41}$$

and for

$$M_k \stackrel{\rm def}{=} [g_k, H_k] \in \mathbb{R}^{n \times n+1}, \tag{42}$$

we have that

$$\|S_k M_k z\| \ge \alpha_S \|M_k z\| \quad \text{for all} \quad z \in \mathbb{R}^{n+1}, \tag{43}$$

where $H_k = \nabla_s^2 f(x_k)$ if $p = 2$ and $H_k = 0_{n\times n}$ if $p = 1$. This condition is said to define a *one-sided random embedding* of the second-order Taylor's series.

Given such a one-sided random embedding, we now adapt an argument of [22] and verify that (27) holds at $(\alpha_S, S_{\max})$-embedded iterations.

Lemma 4.1 Suppose that $p \in \{1,2\}$, $p! \le L_p$, that AS.1, AS.2, AS.3, AS.4 and AS.5 hold and that iteration $k \ge 0$ of the SKOFFARp algorithm is $(\alpha_S, S_{\max})$-embedded (in the sense of (43)). Then

$$\|s_k\|^p \ge \frac{p!\,\alpha_S}{\alpha_S L_p + \theta S_{\max}\sigma_{\max}} \|g_{k+1}\|. \tag{44}$$

Thus iteration $k \in \{0,\ldots,N_1(\epsilon)-2\}$ is ω-true (in the sense of (27)) with $\omega = \frac{p!\,\alpha_S}{\alpha_S L_p + \theta S_{\max}\sigma_{\max}}$.

Proof. First note that applying the chain rule gives that

$$\nabla_{\widehat{s}}^1 \widehat{T}_{f,p}(x_k, \widehat{s}_k) = S_k \nabla_s^1 T_{f,p}(x_k, s_k) = S_k(g_k + H_k s_k) = S_k M_k (1, s_k^T)^T$$

and, since the iteration k is $(\alpha_S, S_{\max})$-embedded, (43) gives that

$$\|\nabla_{\widehat{s}}^1 \widehat{T}_{f,p}(x_k, \widehat{s}_k)\| \ge \alpha_S \|M_k(1, s_k^T)^T\| = \alpha_S \|\nabla_s^1 T_{f,p}(x_k, s_k)\|.$$

Condition (9), the definition $s_k = S_k^T \widehat{s}_k$ and (41) then yield that

$$\|\nabla_s^1 T_{f,p}(x_k, s_k)\| \le \frac{\|\nabla_{\widehat{s}}^1 \widehat{T}_{f,p}(x_k, \widehat{s}_k)\|}{\alpha_S} \le \frac{\theta \frac{\sigma_k}{p!} S_{\max} \|S_k^T \widehat{s}_k\|^p}{\alpha_S} \le \frac{\theta S_{\max}\sigma_k}{p!\,\alpha_S} \|s_k\|^p. \tag{45}$$

Successively using the triangle inequality, condition (45) and (13) (for $p \in \{1,2\}$), we deduce that

$$\|g_{k+1}\| \le \|g_{k+1} - \nabla_s^1 T_{f,p}(x_k, s_k)\| + \|\nabla_s^1 T_{f,p}(x_k, s_k)\| \le \frac{1}{p!} L_p \|s_k\|^p + \frac{\theta S_{\max}\sigma_k}{p!\,\alpha_S} \|s_k\|^p.$$

The inequality (44) follows by rearranging the terms and using the bound (25) in Lemma 3.8. That iteration k is ω-true for $k \in \{0,\ldots,N_1(\epsilon)-2\}$ follows from the fact that, by definition, $\|g_{k+1}\| \ge \epsilon$ for these values of k. □

Although this lemma essentially recovers the result of [28, Lemma 5.3.2], its proof is considerably simpler. Note that (44) is significantly stronger than (27), suggesting that (43) might itself be stronger than necessary. Also observe that we could replace condition (9) by the more permissive

$$\|\nabla_{\widehat{s}}^1 \widehat{T}_{f,p}(x_k, \widehat{s}_k)\| \le \theta \frac{\sigma_k}{p!} \|S_k\|\, \|S_k^T \widehat{s}_k\|^p$$

or

$$\|\nabla_{\widehat{s}}^1 \widehat{T}_{f,p}(x_k, \widehat{s}_k)\| \le \theta \frac{\sigma_k}{p!} \kappa_{S,k}\, \|S_k^T \widehat{s}_k\|^p$$

without altering the above theory, but at the price of computing $\|S_k\|$ or estimating a uniform bound on $\kappa_{S,k}$ (such as $S_{\max}$).

It is also possible to apply Shao's approach to "sparse Hessians" (for $p = 2$) as follows. For some constants (α_S, γ_S) such that $\alpha_S \in (0,1)$ and $\gamma_S \in [0, 2\alpha_S)$ and $S_{\max} > 0$, we now (re)define iteration k to be $(\alpha_S, \gamma_S, S_{\max})$-embedded whenever

$$\|S_k\| \le S_{\max}, \quad \|S_k g_k\| \ge \alpha_S \|g_k\| \quad \text{and} \quad \|S_k H_k\| \le \sqrt{\gamma_S \|g_{k+1}\|}. \tag{46}$$

We then obtain the following result based on [28, Lemma 5.4.1].

Lemma 4.2 Suppose that AS.1 and AS.3 hold and that, for a particular realization, iteration $k \ge 0$ of the SKOFFAR2 algorithm is $(\alpha_S, \gamma_S, S_{\max})$-embedded (in the sense of (46)). Then (27) holds and iteration k is ω-true.

Proof. Let $a = \|S_kH_k\|$. Then (9) gives that

$$\alpha_S\|g_k\| \le \|S_kg_k\| \le \|S_k(g_k+H_ks_k)\| + \|S_kH_ks_k\| \le \tfrac{1}{2}\theta S_{\max}\sigma_k\|s_k\|^2 + a\|s_k\|,$$

and therefore, using the triangle inequality, (13) and the fact that iteration k is $(\alpha_S, \gamma_S, S_{\max})$-embedded,

$$\alpha_S\|g_{k+1}\| \le \alpha_S\|g_{k+1}-g_k\| + \alpha_S\|g_k\| \le \tfrac{1}{2}\alpha_S L_2\|s_k\|^2 + \tfrac{1}{2}\theta S_{\max}\sigma_k\|s_k\|^2 + a\|s_k\|.$$

Defining $b = \alpha_S L_2 + \theta S_{\max}\sigma_k$, we obtain that

$$\|s_k\|^2 + \left(\frac{2a}{b}\right)\|s_k\| - \frac{2\alpha_S\|g_{k+1}\|}{b} \ge 0,$$

yielding that

$$\Big(\|s_k\| + \frac{a}{b}\Big)^2 \ge \frac{2\alpha_S\|g_{k+1}\|}{b} + \Big(\frac{a}{b}\Big)^2$$

and thus that

$$\|s\| \ge \sqrt{\frac{2\alpha_S\|g_{k+1}\|}{b} + \Big(\frac{a}{b}\Big)^2} - \frac{a}{b}.$$

Assuming, without loss of generality, that $b = \alpha_S L_2 + \theta S_{\max}\sigma_k \ge 1$, we deduce that

$$\|s\| \ge \frac{1}{b}\left[\sqrt{2\alpha_S\|g_{k+1}\| + a^2} - a\right]$$

Since the function $\sqrt{c+t^2} - t$ (for $c > 0$) is decreasing as a function of $t \ge 0$ and since $a = \|S_kH_k\| \le \sqrt{\gamma_S\|g_{k+1}\|}$ because iteration k is (α_S, γ_S)-true, we deduce that

$$\begin{aligned}
\|s\| &\ge \frac{1}{b}\left[\sqrt{2\alpha_S\|g_{k+1}\| + \gamma_S\|g_{k+1}\|} - \sqrt{\gamma_S\|g_{k+1}\|}\right] \\
&\ge \frac{\sqrt{2\alpha_S} - \sqrt{\gamma_S}}{\alpha_S L_2 + \theta S_{\max}\sigma_k}\sqrt{\|g_{k+1}\|} \\
&\ge \frac{\sqrt{2\alpha_S} - \sqrt{\gamma_S}}{\alpha_S L_2 + \theta S_{\max}}\sigma_{\max}\sqrt{\|g_{k+1}\|},
\end{aligned}$$

where we again used Lemma 3.8 to obtain the last inequality. □

Thus, an $(\alpha_S, \gamma_S, S_{\max})$-embedded iteration (in the sense of (46)) is ω-true (in the sense of (27)) for $\omega = (\sqrt{2\alpha_S} - \sqrt{\gamma_S})/(\alpha_S L_2 + \theta S_{\max}\sigma_{\max})$. Also notice that, should we replace (46) by

$$\|S_k\| \le S_{\max}, \quad \|S_kg_k\| \ge \alpha_S\|g_k\| \quad \text{and} \quad \|S_kH_k\| \le \sqrt{\gamma_S\epsilon} \quad \text{for} \quad k < N_1(\epsilon) - 1, \tag{47}$$

then the definition of an $(\alpha_S, \gamma_S, S_{\max})$-embedded iteration is closer to that of [28], obviously ensuring (46) with a right-hand side of its third part now independent of S_k.

The reader may now recall that AS.6 states that (27), (43), (46) or (47) (or the first part of (46) or (47)) should hold at iteration k with positive probability $\pi_S^{(1)}$. In [28, Lemma 5.3.1] or [32, Theorem 2.3] (see also [30, Lemma 3.1]) it is argued that by choosing $\mathcal{S}$ to be the distribution of $\ell \times n$ scaled Gaussian matrices, (43) holds with probability

$$\pi_S^{(1)} = 1 - e^{-\frac{\ell(1-\alpha_S)}{C_\ell} + \operatorname{rank}(M_k)}, \tag{48}$$

where $C_\ell > \frac{1}{4}$ is an absolute constant.

Unfortunately, the expression (48) requires that

$$\operatorname{rank}(M_k) < \frac{\ell(1-\alpha_S)}{C_\ell}, \tag{49}$$

thereby limiting the applicability of the result for $p > 1$ when considering general problems with full-rank Hessians. But this can be acceptable for a class of problems with low-rank Hessian, as we illustrate in Section 5. Satisfying the third part of (47) with positive probability is also possible when H_k is very sparse, which also imposes a significant restriction. Other choices for the distribution exist, such as hashing, scaled hashing, sampling matrices, or "fast Lindenstrauss transforms" (see [28, Chapter 2] or [32, page 16]). Although possibly more economical in terms of algebraic operations, they appear to suffer from the same geometric precondition: their number of rows ℓ should be of the order of the Hessian's rank, which is problematic for the general case where the Hessian is full rank. However, note that $\mathrm{rank}(M_k) = 1$ when $p = 1$, essentially avoiding this problem, making the first-order variant of the algorithm applicable to a much larger class of problems.

Should one be ready to trade the optimal complexity for getting rid of the low-rank requirement, an algorithm using quadratically regularised quadratic models with inexact Hessians can also be defined and analysed (see Appendix). Under suitably modified assumptions, the evaluation complexity of this algorithm can be shown to be of order $\mathcal{O}(\epsilon^{-2})$, matching the theoretical results of [6] for a random subspace version of the adaptive-regularisation algorithm *using function values*. Unfortunately, our numerical experience matches the cautious conclusions of this reference, which is why we do not investigate it further.

Finally, note that the constant (34) involves $S_{\max}^{\frac{p+1}{p}}$ due to its dependence on $\omega^{\frac{p+1}{p}}$. In the case of scaled Gaussian matrices, we know that

$$S_{\max} \le \beta \stackrel{\rm def}{=} 1.5 + \sqrt{n/\ell} \tag{50}$$

with high probability for the values considered of δ_1 (see [28, Lemma 4.4.4] for instance), resulting in a dependence of the constant (34) on $(n/\ell)^{\frac{p+1}{2p}}$. For $p = 1$, this offsets, complexity-wise, the benefit of cheaper gradient evaluations by a factor of ℓ/n, but the complexity bound is rarely tight and savings in gradient evaluations are sometimes possible in practice. For $p = 2$, the advantage of cheaper gradients (assuming (49)) increases compared to $p = 1$ because the denominator of (34) now depends on $(n/\ell)^{3/4}$. We show in Section 5 that this theoretical advantage translates into significantly better numerical behaviour. Moreover, the cost of computing the step s_k is reduced by the decrease in dimension of the linear system. The advantage of cheaper gradients grows when p grows (and the method is applicable).

5 Numerical illustration

We now numerically illustrate the behaviour of SKOFFAR$_2$, the second-order version of SKOFFARp. We report results obtained using Matlab R2024a for 14 problems from the CUTEst test problems [18] as provided in Matlab by OPM [21]. All problems except `arglina` and `tridia` are nonconvex. The original dimension of the problem, say $\hat{n}$, was enlarged using the affine transformation $x = A\hat{x}$, $\hat{x} \in R^{\hat{n}}$, $x \in R^n$, $n \ge \hat{n}$, $A \in R^{n\times\hat{n}}$ being an orthonormal matrix generated by the (Matlab supplied) discrete cosine transform, therefore yielding problems with Hessians of rank at most $\hat{n}$ and ensuring (49) when n grows.

We ran a Matlab implementation of a modified version of the SKOFFARp where we defined $\mathcal{S}$ to be the distribution of $\ell \times n$ scaled Gaussian matrices. The first modification is identical to that described in [20] for the OFFARp algorithm, in that (6) is replaced by

$$\sigma_k = \max[\vartheta\nu_k, \xi_k\mu_k]$$

where $\xi_k \in (0, 1)$ is an adaptive scaling parameter (see [20] for details) and where μ_k is defined by (7) with $\mu_{-1} = 10^3$. The second change avoids the (potentially very) costly computation of $\|S_k\|$ by using $\kappa_{S,k} = \beta$ as given by (50). This change was made after running the more expensive code using $\kappa_{S,k} = \|S_k\|$ as suggested by (19) on a few problems and observing that the results obtained with the theoretically weaker $\kappa_{S,k} = \beta$ did not degrade the code efficiency, if at all. The regularised quadratic was minimized approximately ($\theta = 1.01(1 + \sqrt{n/\ell})$ using a Lanczos-based solver for such functions (see [11, Section 10.2]). We also chose $\vartheta = 10^{-3}$ and terminated the optimization as soon as the threshold $\|g_k\| \leq 10^{-3}$ was reached. All computations were performed on a Dell Precision laptop with 16 cores at 2.6 GHz and 62.5 GB of memory, running Matlab 2024a with Ubuntu 20.04.6 LTS.

For each run, we computed the number of gradient evaluations weighted to reflect the reduced evaluation cost in the subspace of dimension ℓ. We counted the cost of evaluating a Hessian as the product of the dimension times the cost of evaluating one gradient (as happens for finite-difference approximations), the weighted cost of an iteration (now involving the computation of one gradient and one Hessian) then becoming $w_2(\tau, n) = (\tau + n\tau^2)/(1+n)$, where $\tau = \ell/n$. Thus this w_2 weighting reflects the cost of running the second-order SKOFFAR$_2$ with sketching parameter τ compared to running it in full-dimension. To take this into account, the maximum number of iterations was set to 10^5 divided by this factor. Table 1 reports the average w_2-weighted iteration costs for SKOFFAR$_2$ to reach convergence for $n = 1000\,\hat{n}$ and for decreasing values of the ratio τ from 1 (irrealistic) to 10^{-3}, averaged over 10 independent runs.

		SKOFFAR$_2$						
Problem	n	$\tau = 1$	10^{-1}	$5 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	10^{-2}	5.10^{-3}	10^{-3}
arglina	10000	19.6000	0.8358	0.3394	0.1399	0.0443	0.0188	0.0027
arwhead	10000	6.7000	0.1191	0.0363	0.0125	0.0033	0.0013	0.0002
broyden3d	10000	7.0000	0.1441	0.0393	0.0161	0.0054	0.0020	0.0004
chandheu	10000	6.0000	0.1231	0.0436	0.0160	0.0062	0.0032	0.0006
dixmaana	12000	13.0000	0.4253	0.2429	0.1290	0.0510	0.0244	0.0041
eg2	10000	5.3000	0.1611	0.0614	0.0233	0.0068	0.0028	0.0004
engval2	3000	18.4000	0.4082	0.2493	0.1323	0.0558	0.0275	0.0055
helix	10000	30.8000	1.7726	0.8606	0.4182	0.1892	0.1234	0.0241
kowosb	10000	2715.9000	152.9072	64.4489	27.2542	8.8252	3.8179	0.6299
nzf1	13000	104.9000	7.3581	3.2245	1.4262	0.4701	0.2006	0.0297
rosenbr	10000	98 .7000	7.4097	3.7827	1.6012	0.5338	0.2675	0.0474
sensors	10000	18.0000	0.8398	0.3399	0.1393	0.0443	0.0192	0.0029
tridia	10000	14.7000	0.5565	0.2367	0.0961	0.0272	0.0112	0.0029
watson	10000	44.0000	3.9886	1.9084	0.8106	0.2847	0.1342	0.0146

Table 1: Using the SKOFFAR$_2$ algorithm: average w_2-weighted number of iterations for varying ratio $\tau = \ell/n$.

The results in Table 1 show that the use of random subspaces can bring substantial benefits, as long as (49) holds. In this context, SKOFFAR$_2$ is reliable and efficient on nearly all problems (except for kowosb) and tested values of τ. Performance globally increases with decreasing values of τ; it appears to be best for the smallest value $\tau = 10^{-3}$. Limited to the set of problems considered, these results show that sketching pays off handsomely in terms of gradient evaluations when used with low-rank Hessians and second-order models. This, admittedly, ignores the cost of linear algebra, which increases because products with S_k have to be computed but also decreases because the calculation of the step in the subspace is significantly cheaper than in the full space.

The reader might wonder at this point how (sketched) second-order methods might compare to standard first-order algorithms, where the Hessian is not evaluated. We attempt to clarify this question by comparing, in Table 2, our results for SKOFFAR$_2$ with those obtained by the well-known objective-function-free ADAGRAD-Norm algorithm [16, 31] and the norm-wise variant of ADAM[6] [25]. To make the comparison fair, we have re-weighted the iteration counts in order to make them relative to a single gradient evaluation (as is case for one iteration of ADAGRAD and ADAM) by using $w_1(\tau, n) = \tau + n\tau^2$. In this table, the string ">100000" indicates that the maximum number of iterations was reached.

[6] Using the momentum discounting factor $\beta = 0.9999$. It failed to converge on most problems with smaller values.

Problem	n	ADAM-N	ADAG-N	SKOFFAR$_2$ 10^{-1}	$5 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	10^{-2}	5.10^{-3}	10^{-3}
`arglina`	10000	125	126	8358	3394	1399	443	118	27
`arwhead`	10000	45	45	1191	363	125	33	13	2
`broyden3d`	10000	40	40	1441	393	160	54	20	4
`chandheu`	10000	51	51	1231	435	160	62	32	6
`dixmaana`	12000	697	710	5104	2915	1549	612	293	47
`eg2`	10000	106	104	1611	614	233	68	28	4
`engval2`	3000	> 100000	19266	1225	748	397	168	83	16
`helix`	10000	26142	53907	17727	8607	4183	1892	1234	241
`kowosb`	10000	295	296	611781	257860	109043	35309	15275	2520
`nzf1`	13000	8335	10323	95662	41921	18540	6112	2608	387
`rosenbr`	10000	26748	56173	74104	37830	16013	5338	2675	474
`sensors`	10000	189	167	8393	3399	1393	443	192	29
`tridia`	10000	50	50	5566	2367	962	272	112	29
`watson`	10000	> 100000	15132	39889	19085	8107	2848	1342	146

Table 2: Using the SKOFFAR$_2$ algorithm: average w_1-weighted number of iterations for varying ratio $\tau = \ell/n$.

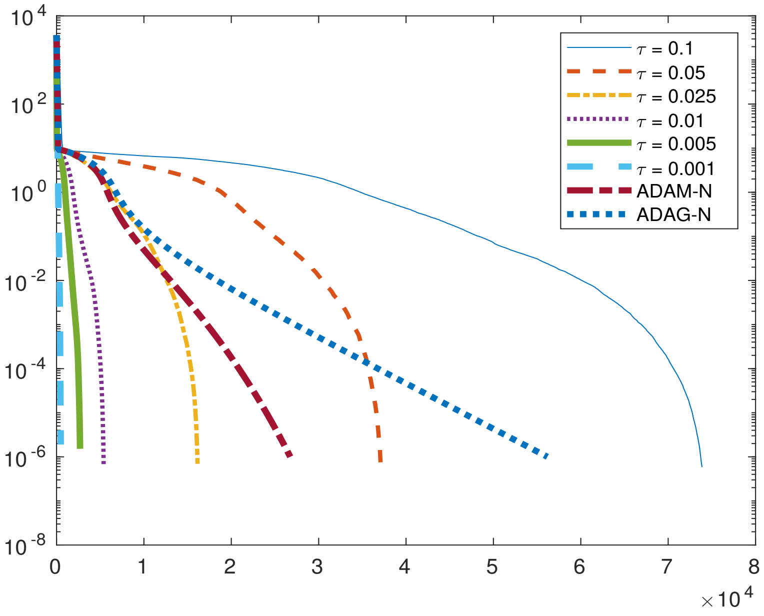

Figure 1: The behaviour of $f(x)$ when ADAM-N, ADAG-N and SKOFFAR$_2$ are run on `rosenbr`, as a function of w_1-weighted iteration numbers, where SKOFFAR$_2$ uses $\tau = 10^{-1}, 5 \cdot 10^{-2}, 2.5 \cdot 10^{-2}, 10^{-2}, 5.10^{-3}$ and 10^{-3} (from right to left)

These re-weighted results indicate that *sketched second-order methods may be competitive with existing first-order algorithms when using a small τ for problems with low-rank Hessians.* We also note the difference of performance between ADAGRAD and ADAM: while the latter may be more efficient when it works, it is less reliable than the former (as predicted by the theory). To provide further intuition, we also show, in Figure 1, how the value of the objective function evolves (although it is never computed in the course of the algorithm) for one instance of applying ADAM, ADAGRAD and SKOFFAR$_2$ to the `rosenbr` problem, the latter with various choices of $\tau < 1$. Beyond the clearly faster convergence for smaller τ, one also notices the concave nature of the curves, which contrasts with the convex curves one would often expect when using first-order methods.

(The nearly vertical part of the curves between 10^4 and 10^1 corresponds to the first phase of minimization where all algorithms reach for the bottom of the steep, curving valley that is typical of `rosenbr`.) Because of this concavity, one also sees that requesting higher accuracy in $\mathsf{SKOFFAR}_2$ is unlikely to require many more iterations.

6 Conclusions and perspectives

We have introduced an adaptive-regularisation algorithm for nonconvex unconstrained optimization that uses random subspaces and never computes the objective function's value, and have shown that its evaluation complexity is, in order, identical to that of the "optimal" adaptive full-space regularisation methods using function values. The analysis covers finding approximate first-order critical points, but it is possible to extend the algorithm to ensure second-order criticality (along the lines of the MOFFAR algorithm in [20]), albeit at the price of a very strong assumption on the recovery of the Hessian's minimum eigenvalue in random subspaces, a notoriously thorny problem (see [4, Section 4.2.3], for instance). Our analysis also allows the use of models of arbitrary degree, but this generality may be of limited practical use since using degree higher than two appears to be mostly applicable to problems with low-rank, very sparse or partially separable derivatives.

Our theoretical and numerical results show that the approach is theoretically sound and that it can be significantly advantageous when its second-order variant is used on problems with low-rank Hessians.

Data availability

The test problems used in this study are available at `https://github.com/gratton7/OPM`.

Acknowledgement

Ph. Toint acknowledges the partial support of INDAM (through the GNCS grant CUP E53C22001930001) during his visit to the Department of the Industrial Engineering (DIEF) at the University of Florence in the fall 2023. He is also grateful for the continued and friendly support of DIEF and ANITI (Toulouse).

References

[1] S Bellavia, G Gurioli, B Morini and Ph. L.Toint. *Adaptive regularization for nonconvex optimization using inexact function values and randomly perturbed derivatives.* Journal of Complexity 68, 101591, 2022.

[2] S Bellavia, N Krejić, B Morini and S Rebegoldi. *A stochastic first-order trust-region method with inexact restoration for finite-sum minimization.* Computational Optimization and Applications, 84(1);53–84, 2023.

[3] R. Bollapragada and S. M. Wild. *Adaptive sampling quasi-Newton methods for zeroth-order stochastic optimization.* Mathematical Programming Computation, 15(2):327–364, 2023.

[4] E. Boman. *Infeasibility and Negative-Curvature Directions in Optimization.* Ph.D Thesis, Stanford University, California, USA, 1999.

[5] A. S. Berahas, L. Cao, and K. Scheinberg. *Global convergence rate analysis of a generic line search algorithm with noise.* SIAM Journal on Optimization 31(2):1489–1518, 2021

[6] C. Cartis, J. Fowkes, and Z. Shao. *Randomised subspace methods for non-convex optimization, with applications to nonlinear least-squares*, arXiv:2211.09873, 2022.

[7] C. Cartis, Z. Shao, and E. Tansley, *Random Subspace Cubic-Regularization Methods, with Applications to Low-Rank Functions*, arXiv:2501.09734, 2025.

[8] C. Cartis, N. I. M. Gould, and Ph. L. Toint. *Adaptive cubic overestimation methods for unconstrained optimization. Part I: motivation, convergence and numerical results.* Mathematical Programming, Series A, 127(2):245–295, 2011.

[9] C. Cartis, N. I. M. Gould, and Ph. L. Toint. *Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization.* In B. Sirakov, P. de Souza, and M. Viana, editors, Invited Lectures, Proceedings of the 2018 International Conference of Mathematicians (ICM 2018), vol. 4, Rio de Janeiro, pages 3729–3768. World Scientific Publishing Co Pte Ltd, 2018.

[10] C. Cartis, N. I. M. Gould, and Ph. L. Toint. *Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints.* SIAM Journal on Optimization, 30(1):513–541, 2020.

[11] C. Cartis, N. I. M. Gould, and Ph. L. Toint. *Evaluation complexity of algorithms for nonconvex optimization.* Number 30 in MOS-SIAM Series on Optimization. SIAM, Philadelphia, USA, June 2022.

[12] A. R. Conn and N. I. M. Gould and Ph. L. Toint. LANCELOT*: a Fortran package for large-scale nonlinear optimization (Release A).* Springer Series in Computational Mathematics, 17, 1992.

[13] F. E. Curtis and K. Scheinberg. *Adaptive stochastic optimization: A framework for analyzing stochastic optimization algorithms.* IEEE Signal Processing Magazine 37(5):32–42, 2020.

[14] S. Dasgupta and A. Gupta. *An elementary proof of a theorem of Johnson and Lindenstrauss.* Random structures Algorithms, 21(1):60–65, 2002.

[15] K. J. Dzahini and S M. Wild. *Stochastic trust-region algorithm in random subspaces with convergence and expected complexity analyses.* arXiv2207.06452, 2022.

[16] J. Duchi, E. Hazan, and Y. Singer. *Adaptive subgradient methods for online learning and stochastic optimization.* Journal of Machine Learning Research, 12, 2011.

[17] Z. Feng, F. Roosta, D. P. Woodruff. *Non-PSD Matrix Sketching with Applications to Regression and Optimization.* Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021), PMLR. 161:1841–1851, 2021.

[18] N. I. M. Gould and D. Orban and Ph. L. Toint. CUTEst*: a Constrained and Unconstrained Testing Environment with safe threads for mathematical optimization*, Computational Optimization and Applications, 60:545–557, 2015.

[19] G. N. Grapiglia and G. F. D. Stella. *An adaptive trust-region method without function evaluation.* Computational Optimization and Applications, 82:31–60, 2022.

[20] S. Gratton, S. Jerad, and Ph. L. Toint. *Convergence properties of an objective-function-free optimization regularization algorithm, including an $\mathcal{O}(\epsilon^{-3/2})$ complexity bound.* SIAM Journal on Optimization, 33(3):1621–1646, 2022.

[21] S. Gratton and Ph. L. Toint. OPM*, a collection of Optimization Problems in Matlab*, arXiv:2112.05636, 2021.

[22] S. Gratton and Ph. L. Toint. *Adaptive regularization minimization algorithms with non-smooth norms.* IMA Journal of Numerical Analysis, 43(2):920-949 , 2023.

[23] S. Gratton, S. Jerad, and Ph. L. Toint, *Convergence properties of an Objective-Function-Free Optimization regularization algorithm, including an $\mathcal{O}(\epsilon^{-3/2})$ complexity bound*, SIAM Journal on Optimization, 33(3):1621–1646, 2023.

[24] A. Griewank and Ph. L. Toint. *Partitioned variable metric updates for large structured optimization problems*, Numerische Mathematik, 39:119–137, 1982.

[25] D. Kingma and J. Ba. *Adam: A method for stochastic optimization.* Proceedings in the International Conference on Learning Representations (ICLR), 2015.

[26] J. Nocedal and S. J. Wright, *Numerical Optimization.* Springer Verlag, Heidelberg, Series in Operations Research, 1999.

[27] K. Scheinberg and M. Xie. *Stochastic Adaptive Regularization Method with Cubics: A High Probability Complexity Bound.* arXiv: 2308.13161, 2023.

[28] Z. Shao. *On Random Embeddings and Their Application to Optimization.* PhD thesis, University of Oxford, Oxford, UK, 2021.

[29] T. Tieleman and G. Hinton. Lecture 6.5-RMSPROP. COURSERA: Neural Networks for Machine Learning, 2012.

[30] K. Vu, P.-L. Poiron, C. D'Ambrosio, and L. Liberti. *Random projections for quadratic programs over a Euclidean ball.* Integer Programming and Combinatorial Optimization (IPCO), Ann Arbour (USA), 2019. HAL-02869206.

[31] R. Ward, X. Wu, and L. Bottou. *Adagrad stepsizes: sharp convergence over nonconvex landscapes.* Proceedings in the International Conference on Machine Learning (ICML2019), 2019.

[32] D. P. Woodruff. *Sketching as a tool for numerical linear algebra.* Foundations and Trends in Theoretical Computer Science, 10(1-2):1–157, 2014.

[33] X. Wu, R. Ward, and L. Bottou. *WNGRAD: Learn the learning rate in gradient descent.* arXiv:1803.02865, 2018.

A Quadratic regularisation for approximate second-order models

We discuss here a context in which the low-rank assumption is unnecessary and, motivated by [6], consider using quadratic regularisation in conjunction with approximate quadratic models in which the Hessian $\nabla_x^2 f(x)$ is approximated by a positive-semidefinite symmetric matrix B_k. At x_k, the regularised model $m_{k,B}(s)$ of $f(x_k+s)$ then takes the form

$$m_{k,B}(s) \stackrel{\rm def}{=} T_{k,B}(x_k,s) + \frac{\sigma_k}{2}\|s\|^2, \tag{51}$$

with

$$T_{k,B}(x_k,s) \stackrel{\rm def}{=} f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s. \tag{52}$$

To make the use of this model well-defined, we complete AS.2, AS.4 (for $p=1$) and AS.6 and make the following assumptions.
AS.7 f is continuously differentiable in $\mathbb{R}^n$.
AS.8 The gradient of f is globally Lipschitz continuous, that is, there exist a non-negative constant L_1 such that

$$\|\nabla_x^1 f(x) - \nabla_x^1 f(y)\| \le L_1\|x-y\| \;\text{ for all } x,y \in \mathbb{R}^n.$$

AS.9 The matrix B_k is symmetric, positive-semidefinite and bounded for all $k \ge 0$, so that there exist a positive scalar κ_B such that

$$\|B_k\| \le \kappa_B \quad \text{for} \quad k \ge 0. \tag{53}$$

Notice that AS.9[7] prevents the quadratic model (51) to be unbounded below. In particular, the use of the Gauss-Newton Hessian approximation for nonlinear least-squares problem is covered by AS.9, as well as the use of several quasi-Newton updating formulae.

Proceeding as in SKOFFARp, we let S_k be drawn from an iteration-independent distribution $\mathcal{S}$ of $\ell \times n$ random matrices (with $\ell < n$), let $s = S_k^T \widehat{s}$ be the full dimensional step and consider minimizing the sketched regularised model

$$\widehat{m}_{k,B}(\widehat{s}) \stackrel{\rm def}{=} \widehat{T}_{k,B}(x_k,\widehat{s}) + \frac{1}{2}\sigma_k\|S_k^T\widehat{s}\|^2, \tag{54}$$

where

$$\widehat{T}_{k,B}(x_k,\widehat{s}) \stackrel{\rm def}{=} f(x_k) + g_k^T S_k^T \widehat{s} + \frac{1}{2}\widehat{s}^T S_k B_k S_k^T \widehat{s}.$$

We note that, similarly to (5), $\widehat{m}_{k,B}(\widehat{s}) = m_{k,B}(s)$. The resulting SKOFFAR2B algorithm is stated on the following page.

The evaluation complexity analysis for the SKOFFAR2B algorithm is very closely related to that of SKOFFARp, and we now discuss how the results of Section 3 can be adapted to the new context.

1. Restricting our use of the Lipschitz condition to the gradient ($p=1$), Lemma 3.1 now states that

$$f(x_{k+1}) - \widehat{T}_{k,B}(x_k,\widehat{s}_k) = f(x_{k+1}) - T_{f,p}(x_k,s_k) \le \frac{\kappa_{LB}}{2}\|s_k\|^2, \tag{57}$$

and

$$\|g_{k+1} - \nabla_s^1 T_{k,B}(x_k,s_k)\| \le \kappa_{LB}\|s_k\|,$$

where $\kappa_{LB} \stackrel{\rm def}{=} L_1 + \kappa_B$.

[7] Alternatively, we could replace the condition that B_k is positive-semidefinite by the weaker condition that $B_k + \sigma_k I$ is positive-semidefinite.

Algorithm A.1: OFFO adaptive regularisation with approximate second-order models (SKOFFAR2B)

Step 0: Initialization: An initial point $x_0 \in \mathbb{R}^n$, a regularisation parameter $\nu_0 > 0$ and a requested final gradient accuracy $\epsilon \in (0, 1]$ are given, as well as the parameters $\theta > 1, \mu_{-1} \geq 0$ and $0 < \vartheta < 1$. Set $k = 0$.

Step 1: Step calculation: If $k = 0$, set $\sigma_0 = \nu_0$. Otherwise, select a matrix B_k satisfying AS.9 and

$$\sigma_k \in \Big[\vartheta\nu_k, \max[\nu_k, \mu_k]\Big],$$

where

$$\mu_k = \max\left[\mu_{k-1}, \frac{\|S_{k-1}g_k\| - \|\nabla_s^1\widehat{T}_{k,B}(x_k, s_k)\|}{\kappa_{S,k-1}\cdot\|s_{k-1}\|}\right]$$

with some $\kappa_{S,k-1}$ such that $\|S_{k-1}\| \leq \kappa_{S,k-1}$. Draw a random matrix $S_k \in \mathbb{R}^{\ell\times n}$ from $\mathcal{S}$ and compute a step $s_k = S_k^T\widehat{s}_k$ such that $\widehat{s}_k$ sufficiently reduces the random model $\widehat{m}_{k,B}$ defined in (54) in the sense that

$$\widehat{m}_{k,B}(\widehat{s}_k) - \widehat{m}_{k,B}(0) < 0 \tag{55}$$

and

$$\|\nabla_{\widehat{s}}^1\widehat{T}_{k,B}(x_k, \widehat{s}_k)\| \leq \theta\sigma_k\|S_kS_k^T\widehat{s}_k\|. \tag{56}$$

Step 2: Updates. Set $x_{k+1} = x_k + s_k$ and $\nu_{k+1} = \nu_k + \nu_k\|s_k\|^2$. Increment k by one and go to Step 1.

2. Using now the decrease (55) of the model with quadratic regularisation, the decrease condition of Lemma 3.2 becomes

$$T_{k,B}(x_k, 0) - T_{k,B}(x_k, s_k) > \frac{\sigma_k}{2}\|s_k\|^2. \tag{58}$$

3. As in Lemma 3.3, we now exploit (57) to obtain that, if $\sigma_k \geq 2\kappa_{LB}$, then

$$f(x_k) - f(x_{k+1}) > \frac{\sigma_k}{4}\|s_k\|^2. \tag{59}$$

4. Lemma 3.4 is no longer valid because its assumes that the regularisation order is one above that of the highest derivative used, while both these orders are now equal to two. But a simple bound on the steplength can still be derived easily.

Lemma A.1 Suppose that AS.7 and AS.9 hold. At each iteration k, we have that

$$\|s_k\| \leq \frac{2\|g_k\|}{\vartheta\nu_0}. \tag{60}$$

Proof. Using (55) and $\widehat{m}_{k,B}(\widehat{s}_k) = m_{k,B}(s_k)$ it follows that

$$\frac{1}{2}\sigma_k\|s_k\|^2 \leq -g_k^Ts_k - \frac{1}{2}s_k^TB_ks_k \leq \|g_k\|\|s_k\|$$

and the thesis follows from the fact that $\sigma_k \geq \vartheta\nu_0$. □

5. The proof of Lemma 3.5 is easily adapted to the case where $p = 1$, yielding that, for all $k \geq 0$,
$$\mu_k \leq \max[\mu_{-1}, \kappa_{LB}].$$

6. The bounds (20) and (21) may now be re-writtten as $\nu_k \geq 2\kappa_{LB}/\vartheta$ and
$$k_1 \stackrel{\text{def}}{=} \min\left\{k \geq 1 \mid \nu_k \geq \frac{2\kappa_{LB}}{\vartheta}\right\},$$
respectively.

7. Using (60), Lemma 3.6 then becomes
$$\nu_{k_1} \leq \nu_{\max} \stackrel{\text{def}}{=} \frac{2\kappa_{LB}}{\vartheta}\left[1 + \left(\frac{\kappa_{\rm g}}{\vartheta\nu_0}\right)^2\right].$$

8. The revised version of inequality (24) in Lemma 3.7 is now given by
$$f(x_{k_1}) \leq f_{\max} \stackrel{\text{def}}{=} f(x_0) + \frac{1}{2}\left(\frac{\kappa_{LB}}{\sigma_0}\nu_{\max} + \vartheta\sigma_0\right), \tag{61}$$
and the bound (25) in Lemma 3.8 is now valid with
$$\sigma_{\max} \stackrel{\text{def}}{=} \max\left[\frac{4}{\vartheta}\left[f(x_0) - f_{\rm low} + \frac{1}{2}\left(\frac{\kappa_{LB}}{\sigma_0}\nu_{\max} + \vartheta\sigma_0\right)\right] + \nu_{\max}, \mu_{-1}, L_1 + \kappa_B, \frac{2\kappa_{LB}}{\vartheta}, \nu_0\right]. \tag{62}$$

9. It is of course necessary to revise our definition of a true iteration.

   Definition A.2 *Iteration $k \in \{0, \ldots, N_1(\epsilon) - 1\}$ is ω-true whenever,*
$$\|s_k\| \geq \omega\epsilon. \tag{63}$$
   We say that, for some given "preservation parameter" $\alpha_S \in (0, 1)$ and a constant $S_{\max} > 0$, iteration k is $(\alpha_S, S_{\max})$-embedded whenever,
$$\|S_k g_k\| \geq \alpha_S\|g_k\| \quad \text{and} \quad \|S_k\| \leq S_{\max}. \tag{64}$$

10. Lemma 3.10 remains valid with
$$k_* \stackrel{\text{def}}{=} \left\lceil \frac{2\kappa_{LB}\epsilon^{-2}}{\vartheta\nu_0\,\omega^2}\right\rceil \quad \text{and} \quad \sigma_k \geq 2\kappa_{LB}, \quad \text{for all} \quad k \geq k_1 \tag{65}$$
while Lemmas 3.11 and 3.12 are unchanged.

11. Since, for algorithm SKOFFAR2B, $\|g_k\| > \epsilon$ for all $k \leq N_1(\epsilon) - 1$ (instead of $\|g_{k+1}\| > \epsilon$ for $k \leq N_1(\epsilon) - 2$ for SKOFFARp), we may continue to use the proof of Lemma 3.13 and obtain the following evaluation complexity result for the SKOFFAR2B algorithm.

Theorem A.3 Suppose that AS.2, AS.4, AS.6, AS.7, AS.8 and AS.9 hold, that $\delta_1 \in (0,1)$ is given and that the SKOFFAR2B algorithm is applied to problem (1). Define

$$\kappa_{\rm SKOFFAR2B} \stackrel{\rm def}{=} \frac{4\,[L_1+\kappa_B+2(f_{\max}-f_{\rm low})]}{\vartheta\nu_0\omega^2(1-\delta_1)\pi_S^{(1)}} \tag{66}$$

where $f_{\max}$ is defined in (61). Then

$$\mathbb{P}\Big[N_1(\epsilon) \leq \kappa_{\rm SKOFFAR2B}\,\epsilon^{-2}+4\Big] \geq \left(1-e^{-\frac{\delta_1^2}{2}\pi_S^{(1)}k_\diamond}\right)^2$$

where $k_\diamond = \left\lceil \frac{k_*}{(1-\delta_1)\pi_S^{(1)}}\right\rceil$ with k_* given by (65).

Of course, using quite loose Hessian approximations in (51) has the consequence that the complexity order is now $\mathcal{O}(\epsilon^{-2})$, which is identical to that of other methods (such as deterministic and stochastic trust-region or regularisation) using the same type of approximations and objective function values.

12. We finally consider how Lemma 4.1 can be adapted for the use of Gaussian scaled matrices within the SKOFFAR2B algorithm.

Lemma A.4 Suppose that AS.4, AS.7, AS.8 and AS.9 hold and that iteration $k \geq 0$ of the SKOFFAR2B algorithm is $(\alpha_S, S_{\max})$-embedded (in the sense of (64)). Then

$$\|s_k\| \geq \frac{\alpha_S}{S_{\max}(\kappa_B+\theta\sigma_k)}\|g_k\|.$$

Thus iteration $k \in \{0,\ldots,N_1(\epsilon)-1\}$ is ω-true (in the sense of (63)) with $\omega = \frac{\alpha_S}{S_{\max}(\kappa_B+\theta\sigma_{\max})}$.

Proof. Since

$$S_k g_k = \nabla_s^1 \widehat{T}_{k,B}(x_k,\widehat{s}_k) - S_k B_k S_k^T \widehat{s}_k,$$

we obtain from (56) and the definition of $(\alpha_S, S_{\max})$-embedded iteration that

$$\alpha_S\|g_k\| \leq \|S_k g_k\| \leq S_{\max}(\kappa_B+\theta\sigma_k)\|s_k\|$$

Using the bound (62) yields the desired result. □

We see that the constant (66) now involves $S_{\max}^2$. In the case of scaled Gaussian matrices, (50) then gives a dependence of the constant (66) in n/ℓ, as is the case for the (non-OFFO) trust-region method of [6].

We conclude this discussion by noting that, should the Gauss-Newton method for nonlinear least-squares be considered, AS.3 (for $p=1$) and AS.4 can be replaced by assuming the Lipschitz continuity of the problem's Jacobian and the boundedness of the Jacobian and residual (see [26, page 295] for a proof that this is sufficient to ensure Lipschitz continuity and boundedness of the objective function's gradient).