# Regularized Saddle-Free Newton: A Globally Convergent and Efficient Non-Convex Newton Method

Cooper Simpson[1] and Stephen Becker[2]

[1]University of Colorado Boulder
[2]Applied Mathematics, CU Boulder
{*cooper.simpson, stephen.becker*}@*colorado.edu*

December 5, 2023

## Abstract

We present a new second-order method for unconstrained non-convex optimization, which we call Regularized Saddle-Free Newton (R-SFN). This work builds upon a number of recent ideas related to improving the theoretical and practical performance of the classic Newton's method. Our method applies to $C^2$ objectives with Lipschitz Hessian, and our analysis will require the existence of a third continuous derivative. In particular, we develop a nonlinear transformation to the Hessian which ensures it is positive definite at each iteration by approximating the regularized matrix absolute value. We show that with an appropriate random initialization and arbitrarily small additive noise our method avoids saddle points with probability one. Further, we prove convergence to first-order critical points when a particular form of regularization proportional to the norm of the gradient is used. Together, these results imply global convergence to second-order stationary points with probability one. In the convex case we prove a global $\mathcal{O}(1/k^2)$ convergence rate, and in general our method enjoys local super-linear convergence. The form of our nonlinear transformation facilitates an efficient matrix-free approach to computing the update via Krylov based quadrature, making our method scalable to high dimensional problems. We also consider a line-search procedure for when the Hessian Lipschitz constant is unknown. We thoroughly compare R-SFN against other second-order optimizers to show that ... Using two analytic non-convex benchmarks, we investigate the performance relative to increasing problem dimension, the size of the Krylov sub-space, and the quadrature order. Using the CUTEst benchmark, we ...

# 1 Introduction

We consider the following unconstrained optimization problem:

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

for a twice continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, where we make no assumptions on the convexity of the objective. To solve this problem we propose the following Newton-type update rule:

$$\boldsymbol{x}_{(k+1)} = \boldsymbol{x}_{(k)} - \eta_{(k)} \left( \left( \nabla^2 f(\boldsymbol{x}_{(k)}) \right)^2 + \lambda_{(k)} \mathbf{I} \right)^{-1/2} \left( \nabla f(\boldsymbol{x}_{(k)}) + \boldsymbol{\zeta}_{(k)} \right) \tag{2}$$

which we call Regularized Saddle-Free Newton (R-SFN). Equation (2) is presented quite generally, as for now we only require the step-size $\eta_{(k)}$ and the regularization $\lambda_{(k)}$ to be non-negative scalars, and $\boldsymbol{\zeta}_{(k)}$ additive random noise – all with a possible dependence on $\boldsymbol{x}_{(k)}$. We write eq. (2) with the noise added to the gradient, but we note that it could also be added to the update direction itself, i.e. outside of the linear system. It's presented form, however, is much more convenient to analyze. Our method draws inspiration from Saddle-Free Newton, which uses the absolute value of the Hessian, and recent theoretical results for regularized Newton in the convex case. Based on this, one might instead expect our update to use the following:

$$\left(\left|\nabla^2 f(\boldsymbol{x}_{(k)}\right| + \lambda_{(k)}\right)^{-1}$$

It is unclear if this would be preferable, but regardless, our approximation via the matrix square-root is smooth and admits an efficient implementation that is not available for the alternative above.

Under some further assumptions, we will show in Section 3 that eq. (2) avoids saddle points, and with a specific form of regularization Section 4 will show that it converges to a first-order stationary point. Together, these results imply global convergence to second-order stationary points with probability 1. Specializing to a convex objective we show that our method achieves global $\mathcal{O}(1/k^2)$ convergence in Section 4.1. A simple result from Section 4.2 gives super-linear convergence in any scenario. Importantly, in Section 5 we will also provide details on an efficient implementation of eq. (2) by computing the update via Krylov based quadrature. Detailed in Section 5.1 is a backtracking line-search procedure for when the Hessian Lipschitz constant is unknown. Section 6 presents an experimental investigation of our method's performance including comparisons against other Newton-type methods on the CUTEst non-linear optimization benchmark.

## 1.1 A Motivating Example

Consider the following two dimensional quadratic:

$$f(\boldsymbol{x}) = x_1^2 - x_2^2$$

This function is unbounded below, and so has no minima, but it does have a saddle point at $(0, 0)$. We will consider minimizing this objective using a noisy Newton's method of the following form:

$$\boldsymbol{x}_{(k+1)} = \boldsymbol{x}_{(k)} - \left(\nabla^2 f(\boldsymbol{x}_{(k)})\right)^{-1} \nabla f(\boldsymbol{x}_{(k)}) + \boldsymbol{\zeta}_{(k)} \tag{3}$$

The gradient and the Hessian of $f$ are easily computed, so we can write out the form of the update as follows[1]:

$$\boldsymbol{x}_{(k+1)} = \boldsymbol{x}_{(k)} - \begin{bmatrix} 1/2 & 0 \\ 0 & -1/2 \end{bmatrix} \begin{bmatrix} 2x_1^{(k)} \\ -2x_2^{(k)} \end{bmatrix} + \boldsymbol{\zeta}_{(k)} = \boldsymbol{x}_{(k)} - \boldsymbol{x}_{(k)} + \boldsymbol{\zeta}_{(k)} = \boldsymbol{\zeta}_{(k)}$$

If the noise were in fact zero, then we would see that Newton's method converges in a single step to the saddle point from any initialization. This is perhaps encouraging because we see extremely fast convergence, but discouraging because we would at least like to see behaviour that reduces the function value. For non-zero noise, subsequent iterations would have the same result, in that the updated iterate is simply the currently applied noise. While this does technically avoid convergence to the saddle point, it is hardly an improvement, as, for example, noise with norm $\epsilon$ would stay within $\epsilon$ of the saddle point. Now, let us make a minor modification to eq. (3) and take the absolute value of the Hessian. This yields the following result[1]:

$$\boldsymbol{x}_{(k+1)} = \boldsymbol{x}_{(k)} - \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} 2x_1^{(k)} \\ -2x_2^{(k)} \end{bmatrix} + \boldsymbol{\zeta}_{(k)} = \begin{bmatrix} 0 \\ 2x_2^{(k)} \end{bmatrix} + \boldsymbol{\zeta}_{(k)}$$

Considering first the case of zero noise, we see that the saddle point is avoided if the initial iterate is not chosen along the line $(x_1, 0)$, and we can observe that a fast exponential rate of functional decrease is preserved. In the non-zero noise case, the first coordinate of the updated iterate is simply the first coordinate of the applied noise, while the second coordinate is a perturbation of twice the previous coordinate. Assuming that the noise is not too large, and that it possibly decays to zero, we can see that the desired rate of functional decrease is maintained. As well, it is now impossible to converge to the saddle point even if the initial iterate is contained in the subspace $(x_1, 0)$, although once again, the result is not exactly better.

[1] We briefly change notation here by putting the iteration counter in the superscript, so that the subscript can be used to indicate coordinate.

While we are certainly not the first to observe this phenomenon, this simple example perfectly highlights the extremes of Newton's method, and seems to suggest that a simple modification could eliminate its issues. For this reason, the absolute value modification has been used as a heuristic for decades as discussed in the subsequent section. However, it also makes clear that the absolute value does not necessarily ensure saddle avoidance as the moniker given to this approach, saddle-free, would seem to suggest. Indeed, some type of probabilistic argument is needed, or the introduction of other mechanics to the iteration to avoid cases like an initialization of $(x_1, 0)$.

## 1.2 Related Work

A naive application of Newton's method may prove to be quite ineffective, as we have seen in Section 1.1. In particular, for non-convex objectives, the Hessian is no longer positive definite, so the Newton update is not a descent direction, which can result in convergence to saddle points or even maxima. Despite their apparent issues, second order optimization methods have long been a focus of research due to their potential for fast convergence. The local quadratic convergence of Newton's method is one of the canonical results in the field [Boyd and Vandenberghe, 2004], [Nocedal and Wright, 2006]. However, even in the convex case, it cannot be shown that Newton's method converges without a line-search and certain special assumptions about the objective function. Even equipped with this, Newton's method can still fail [Jarre and Toint, 2016]. A practical implementation of Newton's method is also not trivial for higher dimensional problems, as it requires solving a very large linear system.

Recent work of [Mishchenko, 2023] and [Doikov and Nesterov, 2023] showed that an appropriately regularized Newton's method will converge at a rate of $\mathcal{O}(1/k^2)$ for a convex objective function from any global initialization. In the same setting, [Hanzely et al., 2022] developed the first step-size schedule for damped Newton, obtaining the same global rate. Both methods also maintain local super-linear convergence. While their theoretical and empirical results are outstanding, these methods are not applicable to non-convex objectives and require the knowledge of often unknown constants. Other strategies to circumvent the convergence shortcomings of Newton's method and extend to the non-convex regime have also been investigated. One of the most notable variants is that of Cubic Newton [Nesterov and Polyak, 2006], which achieves fast global $\mathcal{O}(1/k^2)$ convergence and local super-linear convergence. While this method is effective and applicable in the non-convex regime, they are slow per iteration and employ a number of hyperparameters. Adaptive Regularization with Cubics (ARC) was introduced as an improvement and extension of cubic newton [Cartis et al., 2011], [Cartis et al., 2011]. The strong convergence characteristics are maintained, while an adaptive regularization term makes the algorithm much more practical.

It has been noted in the literature for a long time [Nocedal and Wright, 2006] that the issue of negative eigenvalues and convergence to non-minima may be mitigated by taking the absolute value of the Hessian. The work of [Dauphin et al., 2014] popularized this idea for deep learning, and called their method Saddle-Free Newton (SFN). They showed promising empirical results and gave some intuition as to why this approach may be valid, but there was still no solid theory backing it up. While the addition of the absolute value may seem like a convenient solution to descent issue, it only further complicates the implementation issue. In order to apply the absolute value to the Hessian, the standard approach has been to decompose the matrix first and then apply the absolute value to the eigenvalues. When it comes to high-dimensional optimization, this can completely prevent the practical use of these methods. The Low-Rank Saddle-Free Newton (LRSFN) method introduced in [O'Leary-Roseberry et al., 2020] is a simple variant of SFN that attempts to circumvent the implementation issues by using a low-rank approximation to the Hessian. They construct this approximation using a low-rank randomized eigenvalue decomposition of the dominant modes. Their analysis is focused on the stochastic (due to subsampling) setting, for which they consider linear stability and Levenberg-Marquardt type regularization. However, no convergence theory is provided, and saddle avoidance is only shown via limited numerical examples. A matrix-free technique is given by [Arjovsky, 2015], where they compute the absolute value as the square root of the squared matrix via a specific ODE. The approach of using the square root of the square to compute the matrix absolute value is the same as our own, but the method by which this is achieved (solving an ODE) is quite different. Beyond the algorithm itself, no theory or numerical experiments are considered. Continuing to employ the absolute value of the Hessian, [Paternain et al., 2019] introduced the Non-Convex Newton method. In addition to using the matrix absolute value, sufficiently small eigenvalues are replaced with a constant, and small amounts of noise are added in certain specific scenarios. This technique then allows them to show avoidance of saddle points and global convergence. The theory is strong, and yields a global non-convex convergence rate, but the method is difficult to implement and not necessarily fast. It's practical performance is also somewhat unclear due to limited numerical experiments. Perhaps the most similar to our work here is that of [Truong et al., 2023], who present New Q-Newton's method. This Newton variant regularizes the Hessian by a gradient term randomly scaled to ensure invertibility, and then applies the matrix absolute value. The authors attempt to show saddle avoidance using the Stable Manifold Theorem and local quadratic convergence, but a global convergence theory is not provided. Our work differs in a number of ways. First, our methods are fundamentally different despite having similar inspiration, and thus they require different analysis. We regularize by a different term, and compute the square root of a positive definite matrix. Second, our analysis is both more robust and more general. Third, our method admits an efficient implementation, whereas theirs suffers from the same issues as SFN as they require an eigenvalue decomposition.

When it comes to practical implementations of Newton-type methods, a key object is a matrix-free operator for applying the Hessian. This idea was... Even equipped with this, some methods still suffer due to their globalization strategy. Of particular note to this paper are the drawbacks of ARC, especially in high dimensions. Recently, [Dussault et al., 2023] proposed a variation called $\text{ARC}_\text{q}\text{K}$, which solves a set of shifted Newton-type linear systems at once using the Shifted CG-Lanczos Krylov solver. This incurs a small extra overhead, but allows them to avoid the costly re-computations normally associated with ARC.

Crucial to our analysis of saddle-avoidance are the works of [Lee et al., 2016] and [Panageas and Piliouras, 2017]. Together, these established very general almost sure avoidance of saddle points for gradient descent using the Stable Manifold Theorem (SMT)... Recent work on saddle avoidance of Newton-like dynamical systems through the (SMT) is presented in [Castera, 2023], but...

Broadly, one may consider a Newton-type update of the following form:

$$\boldsymbol{x}_{(k+1)} = \boldsymbol{x}_{(k)} - \eta_{(k)}\boldsymbol{B}_{(k)}^{-1}\nabla f(\boldsymbol{x}_{(k)}) \tag{4}$$

where $\boldsymbol{B}_{(k)}$ is a matrix that depends on the point $\boldsymbol{x}_{(k)}$. Table 1 outlines the various Newton method variants we have discussed so far. Our method can be seen as a combination of regularized Newton and saddle-free Newton. ...

| Method | $\boldsymbol{B}_{(k)}$ | $\eta_{(k)}$ | $\lambda_{(k)}$ | Global Convergence | Non-convex | Fast Impl. | Details |
|---|---|---|---|---|---|---|---|
| Newton | $\nabla^2 f(\boldsymbol{x}_{(k)})$ | N/A | N/A | ✗ | N/A | ✓ | N/A |
| Reg. Newton [8], [19] | $\nabla^2 f(\boldsymbol{x}_{(k)}) + \lambda_{(k)}\mathbf{I}$ | 1 | $\sqrt{M\\|\nabla f(\boldsymbol{x}_{(k)})\\|}$ | ✗ | ✗ | ✓ | N/A |
| AICN [13] | $\nabla^2 f(\boldsymbol{x}_{(k)})$ | $\frac{-1+\sqrt{1+2G}}{G}$ | 0 | ✓ | ✗ | ✓ | $G$ is a local smoothness constant |
| SFN [7] | $\left\lvert\nabla^2 f(\boldsymbol{x}_{(k)})\right\rvert$ | $(0,1]$ | 0 | ✗ | ✓ | ✗ | N/A |
| LRSFN [23] | $\left\lvert\nabla^2 f(\boldsymbol{x}_{(k)})\right\rvert_r + \lambda\mathbf{I}$ | $(0,1]$ | $(0,1]$ | ✗ | ✓ | ✓ | Rank-$r$ approximation |
| Cubic Newton [21, 4] | $\nabla^2 f(\boldsymbol{x}_{(k)})$ | 1 | $M\\|\boldsymbol{x}_{(k+1)} - \boldsymbol{x}_{(k)}\\|$ | ✓ | ✓ | ✓ | Requires solving complicated sub-problem |
| NCN [26] | $\left\lvert\nabla^2 f(\boldsymbol{x}_{(k)})\right\rvert_m$ | 1 | 0 | ✓ | ✓ | ✗ | Small eigenvalues replaced by $m$, requires complex perturbations |
| RSFN (Ours) | $\left(\left(\nabla^2 f(\boldsymbol{x}_{(k)})\right)^2 + \lambda_{(k)}\mathbf{I}\right)^{1/2}$ | 1 | $M\\|\nabla f(\boldsymbol{x}_{(k)})\\|$ | ✓ | ✓ | ✓ | N/A |

Table 1: Newton variants

## 2 Preliminaries

Throughout, we will use lowercase bold letters to denote vectors, and uppercase bold letters to denote matrices or operators. A parenthetic subscript will indicate an iteration count. Unless otherwise specified, the norm we employ is the 2-norm, denoted as $\|\cdot\|$, and its induced forms. For matrices, this results in the spectral norm, i.e. the largest singular value. We employ the notation $\boldsymbol{B} \preceq \boldsymbol{A}$, for symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, to indicate that $\boldsymbol{A} - \boldsymbol{B}$ is positive semi-definite. We will denote the Jacobian operator as $\boldsymbol{D}$, so that the following holds for a function $\phi : \mathbb{R}^n \to \mathbb{R}^m$:

$$\boldsymbol{D}\phi(\boldsymbol{x}) = \frac{\partial\phi(\boldsymbol{x})}{\partial\boldsymbol{x}^T} \qquad \& \qquad [\boldsymbol{D}\phi(\boldsymbol{x})]_{ij} = \frac{\partial\phi_i}{\partial\boldsymbol{x}_j} \tag{5}$$

for $i = 1\ldots, m$ and $j = 1, \ldots, n$ – given all such partial derivatives exist. We note that in order for $\phi$ to be considered differentiable at a point, the partial derivatives must also be continuous at that point, in which case the Jacobian is the derivative [Magnus and Neudecker, 2019]. To make things somewhat easier to parse we will also employ the following notation:

- $\boldsymbol{g} = \boldsymbol{g}(\boldsymbol{x}) = \nabla f(\boldsymbol{x}) = (\boldsymbol{D}f(\boldsymbol{x}))^T$
  with $\boldsymbol{g}_{(k)} = \boldsymbol{g}(\boldsymbol{x}_{(k)})$

- $\boldsymbol{H} = \boldsymbol{H}(\boldsymbol{x}) = \nabla^2 f(\boldsymbol{x}) = \boldsymbol{D}\left(\boldsymbol{D}f(\boldsymbol{x})\right)^T$
  with $\boldsymbol{H}_{(k)} = \boldsymbol{H}(\boldsymbol{x}_{(k)})$

Throughout, for our analysis, we will make the following assumption:

**Assumption 1: Differentiability**

The objective function $f$ has a continuous third derivative, i.e., $f \in C^3$

Other specific assumptions for the (anti-)convergence, which are mutually exclusive, will be stated in their respective sections. Because the Hessian is a real symmetric matrix, it is orthogonally diagonalizable, so we may write the following decomposition:

$$\boldsymbol{H} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{6}$$

where $\boldsymbol{V}$ is orthonormal, and $\boldsymbol{\Sigma}$ is a diagonal matrix consisting of the eigenvalues of $\boldsymbol{H}$. We will denote these eigenvalues as follows:

$$\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$$

Often, for an optimization problem of the form eq. (1), the goal is to show convergence to the following:

**Definition 1: Second-Order Stationary Point**

A point $\boldsymbol{x}_c \in \mathbb{R}^n$ is a second-order stationary point if $\boldsymbol{g}(\boldsymbol{x}_c) = \boldsymbol{0}$ and $\boldsymbol{0} \preceq \boldsymbol{H}(\boldsymbol{x}_c)$, i.e. $\boldsymbol{x}_c$ is a critical point where the Hessian is positive semi-definite.

A general saddle point is a critical point that is not a local minimum or maximum, and thus some saddle points are also second-order stationary points. To distinguish the two, we introduce the following:

**Definition 2: Strict Saddle Point**

A strict saddle point $\boldsymbol{x}_s$ is a critical point, i.e. $\boldsymbol{g}(\boldsymbol{x}_s) = \boldsymbol{0}$, where there is at least one direction of negative curvature, so the smallest eigenvalue of $\boldsymbol{H}(\boldsymbol{x}_s)$ is strictly less than 0.

If it holds that all saddle points are strict, then convergence to a second-order stationary point is the same as convergence to a local minimum.

# 3 Saddle Avoidance

The motivation behind using the absolute value of the Hessian is that it allows one to retain the "appropriate"[2] scaling of Newton's method, while preventing the possibility for convergence to saddle points. However, as we saw in Section 1.1 this isn't necessarily guaranteed, but in this section we will show that our method avoids saddle points given an arbitrarily small noisy perturbation. We consider our analysis to be a hybrid of the two main approaches used in the literature for showing saddle avoidance. That is th use of the Stable Manifold Theorem (Theorem 1) to make an almost-sure type argument, and the application of specific perturbations. Our method, in contrast, uses the Stable Manifold Theorem to argue that a very simple (and small) perturbation can be applied at every iteration. This approach alleviates some theoretical requirements for the analysis, but also avoids complex modifications of the algorithm to ensure the perturbations are applied correctly.

[2] Exactly what the appropriate scaling along directions of negative curvature is not well understood.

**Theorem 1: Stable Manifold [Shub, 1987]**

Let $\boldsymbol{x}_c$ be a fixed point for the $C^r$ local diffeomorphism $\phi : U \to \mathbb{R}^n$, where $r \geq 1$ and $U \subset \mathbb{R}^n$ is a neighborhood of $\boldsymbol{x}_c$. Let $E_s \oplus E_u$ be the invariant splitting of $\mathbb{R}^n$ into the subspaces corresponding to the eigenvalues of $D\phi(\boldsymbol{x}_c)$ less than or equal to 1, and greater than 1 respectively. Associated with $E_s$ is a local $\phi$ invariant $C^r$ embedded disc $W(\boldsymbol{x}_c) \subset E_s$, and ball $B$ around $\boldsymbol{x}_c$ such that the following hold:

$$\phi(W(\boldsymbol{x}_c)) \cap B \subset W(\boldsymbol{x}_c) \qquad \text{and} \qquad \phi^k(\boldsymbol{x}) \in B \; \forall k \geq 0 \implies \boldsymbol{x} \in W(\boldsymbol{x}_c)$$

In this context, the embedded disc $W(\boldsymbol{x}_c)$ is an open ball in the subspace $E_s$. The final result says that if a point $\boldsymbol{x}$ converges to the critical point $\boldsymbol{x}_c$ under the map $\phi$, then that point must have originated in $W(\boldsymbol{x}_c)$. This disc is referred to as the local stable center manifold of $\boldsymbol{x}_c$. Because it is contained in the subspace associated with the eigenvalues of $\boldsymbol{D}\phi(\boldsymbol{x}_c)$ that are less than or equal to 1, it has at most the same dimension as that subspace. This will be a key fact moving forward.

First, we will define the deterministic map associated with the R-SFN update rule (eq. (2)) as follows:

$$\Phi(\boldsymbol{x}) = \boldsymbol{x} - \eta \boldsymbol{A}(\boldsymbol{x})^{-1/2}\boldsymbol{g} = \boldsymbol{x} - \eta \left(\boldsymbol{H}^2(\boldsymbol{x}) + \lambda^2(\boldsymbol{x})\mathbf{I}\right)^{-1/2} \boldsymbol{g} \tag{7}$$

where we define $\boldsymbol{A}(\boldsymbol{x}) = \left(\boldsymbol{H}^2(\boldsymbol{x}) + \lambda^2(\boldsymbol{x})\mathbf{I}\right)$. This form of our method will be useful for the forthcoming analysis as we can view eq. (2) as $\Phi(\boldsymbol{x}) + \boldsymbol{\zeta}$. Note that the fixed points of $\Phi$ are exactly the critical points of $f$ when $\boldsymbol{A}$ is non-singular, which is ensured by part 1 of assumption 2. This assumption collects the conditions required for saddle-avoidance.

**Assumption 2: Saddle Avoidance**

1. The regularization $\lambda$ is a positive continuously differentiable function of $\boldsymbol{x}$
2. The step-size is a constant that satisfies $\eta \in (0, 1]$
3. The distribution of the noise $\boldsymbol{\zeta}$ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^n$

We require positivity of the regularization in order for Lemma 1 to hold, but as we can see this is a technical necessity, so we can take the scale to be arbitrarily small. The second part of assumption 2 is somewhat narrow, but is sufficient for the variants of eq. (2) that we consider here. One may consider different step-sizes that depend on $\boldsymbol{x}$, and under certain conditions it is likely that our saddle avoidance argument may still hold. To apply Theorem 1 it will be necessary to have an explicit form for the derivative $\boldsymbol{D}\Phi$, so we will derive this before proceeding any further. We refer the reader to [Magnus and Neudecker, 2019] for references on matrix calculus. The derivative of $\Phi$ is:

$$\boldsymbol{D}\Phi = \frac{\partial \Phi}{\partial \boldsymbol{x}^T} = \mathbf{I} - \eta \frac{\partial}{\partial \boldsymbol{x}^T}\left(\boldsymbol{A}^{-1/2}\boldsymbol{g}\right)$$

From Appendix A we have the following result:

$$\frac{\partial \boldsymbol{A}^{-1/2}\boldsymbol{g}}{\partial \boldsymbol{x}^T} = \left(\boldsymbol{g}^T \otimes \mathbf{I}\right) \frac{\partial\, \mathbf{vec}\left(\boldsymbol{A}^{-1/2}\right)}{\partial \boldsymbol{x}^T} + \left(1 \otimes \boldsymbol{A}^{-1/2}(\boldsymbol{x})\right) \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}^T}$$

It is easy to see that the second term is given by $\boldsymbol{A}^{-1/2}\boldsymbol{H}$, so we combine all of this together we get the following:

$$\boldsymbol{D}\Phi(\boldsymbol{x}) = \mathbf{I} - \eta\left(\boldsymbol{A}^{-1/2}\boldsymbol{H} + \left(\boldsymbol{g}^T \otimes \mathbf{I}\right)\boldsymbol{D}\,\mathbf{vec}\left(\boldsymbol{A}^{-1/2}\right)\right) \tag{8}$$

First, we will use eq. (8) to prove the following lemma. This will be key to using the stable manifold theorem in a helpful way.

**Lemma 1: Saddle Diffeomorphism**

For $\boldsymbol{x}_s$ a strict saddle point, eq. (7) is a local diffeomorphism in a neighborhood of $\boldsymbol{x}_s$ and $\boldsymbol{D}\Phi(\boldsymbol{x}_s)$ has at least one eigenvalue strictly larger than 1.

**Proof**

A saddle point is a critical point, so the gradient is zero, and thus eq. (8) reduces to the following:

$$\boldsymbol{D}\Phi(\boldsymbol{x}_s) = \mathbf{I} - \eta\left(\boldsymbol{H}^2 + \delta\mathbf{I}\right)^{-1/2}\boldsymbol{H}$$

where $\delta = \lambda^2(\boldsymbol{x}_s) > 0$. The three matrices involved are simultaneously diagonalizable, so we may write the following:

$$\boldsymbol{D}\Phi(\boldsymbol{x}_s) = \boldsymbol{V}\left(\mathbf{I} - \eta\left(\boldsymbol{\Sigma}^2 + \delta\mathbf{I}\right)^{-1/2}\boldsymbol{\Sigma}\right)\boldsymbol{V}^T$$

then the matrix above has eigenvalues of the following form for $i = 1, \dots, n$:

$$1 - \frac{\eta\mu_i}{\sqrt{\mu_i^2 + \delta}} > 0$$

By assumption, at least $\mu_n$ is negative, which implies the following:

$$1 - \frac{\eta\mu_n}{\sqrt{\mu_n^2 + \delta}} > 1$$

Thus the second results holds.

From the result above, we have that all the eigenvalues at $\boldsymbol{x}_s$ are positive, and so $\boldsymbol{D}\Phi$ is invertible there. Thus, via the Inverse Function Theorem [Spivak, 1965] the map $\Phi$ is a local diffeomorphism, and the first result holds. ▲

As mentioned earlier, $\delta$ was necessary to ensure invertibility of the derivative at saddle points, but it can be arbitrarily small and the result still holds. Now we move to the the main results of this section:

**Theorem 2: Saddle Avoidance**

Let the noise and regularization satisfy assumption 2, then any sequence generated by the R-SFN map eq. (2) avoids strict saddle points with probability 1 assuming the initial point $\boldsymbol{x}_{(0)}$ is chosen according to an absolutely continuous probability distribution.

**Proof**

Let $\boldsymbol{x}_s$ be a strict saddle point. Lemma 1 gives $\Phi : U \to \mathbb{R}^n$ as a $C^1$ local diffeomorphism for $U$ a neighborhood of $\boldsymbol{x}_s$, so applying Theorem 1 yields the $C^1$ manifold $W_s = W(\boldsymbol{x}_s)$. Also from Lemma 1 we know that there is at least one eigenvalue of $\boldsymbol{D}\Phi(\boldsymbol{x}_s)$ that is strictly larger then 1, so we conclude that $\dim(W_s) < n$. In other words, each stable center manifold has measure zero.

Lindelöf's lemma states that every open cover in $\mathbb{R}^n$ has a countable sub-cover[a] [Kelley, 1955]. Applying this, we can find a countable set of these manifolds such that the following holds:

$$W = \bigcup_{m=1}^{\infty} W_{s_m} = \bigcup_{s} W_s$$

i.e. a countable sub-cover for the union of all saddle point manifolds $W_s$. Importantly, $W$ is a countable union of measure zero sets, and thus it too is measure zero.

From here, we proceed by induction. Under the assumption on the probability distribution with which $\boldsymbol{x}_{(0)}$ was chosen we can conclude that $\boldsymbol{x}_{(0)} \notin W$ with probability 1. This follows because sets of measure zero have zero probability under absolutely continuous distributions. From here, assume that some $\boldsymbol{x}_{(k)} \notin W$ is given. Define $\tilde{\boldsymbol{x}}_{(k+1)} = \Phi(\boldsymbol{x}_{(k)})$. It is possible that this point is inside $W$, but after adding the noise $\boldsymbol{\zeta}_{(k)}$ we can conclude that it is not with probability 1. This follows because the probability of the noise lying in the subspace that contains $W$ is zero. Thus, we conclude overall that $\lim_{k\to\infty} \boldsymbol{x}_{(k)} \notin W$, so the result holds. ▲

[a] The result applies to any second-countable space, which includes Euclidean space.

Note that the proof above makes no assumption on the scale of the noise $\boldsymbol{\zeta}_{(k)}$, so we can make it arbitrarily small and the result still holds. The first portion of this proof, to generate the $W$ follows [Paternain et al., 2019] quite closely. The standard approach from here, using the Stable Manifold Theorem, would be to show that the map under question is injective, so the pre-image of sets of measure zero are measure zero. To avoid this criterion on eq. (7), we instead use a perturbation argument.

# 4 Convergence

Having established our non-convergence result, we now move on to convergence to stationary points. We will provide a general result for functional decrease, and then combine this with saddle avoidance to establish second-order convergence for a particular realization of eq. (2). Assumptions 3 and 4 are quite standard [Boyd and Vandenberghe, 2004].

**Assumption 3: Bounded Below**

The function $f$ is bounded below, i.e. there exists an $\boldsymbol{x}_*$ such that $f(\boldsymbol{x}_*) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}$

**Assumption 4: Lipschitz Hessian**

The Hessian, $\boldsymbol{H}$, is $M$-Lipschitz, i.e. the following holds for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$:

$$\|\boldsymbol{H}(\boldsymbol{x}) - \boldsymbol{H}(\boldsymbol{y})\| \leq M\|\boldsymbol{x} - \boldsymbol{y}\|$$

**Corollary 1**

If $f$ has an $M$-Lipschitz Hessian then the following inequality holds:

$$\|\nabla^3 f(\boldsymbol{x})\| \leq M$$

What the induced norm looks like for this third-order derivative is not important here, but for a reference one may examine [8]. Lastly, assumption 5 provides a set of criteria a method must satisfy to guarantee first-order stationary point convergence.

**Assumption 5: Global Convergence**

Define $M_{(k)} = \max_{t\in[0,1]} \|\nabla^3 f(t\boldsymbol{x}_{(k)} + (1-t)\boldsymbol{x}_{(k+1)})\|$

1. $\left(\boldsymbol{H}_{(k)}^2 + \lambda_{(k)}^2 \mathbf{I}\right)^{1/2} \preceq \boldsymbol{A}_{(k)}$
2. $\lambda_{(k)} \geq \sqrt{M_{(k)}\|\boldsymbol{g}_{(k)}\|}$
3. $\|\boldsymbol{\zeta}_{(k)}\| \leq \epsilon_{(k)}\|\boldsymbol{g}_{(k)}\|$ for some $\epsilon_{(k)} \in [0,1)$
4. $\|\boldsymbol{A}_{(k)}^{-1}\boldsymbol{g}_{(k)}\| \to 0 \implies \boldsymbol{g}_{(k)} \to \mathbf{0}$

The final part of assumption 5 may appear a bit strange, but is necessary to get from convergence of function values to convergence to a first-order stationary point. Ex. 1 provides an example of this holding for functions with Lipschitz gradients, but it may hold in other situations as well.

### Example 1

Suppose the gradient $\boldsymbol{g}$ is $L$-Lipschitz continuous, and define $\boldsymbol{A} = \left(\boldsymbol{H}^2 + \lambda^2\mathbf{I}\right)^{1/2}$. A Lipschitz gradient implies the following bound on the eigenvalues of the Hessian: $\|\boldsymbol{H}\| \leq L$. This implies $\|\boldsymbol{A}^{-1}\boldsymbol{g}\| \geq \frac{\|\boldsymbol{g}\|}{\sqrt{L^2+\lambda}}$, which ensures the relation $\|\boldsymbol{g}\| \to 0$ if $\|\boldsymbol{A}^{-1}\boldsymbol{g}\| \to 0$.

Now we present first main theorem of this section, which guarantees convergence to first-order stationary points.

### Theorem 3: Global Convergence

Consider the sequence $\boldsymbol{x}_{(k+1)} = \boldsymbol{x}_{(k)} - \boldsymbol{A}_{(k)}^{-1}\left(\boldsymbol{g}_{(k)} + \boldsymbol{\zeta}_{(k)}\right)$. If $\boldsymbol{A}_{(k)}$ satisfies assumption 5, then there exists an $\epsilon_{(k)}$ such that the image sequence converges to a first-order stationary point.

### Proof

Define $\boldsymbol{r}_{(k)} = \boldsymbol{x}_{(k+1)} - \boldsymbol{x}_{(k)}$. We begin with the following inequality, which is immediately implied by the Taylor remainder theorem applied to $f$ under assumption 4:

$$f_{(k+1)} - f_{(k)} \leq \left\langle \boldsymbol{g}_{(k)}, \boldsymbol{r}_{(k)} \right\rangle + \frac{1}{2}\left\langle \boldsymbol{H}_{(k)}\boldsymbol{r}_{(k)}, \boldsymbol{r}_{(k)} \right\rangle + \frac{M_{(k)}}{6}\|\boldsymbol{r}_{(k)}\|^3 \tag{9}$$

In the first term we will replace $\boldsymbol{g}_{(k)}$ with $-\boldsymbol{A}_{(k)}\boldsymbol{r}_{(k)} - \boldsymbol{\zeta}_{(k)}$. Then, using parts 2 and 3 of assumption 5, we can develop the following bound:

$$\|\boldsymbol{r}_{(k)}\| = \|\boldsymbol{A}_{(k)}^{-1}(\boldsymbol{g}_{(k)} - \boldsymbol{\zeta}_{(k)})\| \leq \frac{\|\boldsymbol{g}_{(k)}\| + \epsilon_{(k)}\|\boldsymbol{g}_{(k)}\|}{\lambda_{(k)}} \leq (1+\epsilon_{(k)})\frac{\lambda_{(k)}}{M_{(k)}} \tag{10}$$

which allows us to upper bound the last term of eq. (9) by $\frac{1+\epsilon_{(k)}}{6}\left\langle \lambda_{(k)}\boldsymbol{r}_{(k)}, \boldsymbol{r}_{(k)} \right\rangle$. Putting these two steps together gives the following:

$$f_{(k+1)} - f_{(k)} \leq \left\langle \left(\frac{1}{2}\boldsymbol{H}_{(k)} + \frac{1+\epsilon_{(k)}}{6}\lambda_{(k)}\mathbf{I} - \boldsymbol{A}_{(k)}\right)\boldsymbol{r}_{(k)}, \boldsymbol{r}_{(k)} \right\rangle - \left\langle \boldsymbol{\zeta}_{(k)}, \boldsymbol{r}_{(k)} \right\rangle \tag{11}$$

The matrices that appear in the first inner product are all simultaneously diagnonalizable. Using part 1 of assumption 5 we can look at the eigenvalues of this sum, and get the following upper bound:

$$\frac{1}{2}\boldsymbol{H}_{(k)} + \frac{1+\epsilon_{(k)}}{6}\lambda_{(k)}\mathbf{I} - \boldsymbol{A}_{(k)} \preceq \lambda_{(k)}\left(\underbrace{\frac{1-3\sqrt{3}}{6}}_{\rho} + \frac{\epsilon_{(k)}}{6}\right) = \lambda_{(k)}\left(\rho + \frac{\epsilon_{(k)}}{6}\right)$$

Plugging this into eq. (11) gives the inequality:

$$f_{(k+1)} - f_{(k)} \leq \lambda_{(k)}\left(\rho + \frac{\epsilon_{(k)}}{6}\right)\|\boldsymbol{r}_{(k)}\|^2 - \left\langle \boldsymbol{\zeta}_{(k)}, \boldsymbol{r}_{(k)} \right\rangle \tag{12}$$

It remains to be shown that we can choose an $\epsilon_{(k)}$ small enough to ensure the RHS is negative. To that end, the second term can be easily bounded through Cauchy-Schwarz using eq. (10) and part 3 of assumption 5. The first term we can bounded above by lower bounding the norm of the residual:

$$\|\boldsymbol{r}_{(k)}\|^2 \geq \left(1-\epsilon_{(k)}\right)^2\|\boldsymbol{A}_{(k)}^{-1}\boldsymbol{g}_{(k)}\|^2$$

which follows by the reverse triangle inequality on the energy-norm defined via the symmetric positive-definite matrix $\boldsymbol{A}_{(k)}^{-1}$. This gives the final result:

$$f_{(k+1)} - f_{(k)} \leq \lambda_{(k)}\left(\rho + \frac{\epsilon_{(k)}}{6}\right)\left(1-\epsilon_{(k)}\right)^2\|\boldsymbol{A}_{(k)}^{-1}\boldsymbol{g}_{(k)}\|^2 + \epsilon_{(k)}\left(1+\epsilon_{(k)}\right)\frac{\|\boldsymbol{g}_{(k)}\|\lambda_{(k)}}{M_{(k)}} \tag{13}$$

so choose $\epsilon_{(k)}$ to satisfy the following:

$$\epsilon_{(k)} \leq \frac{\rho \|\boldsymbol{A}_{(k)}^{-1} \boldsymbol{g}_{(k)}\|^2}{\frac{2\|\boldsymbol{g}_{(k)}\|}{M_{(k)}} + \frac{\|\boldsymbol{A}_{(k)}^{-1} \boldsymbol{g}_{(k)}\|}{6}}$$

This ensures that $f_{(k+1)} - f_{(k)}$ is negative, which implies that $\{f_{(k)}\}$ is a monotonically decreasing sequence, and assumption 3 ensures convergence. Therefore, we have $\lambda_{(k)} \|\boldsymbol{A}_{(k)}^{-1} \boldsymbol{g}_{(k)}\| \to 0$, so with part 4 of assumption 5, we have $\boldsymbol{g}_{(k)} \to 0$. ▲

We present Theorem 3 in a more general form than is immediately necessary, but we will see that it is useful for our consideration of line search procedures.

**Lemma 2: Regularization**

With $\lambda_{(k)} = \sqrt{M\|\boldsymbol{g}_{(k)}\| + \delta}$ for some $0 < \delta$, $\eta = 1$, and $\boldsymbol{\zeta}$ any absolutely continuous distribution satisfying the scale requirement of Theorem 3, then assumptions 2 and 5.

Lemma 2 is not entirely necessary for Theorem 4, as any regularization, step-size, and noise that satisfies the two assumptions will converge to a second-order stationary point. However, the parameters in Lemma 2 are what we will work with in particular.

**Theorem 4: Second-Order Convergence**

Equation (2) with regularization, step-size, and noise as defined in Lemma 2 converges to a second-order stationary point.

The proof is quite simple. Simply apply Theorem 3 to get convergence to a first-order stationary point, and then Theorem 2 ensures it is not a strict-saddle point. Thus, we conclude it must be a second-order stationary point.

## 4.1 Convex Convergence Rate

In lieu of a global non-convex convergence rate, we will provide one in the convex case. We first begin by proving the following lemma:

**Lemma 3**

Let $\boldsymbol{r}_{(k)} = \boldsymbol{x}_{(k+1)} - \boldsymbol{x}_{(k)}$, then with $\lambda_{(k)}$ as in Lemma 2, $\eta = 1$, and $\zeta = \mathbf{0}$ the following holds:

$$\|\boldsymbol{g}_{(k+1)}\| \leq \frac{3}{2}\lambda_{(k)}\|\boldsymbol{r}_{(k)}\| \qquad \& \qquad 2\lambda_{(k)}\|\boldsymbol{r}_{(k)}\| \leq 2\|\boldsymbol{g}_{(k)}\|$$

**Proof**

We start with the RHS inequality.

$$\|\boldsymbol{r}_{(k)}\| = \|\left(\boldsymbol{H}_{(k)}^2 + \lambda_{(k)}^2 \mathbf{I}\right)^{-1/2} \boldsymbol{g}_{(k)}\| \leq \frac{\|\boldsymbol{g}_{(k)}\|}{\lambda_{(k)}}$$

which follows simply and implies the second result. Define $A = \left(\boldsymbol{H}^2 + \lambda \mathbf{I}\right)^{1/2}$, and moving on to the LHS, we write the following via the Fundamental Theorem of Calculus:

$$\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{y}) - \boldsymbol{A}(\boldsymbol{y})(\boldsymbol{x} - \boldsymbol{y}) = \int_0^1 \left(\boldsymbol{H}(\boldsymbol{y} + t(\boldsymbol{x} - \boldsymbol{y})) - \boldsymbol{A}(\boldsymbol{y})\right)(\boldsymbol{x} - \boldsymbol{y})\, dt$$

Define $\boldsymbol{z} = \boldsymbol{y} + t(\boldsymbol{x} - \boldsymbol{y})$, and then take the norm of both sides and apply the triangle inequality to the integral:

$$\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{y}) - \boldsymbol{A}(\boldsymbol{y})(\boldsymbol{x} - \boldsymbol{y})\| \leq \int_0^1 \|\boldsymbol{H}(\boldsymbol{z}) - \boldsymbol{A}(\boldsymbol{y})\| \|\boldsymbol{x} - \boldsymbol{y}\| \, dt$$

Now, add and subtract $\boldsymbol{H}(\boldsymbol{y})$ and once again apply the triangle inequality to upper bound the first term in the integrand as follows:

$$\begin{aligned}\|\boldsymbol{H}(\boldsymbol{z}) - \boldsymbol{A}(\boldsymbol{y})\| &\leq \|\boldsymbol{H}(\boldsymbol{z}) - \boldsymbol{H}(\boldsymbol{y})\| + \|\boldsymbol{H}(\boldsymbol{y}) - \boldsymbol{A}(\boldsymbol{y})\| \\ &\leq M\|\boldsymbol{z} - \boldsymbol{y}\| + \lambda\end{aligned}$$

where the second inequality uses assumption 4. Finishing the integration:

$$\|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{y}) - \boldsymbol{A}(\boldsymbol{y})(\boldsymbol{x} - \boldsymbol{y})\| \leq \frac{1}{2}M\|\boldsymbol{x} - \boldsymbol{y}\|^2 + \lambda\|\boldsymbol{x} - \boldsymbol{y}\|\|$$

We take a brief interlude to show the following bound holds:

$$M\|\boldsymbol{r}_{(k)}\| \leq \frac{M\|\boldsymbol{g}_{(k)}\|}{\lambda_{(k)}} \leq \frac{M\|\boldsymbol{g}_{(k)}\| + \delta}{\lambda_{(k)}} = \lambda_{(k)}$$

which follows simply from the upper bound we have already shown. Now, using the triangle inequality and applying the preceding result to the integral yields the lower bound:

$$\|\boldsymbol{g}_{(k+1)}\| = \|\boldsymbol{g}_{(k+1)} - \boldsymbol{g}_{(k)} - \boldsymbol{A}_{(k)}\boldsymbol{r}_{(k)}\| \leq \frac{3}{2}\lambda_{(k)}\|\boldsymbol{r}_{(k)}\|$$

▲

**Assumption 6**

The diameter of the sub-level set $\{x : f(\boldsymbol{x}) \leq f(\boldsymbol{x}_{(0)}\}$ is bounded by some constant $D > 0$.

...

**Theorem 5: Convex Convergence Rate**

If $f$ is convex, then with $\lambda_{(k)} = \sqrt{M\|\boldsymbol{g}_{(k)}\| + \delta}$ for some $0 \leq \delta \leq 1$, $\eta = 1$, and $\boldsymbol{\zeta} = \boldsymbol{0}$, the iteration eq. (2) converges at a rate of $\mathcal{O}(1/k^2)$.

The proof of Theorem 5 is given in Appendix B as it can be carried out almost identically to that of Theorem 1 in [Mishchenko, 2023]. The only major difference comes from the $\delta$ term in our regularization. Also, Lemma 3 is slightly stronger than the second part of that work's Lemma 2, and the constant $\rho$ from Theorem 3 is slightly better than the constant $-2/3$ obtained in Lemma 3 of [Mishchenko, 2023]. This results in a better constant for the convergence rate, but the same asymptotic result. To get an idea of the quality of this rate, one may note that it is the same as Cubic Newton. Of course, that result holds even in the non-convex regime.

## 4.2 Local Super-linear Convergence

We now forgo the convexity assumption of the objective function from the prior section, yet we are still able to make a strong statement on the rate of local convergence. One should note the super-linear convergence is a general class with the following form:

$$\|\boldsymbol{g}_{(k+1)}\| \leq c\|\boldsymbol{g}_{(k)}\|^\alpha$$

where $c > 0$. The exponent $\alpha$ determines the exact rate. For example, Cubic Newton [Nesterov and Polyak, 2006] and the convex damped Newton of [Hanzely et al., 2022] have $\alpha = 2$, while the convex regularized Newton of

[Mishchenko, 2023] has $\alpha = 3/2$.

**Theorem 6: Local Super-Linear Convergence**

Suppose the second-order stationary point $\boldsymbol{x}_*$ that eq. (2) converges to with $\lambda_{(k)} = \sqrt{M\|\boldsymbol{g}_{(k)}\|}$, $\eta = 1$, and $\boldsymbol{\zeta} = \mathbf{0}$ is strict (in the sense that the $\mathbf{0} \prec \boldsymbol{H}(\boldsymbol{x}_*)$. Then for sufficiently small $\delta$, there exists a region around $\boldsymbol{x}_*$ where the sequence converges super-linearly. In particular the following inequality holds:

$$\|\boldsymbol{g}_{(k+1)}\| \leq \mathcal{O}(\|\boldsymbol{g}_{(k)}\|^{3/2}) + \frac{3\sqrt{\delta}}{2\mu_*}\|\boldsymbol{g}_{(k)}\|$$

where $\mu_*$ is a lower bound on the Hessian eigenvalues in the given region.

The proof is straightforward using Lemma 3, which does not require any convexity assumption itself, and it can be found in Appendix B. Much like our convex convergence rate, this result is quite similar to that of [Mishchenko, 2023], with some variation due the presence of $\delta$ in our regularization. We ...

# 5 Efficient Computation

A major issue with Newton type methods lies in their implementation, and can effectively prevent the practical usage of these methods. Take the standard Newton's method; to perform an update step one must solve a square linear system, and even with efficient solvers this can be quite costly, especially in high dimensions. Some improvements can be made beyond the most naive approach by using a matrix free linear solver, and by computing Hessian vector products using automatic differentiation (AD). For a saddle free Newton method, one in which we are using the absolute value of the Hessian (or an approximation), the problems are only exacerbated. Before one can even apply the update, one must now also perform an eigen-decomposition of the Hessian. Not only is this expensive, but it also requires forming an explicit matrix, which can be very memory intensive. We will see that the form of R-SFN naturally admits an efficient implementation, or at least similarly efficient relative to standard matrix-free Newton methods.

For a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ with spectrum $\sigma(\boldsymbol{A}) \subset \mathbb{R}_+$ we have the following integral representation of the matrix square root [Higham, 2008]:

$$\boldsymbol{A}^{1/2} = \frac{2}{\pi} A \int_0^\infty \left(t^2\mathbf{I} + \boldsymbol{A}\right)^{-1} \, dt$$

If we further enforce the eigenvalues of $\boldsymbol{A}$ to be strictly positive, i.e. $\sigma(\boldsymbol{A}) \subset \mathbb{R}_{++}$, then multiplying on the left by $\boldsymbol{A}^{-1}$ gives us the integral representation for the inverse square root:

$$\boldsymbol{A}^{-1/2} = \frac{2}{\pi} \int_0^\infty \left(t^2\mathbf{I} + \boldsymbol{A}\right)^{-1} \, dt \tag{14}$$

This is the fundamental equation behind our efficient implementation.

To start, we can avoid an eigen-decomposition by using the integral form for the matrix square root (eq. (14)). Applying this to the update portion of R-SFN (eq. (2)) we get the following:

$$\left(\boldsymbol{H}^2_{(k)} + \lambda^2_{(k)}\mathbf{I}\right)^{-1/2} \boldsymbol{g}_{(k)} = \frac{2\eta_{(k)}}{\pi} \int_0^\infty \left(\left(t^2 + \lambda^2_{(k)}\right)\mathbf{I} + \boldsymbol{H}^2_{(k)}\right)^{-1} \boldsymbol{g}_{(k)} \, dt \tag{15}$$

Note that we have taken the noise to be zero, as the theory allows for arbitrarily small scale, so in a practical implementation we simply ignore it. This integral can be approximated using an appropriate quadrature rule, so define the associated nodes as $t_i$ and weights as $w_i$, for $i = 1, \ldots, N$. This leaves us the following:

$$\left(\boldsymbol{H}^2_{(k)} + \lambda^2_{(k)}\mathbf{I}\right)^{-1/2} \boldsymbol{g}_{(k)} \approx \frac{2\eta_{(k)}}{\pi} \sum_{i=1}^{N} w_i \left(\left(t_i^2 + \lambda^2_{(k)}\right)\mathbf{I} + \boldsymbol{H}^2_{(k)}\right)^{-1} \boldsymbol{g}_{(k)} \tag{16}$$

It is important to note two necessary conditions the quadrature rule must satisfy. First, the rule should be applicable to the half-open domain $[0, \infty)$, which is an obvious consideration, but important nonetheless. Second, the weights $w_i$ must be positive, which is a vital requirement if the theoretical results considered in Section 3 have any hope of applying to this approximation. The entire motivation behind using the matrix absolute value, or the approximation in our case, is to ensure that the update in eq. (2) is a descent direction. For this to hold we need the matrix being applied to the gradient to be positive definite. The integrand in eq. (15) satisfies this, but we need positive weights for our approximation (eq. (16)) to as well.

In general, the summand of eq. (16) can be computed efficiently using a Krylov subspace method. In particular, taking inspiration from [Dussault et al., 2023], we propose using the shifted CG-Lanczos method. The core result that powers this algorithm is the fact that a Krylov subspace is shift invariant. For a linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, the associated order $r$ Krylov subspace is given as follows:

$$\mathcal{K}_r(\boldsymbol{A}, \boldsymbol{b}) = \text{span}\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \ldots, \boldsymbol{A}^{r-1}\boldsymbol{b}\}$$

It can then be shown that $\mathcal{K}_r(\boldsymbol{A}, \boldsymbol{b}) = \mathcal{K}_r(\boldsymbol{A} + \lambda\mathbf{I}, \boldsymbol{b})$ for arbitrary shifts $\lambda$, and although we will not state the full algorithm, this feature means one only need construct the Krylov subspace once [Frommer and Maass, 1999]. It can then be reused for the remaining systems, eliminating the major computational cost from those matrix vector products. With this approach, all $N$ vectors in eq. (16) can be computed together for a minimal cost beyond doing so for a single solve. Furthermore, with access to an operator for Hessian-vector products, this entire process can be done matrix free. Such an operator can be easily produced via AD in any modern scientific programming language.

Putting this all together, we present an efficient algorithm for computing the R-SFN update in the alg. 1. Note that this leaves out a number of important parameters, which are not central to the overall presentation. In the Kyrlov solver, we omit tolerances, run time limits, and iteration limits. We also leave the precise form of the regularization and how the quadrature is computed intentionally general. In practice, this algorithm can be optimized in a number of ways. For example, the quadrature need only be computed once, and then it can be reused for the remaining updates. The main computational cost comes from solving the family of linear systems, and in particular, from the associated Hessian-vector products used to form the Krylov subspace. To be more precise, the operation in question is actually applying the Hessian squared to a vector. Thus, there are $2(r-1)$ of these operations, where $r$ is the size of the Krylov subspace. When the operator is constructed using AD, each application of the Hessian requires two passes of the function using some combination of forward and reverse mode. We will not dive into the details of AD and its most efficient application here, but the result is a $\mathcal{O}(r)$ dependence of the algorithm on the size of the Krylov subspace. The cost of a single application of the Hessian operator is of course very dependent on the cost of evaluating the objective function in question.

**Algorithm 1:** R-SFN Update

```
Data: Objective function $f$, Current iterate $\boldsymbol{x}$, Quadrature order $N$, Regularization $\lambda$, Step-size $\eta$
$\boldsymbol{g} \leftarrow \nabla f(\boldsymbol{x})$
$\boldsymbol{H}^2 \leftarrow \left(\nabla^2 f(\boldsymbol{x})\right)^2$                // Could be a matrix-free operator
$\boldsymbol{t}, \boldsymbol{w} \leftarrow \text{Quadrature}(N)$                  // Compute quadrature nodes and weights
$s_i \leftarrow t_i^2 + \lambda^2$                               // Compute shifts, agnostic to form of $\lambda$
$\boldsymbol{Y} \leftarrow \text{ShiftedKrylov}(\boldsymbol{H}^2, \boldsymbol{g}, \boldsymbol{s})$          // Compute solutions to linear systems
for $i = 1 : N$ do
    $\boldsymbol{x} = \boldsymbol{x} - \eta w_i \boldsymbol{Y}_i$
end
```

## 5.1 Line Search

In practice, it is unlikely that one will have knowledge of the Hessian Lipschitz constant $M$ from assumption 4. One option is to apply a line-search procedure directly to the regularization, which is what is done by [Mishchenko, 2023]. Theoretically, one can even implement this quite efficiently by reusing the Krylov subspace that was generated for the previous shifts. However, it is easier to generate a single search direction, and then look for an appropriate step-size along that direction. Thus, we will consider a regularization term that drops any possibly unknown constants.

**Theorem 7**

With $\lambda_{(k)} = \sqrt{\|\boldsymbol{g}_{(k)}\|}$, $\boldsymbol{\zeta}_{(k)} = \mathbf{0}$, and $0 < r < \eta_{(k)} \leq \frac{1}{\sqrt{M_{(k)}}}$, then $\boldsymbol{A}_{(k)} = \frac{1}{\eta_{(k)}} \left(\boldsymbol{H}^2_{(k)} + \lambda^2_{(k)}\mathbf{I}\right)^{1/2}$ satisfies assumption 5.

Essentially, we are assuming that $M_{(k)} = 1$ and then accounting for the error in the step-size. However, if this assumption happened to hold, then we would recover a step-size of 1. It is important to note that step-sizes larger than are possible, and in fact necessary if we want our approximation to be as accurate as possible. It is easy to see that if the objective function satisfies assumption 4, then $r$ can be given by $1/\sqrt{M}$ as $M_{(k)} \leq M$ for all $k$. Under this same assumption, the line-search procedure given in alg. 2 is guaranteed to exit. The procedure is quite simple: the step-size is first increased from its previous value before possibly decreasing it to satisfy the exit condition. Note that with regularization only non-negative and $\boldsymbol{\zeta}_{(k)} = 0$ we do not satisfy assumption 2. In general, the introduction of a selected step-size complicates that analysis, and for a practical implementation we intend to set these parameters in the same way.

**Algorithm 2:** Step-Size Line-Search

```
Data: Objective function $f$, Current iterate $\boldsymbol{x}$, Search direction $\boldsymbol{p}$, Initial step-size $\eta_0$, Factor $\alpha \in (0, 1)$
$\eta \leftarrow \alpha^{-1}\eta_0$
$\lambda \leftarrow \sqrt{\|\nabla f(\boldsymbol{x})\|}$
while $f(\eta\boldsymbol{p}) - f(\boldsymbol{x}) > \rho\lambda\|\eta\boldsymbol{p} - x\|^2$                // Check descent condition
    $\eta \leftarrow \alpha\eta$
end
return $\eta\boldsymbol{p}$
```

# 6 Numerical Experiments

The following experiments were conducted in the Julia programming language, the code for which can be found in the GitHub repository [28]. The R-SFN algorithm itself is also implemented in Julia and publicly available at the GitHub repository [29]. In our implementation we use the Gauss-Laguerre quadrature rule as it is easily accessible, applicable to the non-negative real line, and, importantly, it has positive weights. Our shifted CG-Lanczos Krylov solver is provided by the Julia Smooth Optimizers organization from [Montoison et al., 2020]. In Section 6.1 we use a matrix free operator for Hessian-vector products generated via mixed forward over backward AD. In Section 6.2 we use the Hessian-vector operator provided via the Julia interface. Lastly, the regularization is computed according to the theory provided in Section 3, the form of which is given by Lemma 2. However, we will not apply any noise given that arbitrarily small perturbations still ensure saddle avoidance, and we are limited by machine precision to begin with. Following this same reasoning, we will take $\delta = 0$ in the regularization.

We will employ three main optimization methods: gradient descent, Newton's method, Adaptive Regularization with Cubics (ARC), and our own method R-SFN. For Newton's method we consider stepsizes of 1 and $1/2$, as well as a backtracking line search. For R-SFN, we consider $M = 0$ and $M = 1$ along with the line search procedure described in Section 5.1. Recall that $M = 0$ corresponds to the standard Saddle-Free Newton algorithm, so we label it SFN. Unless otherwise stated, we use ? quadrature nodes and $2d$ maximum Krylov subspace iterations, where $d$ is the dimension of the problem.

## 6.1 Rosenbrock and Michalewicz Benchmark

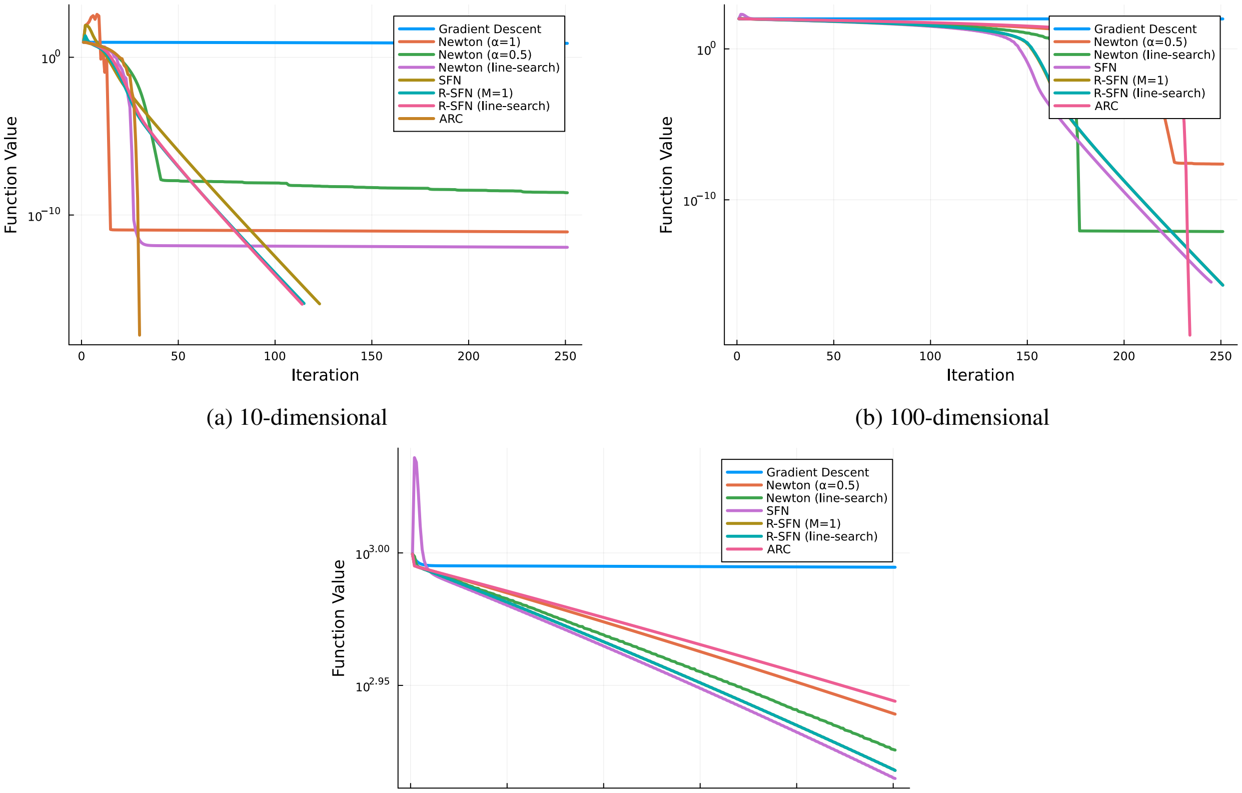
First, we consider the minimization of two standard non-convex optimization benchmarks, namely the Rosenbrock and Michalewicz functions. Both are able to scale to arbitrary dimension $d$, allowing us to investigate performance as a function of this measure. The $d$-dimensional Rosenbrock function is given as follows:

$$f(\boldsymbol{x}) = \sum_{i=1}^{d-1} 100\left(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i^2\right)^2 + (1 - \boldsymbol{x}_i)^2$$

and it has a global minimum of $\mathbf{0}$ which is achieved by the unique minimizer $\mathbf{1}$. The $d$-dimensional Michalewicz[3] function is given as:

$$f(\boldsymbol{x}) = -\sum_{i=1}^{d} \sin(\boldsymbol{x}_i)\sin^{20}\left(\frac{i\boldsymbol{x}_i^2}{\pi}\right)$$

which has $d!$ local minimia. For the Rosenbrock function we start at the initial point $\boldsymbol{x}_{(0)} = \mathbf{0}$, which is the standard initialization. The Michalewicz has no standard starting point, so instead we randomly sample $\boldsymbol{x}_{(0)} \in [0, \pi]^d$, and fix this for all methods. Each method is allowed unlimited runtime, but fix the maximum number of iterations at 100. We begin, in Figs. 1 and 2, by simply comparing the convergence characteristics of our various solvers. ... ...



(a) 10-dimensional

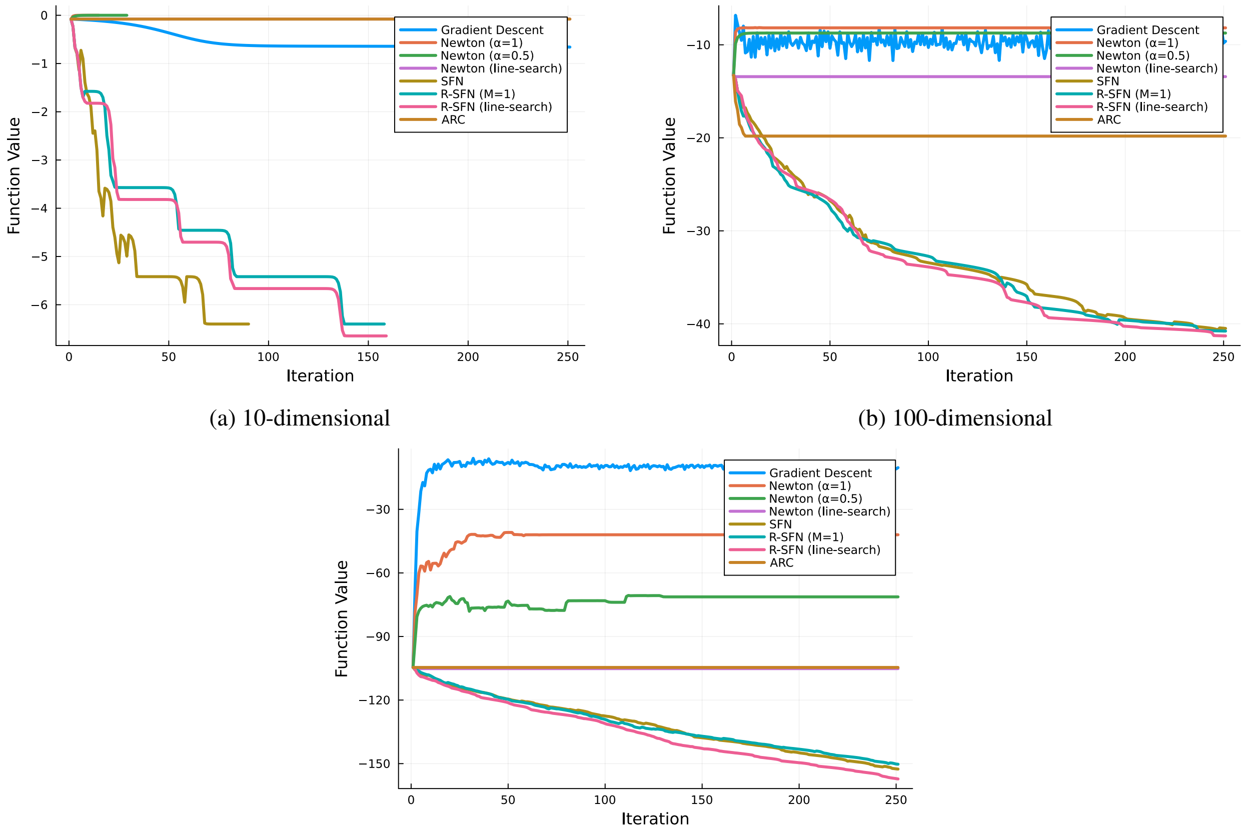
(b) 100-dimensional

(c) 1000-dimensional

Figure 1: Minimization of Rosenbrock function.

The two major hyper-parameters present in our algorithm are the size of the Kyrlov subspace and the number of quadrature nodes used in the finite sum approximation to the integral. As is perhaps obvious, larger values for either of these quantities incurs more substantial computational cost. In Figs. 3 and 4 we investigate R-SFN's behaviour with respect to these quantities for both the Rosenbrock and Michalewicz test functions. ... ...

[3] Technically the power of the sin can be $2m$ for integer $m$, but we use the standard value of $m = 10$.



(a) 10-dimensional

(b) 100-dimensional

(c) 1000-dimensional

Figure 2: Minimization of Michalewicz function.

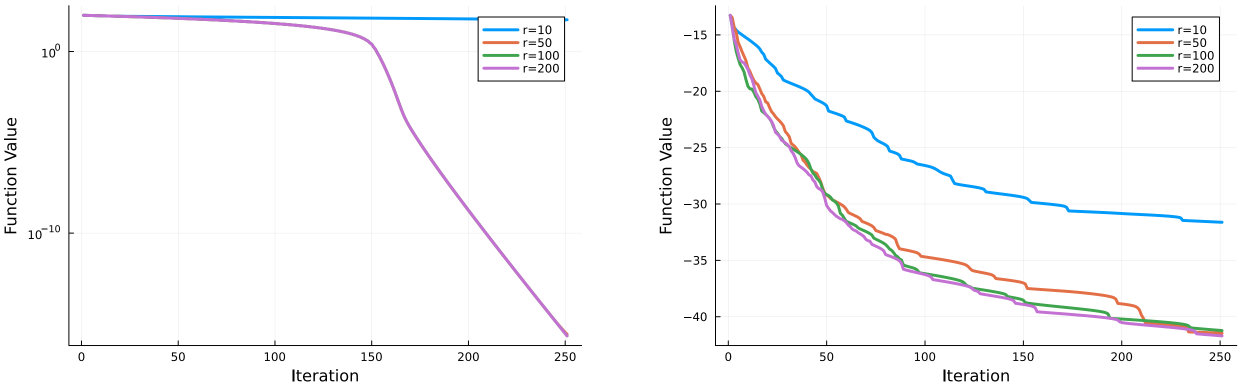
### 6.2 CUTEst

The CUTEst optimization benchmark is a large collection of non-linear optimization problems ?. We interface with these problems using the Julia package CUTEst.jl [Orban et al., 2020]. Only the unconstrained problems are selected, leaving us with 286, varying in dimension from 2 to 121,000. Our results in Fig. 5 are reported in the form of Performance Profiles ?, which... We consider three different cost functions: run-time, iteration count, and Hessian-vector products. ...

## 7 Discussion

In this work we have presented a new second-order optimization scheme which we call regularized saddle-free Newton (R-SFN). Under mild assumptions, we showed that this method almost surely converges to second-order stationary points via almost sure saddle-avoidance and first-order critical point convergence. In addition, we provided a global $\mathcal{O}(1/k^2)$ convergence rate in the convex case, and a local super-linear rate in any case. We also presented an efficient implementation of the method which allows for fast matrix-free updates and doesn't require knowledge of the global Hessian Lipschitz constant. A variety of non-convex numerical benchmarks validated our method as ... Much recent work has been carried out towards developing Newton type methods that avoid saddle points, yield strong global convergence, and can be easily applied in practice. We view R-SFN as a significant step towards combining all of these features. Yet, perhaps more than anything, the work presented here points towards many promising directions for further investigation.

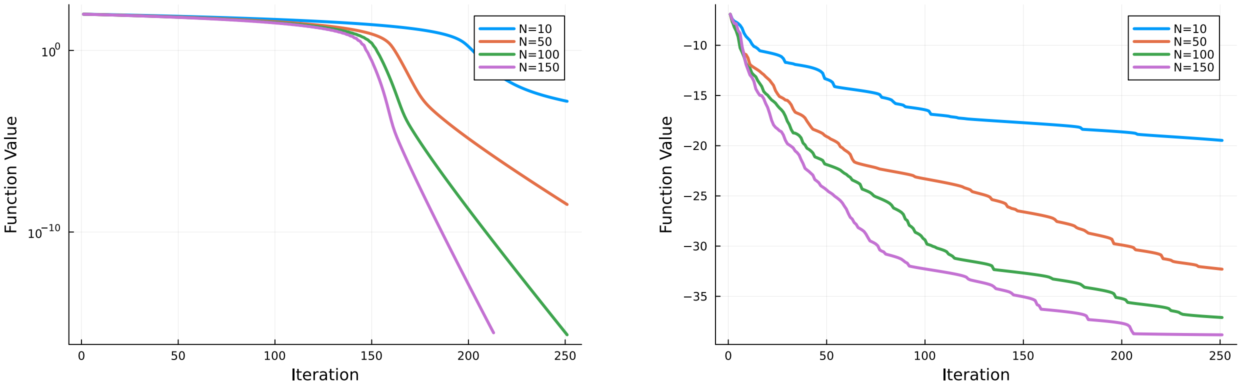
The most glaring omission from our analysis is that of a global non-convex convergence rate... Further, our super-linear convergence result is somewhat vague, in that it doesn't specify exactly what that rate looks like, so a more specific rate would be desirable. Although our practical implementation matches quite closely with the theoretical



(a) Rosenbrock

(b) Michalewicz

Figure 3: Effects of Krylov sub-space size on R-SFN.



(a) Rosenbrock

(b) Michalewicz

Figure 4: Effects of quadrature order on R-SFN.

analysis, there are two main sources of error that are not accounted for: the quadrature approximation and the integrand error introduced by the Kyrlov solver. This means that in reality, our update direction does not quite match the theory. Thus, an analysis that accounts for these sources of error is warranted. Related to this is the question of what is the best quadrature rule to use?

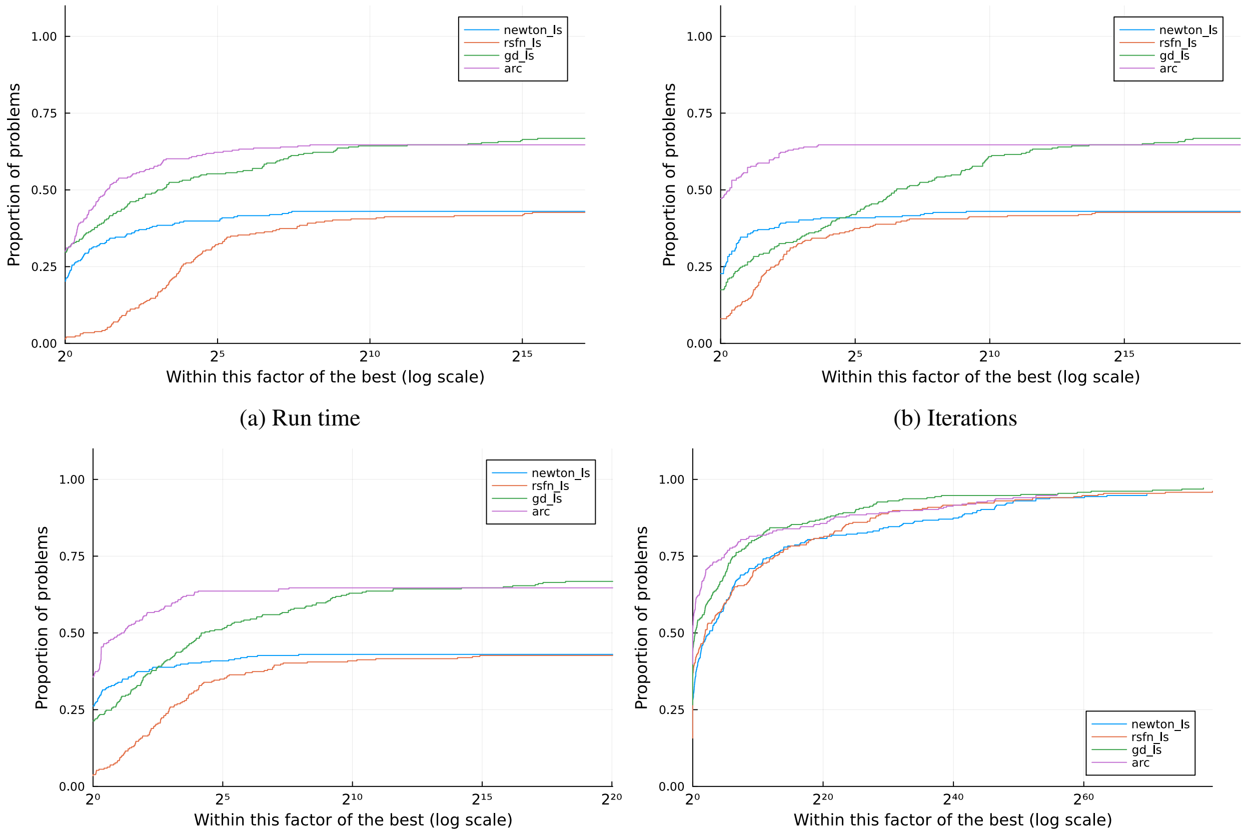
Beyond these immediate directions of research are concepts we would like to integrate for improved performance, but which further complicate the analysis. First and foremost are stochastic gradients and Hessians due to mini-batching in machine learning settings. Some contemporary research seeks to improve the efficiency of computing the update direction via low-rank [Hanzely, 2023] and randomized methods [Frangella et al., 2023]. The latter also implements infrequent updates of the Hessian, which is something we would like to apply to our method via Krylov sub-space recycling [Burke et al., 2022].

# References

[1] Martin Arjovsky. “Saddle-free Hessian-free optimization”. In: *arXiv* (2015). DOI: `10.48550/arXiv.1506.00059`.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[3] Liam Burke et al. “Krylov Subspace Recycling For Matrix Functions”. In: *arXiv* (2022). DOI: `10.48550/arXiv.2209.14163`.



(a) Run time

(b) Iterations

(c) Function evaluations + Hessian-vector products

(d) Relative distance to best minimum

Figure 5: CUTEst performance profiles.

[4] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. “Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results”. In: *Mathematical Programming* 127.2 (2011), pp. 245–295.

[5] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. “Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function-and derivative-evaluation complexity”. In: *Mathematical programming* 130.2 (2011), pp. 295–319.

[6] Camille Castera. “Inertial Newton algorithms avoiding strict saddle points”. In: *Journal of Optimization Theory and Applications* (2023), pp. 1–23.

[7] Yann N Dauphin et al. “Identifying and attacking the saddle point problem in high-dimensional non-convex optimization”. In: *Advances in neural information processing systems* 27 (2014).

[8] Nikita Doikov and Yurii Nesterov. “Gradient regularization of Newton method with Bregman distances”. In: *Mathematical Programming* (2023), pp. 1–25.

[9] Jean-Pierre Dussault, Tangi Migot, and Dominique Orban. “Scalable adaptive cubic regularization methods”. In: *Mathematical Programming* (2023), pp. 1–35.

[10] Zachary Frangella et al. “SketchySGD: Reliable Stochastic Optimization via Randomized Curvature Estimates”. In: *arXiv* (2023). DOI: `10.48550/arXiv.2211.08597`.

[11] Andreas Frommer and Peter Maass. “Fast CG-based methods for Tikhonov–Phillips regularization”. In: *SIAM Journal on Scientific Computing* 20.5 (1999), pp. 1831–1850.

[12] Slavomír Hanzely. “Sketch-and-Project Meets Newton Method: Global $\mathcal{O}(k^{-2})$ Convergence with Low-Rank Updates”. In: *arXiv* (2023). DOI: `10.48550/arXiv.2305.13082`.

[13] Slavomír Hanzely et al. “A Damped Newton Method Achieves Global $\mathcal{O}(1/k^2)$ and Local Quadratic Convergence Rate”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 25320–25334.

[14] Nicholas J Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

[15] Florian Jarre and Philippe L Toint. “Simple examples for the failure of Newton’s method with line search for strictly convex minimization”. In: *Mathematical Programming* 158.1-2 (2016), pp. 23–34.

[16] John L Kelley. *General Topology*. Springer, 1955.

[17] Jason D Lee et al. “Gradient descent only converges to minimizers”. In: *Conference on learning theory*. PMLR. 2016, pp. 1246–1257.

[18] Jan R Magnus and Heinz Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons, 2019.

[19] Konstantin Mishchenko. “Regularized Newton Method with Global $\mathcal{O}(1/k^2)$ Convergence”. In: *SIAM Journal on Optimization* 33.3 (2023), pp. 1440–1462.

[20] A. Montoison, D. Orban, and contributors. *Krylov.jl: A Julia Basket of Hand-Picked Krylov Methods*. https://github.com/JuliaSmoothOptimizers/Krylov.jl. June 2020.

[21] Yurii Nesterov and B.T. Polyak. “Cubic Regularization of Newton Method and its Globabl Performance”. In: *Mathematical Programming* 108.1 (2006), pp. 177–205. DOI: 10.1007/s10107-006-0706-8.

[22] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2e. Springer, 2006.

[23] Thomas O’Leary-Roseberry, Nick Alger, and Omar Ghattas. “Low rank saddle free Newton: A scalable method for stochastic nonconvex optimization”. In: *arXiv* (2020). DOI: 10.48550/arXiv.2002.02881.

[24] D. Orban, A. S. Siqueira, and contributors. *CUTEst.jl: Julia’s CUTEst interface*. https://github.com/JuliaSmoothOptimizers/CUTEst.jl. Oct. 2020.

[25] Ioannis Panageas and Georgios Piliouras. “Gradient Descent Only Converges to Minimizers: Non-Isolated Critical Points and Invariant Regions”. In: *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2017.

[26] Santiago Paternain, Aryan Mokhtari, and Alejandro Ribeiro. “A Newton-based method for nonconvex optimization with fast evasion of saddle points”. In: *SIAM Journal on Optimization* 29.1 (2019), pp. 343–368.

[27] Michael Shub. *Global Stability of Dynamical Systems*. Springer, 1987.

[28] Cooper Simpson. *R-SFN*. https://github.com/RS-Coop/R-SFN. Numerical experiments in Julia. 2023.

[29] Cooper Simpson. *SFN.jl*. https://github.com/rs-coop/SFN.jl. A Julia Implementation of Regularized Saddle-Free Newton. 2022.

[30] Michael Spivak. *Calculus On Manifolds*. Addison-Wesley, 1965.

[31] Tuyen Trung Truong et al. “A Fast and Simple Modification of Newton’s Method Avoiding Saddle Points”. In: *Journal of Optimization Theory and Applications* (2023), pp. 1–26.

# Appendices

## Appendix A

We will show the following result in exercise 9.14.6 of [Magnus and Neudecker, 2019] holds for $\boldsymbol{A}(\boldsymbol{x}) \in \mathbb{R}^{m\times p}$, $\boldsymbol{B}(\boldsymbol{x}) \in \mathbb{R}^{p\times r}$, and $\boldsymbol{x} \in \mathbb{R}^n$:

$$\frac{\partial}{\partial \boldsymbol{x}^T}\left[\boldsymbol{A}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\right] = \left(\boldsymbol{B}(\boldsymbol{x})^T \otimes \mathbf{I}_m\right)\frac{\partial\,\mathbf{vec}\left(\boldsymbol{A}(\boldsymbol{x})\right)}{\partial \boldsymbol{x}^T} + \left(\mathbf{I}_r \otimes \boldsymbol{A}(\boldsymbol{x})\right)\frac{\partial\,\mathbf{vec}\left(\boldsymbol{B}(\boldsymbol{x})\right)}{\partial \boldsymbol{x}^T}$$

where the subscript on the identity indicates its size.

**Proof**

We begin by computing the differential of $\boldsymbol{A}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})$ using the product rule:

$$\partial\left(\boldsymbol{A}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\right) = \partial\left(\boldsymbol{A}(\boldsymbol{x})\right)\boldsymbol{B}(\boldsymbol{x}) + \boldsymbol{A}(\boldsymbol{x})\partial\left(\boldsymbol{B}(\boldsymbol{x})\right)$$

Then apply the **vec** operator to get the following result:

$$\partial\,\mathbf{vec}\left(\boldsymbol{A}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\right) = \left(\boldsymbol{B}(\boldsymbol{x})^T \otimes \mathbf{I}_m\right)\partial\,\mathbf{vec}\left(\boldsymbol{A}(\boldsymbol{x})\right) + \left(\mathbf{I}_r \otimes \boldsymbol{A}(\boldsymbol{x})\right)\partial\,\mathbf{vec}\left(\boldsymbol{B}(\boldsymbol{x})\right)$$

From here we simply read off the result using the First Identification Theorem. ▲

Now, we will need to be able to write out the general derivative of $\boldsymbol{A}^{-1/2}$, where $\boldsymbol{A}$ is defined as in Section 3. We will apply the chain rule, but first, we must determine $\boldsymbol{D}\boldsymbol{A}^{-1}$ and $\boldsymbol{D}\boldsymbol{A}^{1/2}$. The former is a well-known result, available in [18], and is given as $\boldsymbol{D}\boldsymbol{A}^{-1} = -\left(\boldsymbol{A}^{-1} \otimes \boldsymbol{A}^{-1}\right)\boldsymbol{D}\boldsymbol{A}$. The second requires more work, but we begin with the following:

$$\boldsymbol{A} = \boldsymbol{A}^{1/2}\boldsymbol{A}^{1/2}$$

Then, we compute the differential using the product rule:

$$\partial\boldsymbol{A} = \boldsymbol{A}^{1/2}\partial\left(\boldsymbol{A}^{1/2}\right) + \partial\left(\boldsymbol{A}^{1/2}\right)\boldsymbol{A}^{1/2}$$

We can identify this in the form of a Sylvester equation, $\boldsymbol{P}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{Q} = \boldsymbol{R}$, where for our purposes we will assume $\boldsymbol{P}$ and $\boldsymbol{Q}$ are the same size and symmetric. A solution can be developed in the form of a linear system using the **vec** operator: $\mathbf{vec}\left(\boldsymbol{X}\right) = \left(\mathbf{I} \otimes \boldsymbol{P} + \boldsymbol{Q} \otimes \mathbf{I}\right)^{-1}\mathbf{vec}(\boldsymbol{C})$. Applying this to our specific problem yields the following:

$$\partial\,\mathbf{vec}\left(\boldsymbol{A}^{1/2}\right) = \left(\mathbf{I} \otimes \boldsymbol{A}^{1/2} + \boldsymbol{A}^{1/2} \otimes \mathbf{I}\right)^{-1}\partial\,\mathbf{vec}\left(\boldsymbol{A}\right)$$

where we have also swapped the order of the **vec** and differential operations. From here, we can read of the solution as $\boldsymbol{D}\boldsymbol{A}^{1/2} = \left(\boldsymbol{A}^{1/2} \oplus \boldsymbol{A}^{1/2}\right)^{-1}\boldsymbol{D}\boldsymbol{A}$. Applying the chain rule gives us the following:

$$\boldsymbol{D}\boldsymbol{A}^{-1/2} = -\left(\boldsymbol{A} \otimes \boldsymbol{A}^{1/2} + \boldsymbol{A}^{1/2} \otimes \boldsymbol{A}\right)^{-1}\boldsymbol{D}\boldsymbol{A}$$

so all that remains is to find $\boldsymbol{D}\boldsymbol{A}$. This is quite simple, and for a general $\lambda$ we get the following result:

$$\boldsymbol{D}\boldsymbol{A} = \boldsymbol{D}\lambda^2\mathbf{I} + \boldsymbol{D}\boldsymbol{H}^2 = \boldsymbol{D}\lambda^2\mathbf{I} + \left(\boldsymbol{H} \otimes \boldsymbol{H}\right)\boldsymbol{D}\boldsymbol{H}$$

The second term is a result of the power rule, again available in [18]. Putting this all together we get the final result below:

$$\frac{\partial\,\mathbf{vec}\left(\boldsymbol{A}^{-1/2}\right)}{\partial\boldsymbol{x}^T} = \boldsymbol{D}\boldsymbol{A}^{-1/2} = -\left(\boldsymbol{A} \otimes \boldsymbol{A}^{1/2} + \boldsymbol{A}^{1/2} \otimes \boldsymbol{A}\right)^{-1}\left(\frac{\partial\,\mathbf{vec}\left(\lambda^2\mathbf{I}\right)}{\partial\boldsymbol{x}^T} + \left(\boldsymbol{H} \otimes \boldsymbol{H}\right)\boldsymbol{K}\right) \tag{17}$$

where $\boldsymbol{K} = \frac{\partial\,\mathbf{vec}(\boldsymbol{H})}{\partial\boldsymbol{x}^T}$ is the $n^2 \times n$ matrix of third-order partial derivatives.

# Appendix B

**Proof**

*Proof of Lemma 2:*

We start with assumption 2. The second and third parts are already satisfied, which leaves us to show that the regularization is a positive continuously differentiable function of $\boldsymbol{x}$. Positivity is trivially satisfied with $\delta > 0$, and then we consider the derivative as follows:

$$\frac{\partial\, \mathbf{vec}\left(\lambda^2 \mathbf{I}\right)}{\partial \boldsymbol{x}^T} = \begin{bmatrix} c\boldsymbol{g}^T\boldsymbol{H} & 0 & \cdots & 0 & c\boldsymbol{g}^T\boldsymbol{H} & 0 & \cdots & 0 & c\boldsymbol{g}^T\boldsymbol{H} \end{bmatrix}^T$$

where $c = M\|\boldsymbol{g}\|^{-1}$.

Now we move to assumption 5, which holds trivially. Part 1 is satisfied via equality, part 2 is satisfied by definition of $\lambda_{(k)}$, and parts 3 and 4 we assume are already satisfied. ▲

**Proof**

*Proof of Theorem 5:*

We start with the following:

$$\frac{1}{4}\|\boldsymbol{g}_{(k)}\| \leq \|\boldsymbol{g}_{(k+1)}\| \leq \frac{3}{2}\lambda_{(k)}\|\boldsymbol{r}_{(k)}\|$$

where the first inequality follows by assumption and the second from Lemma 3. Rearranging, this lets us write $\|\boldsymbol{r}_{(k)}\| \geq \frac{\|\boldsymbol{g}_{(k)}\|}{6\lambda_{(k)}}$. Now we plug this into the descent result eq. (13) of Theorem 3 to write the following:

$$f_{(k+1)} - f_{(k)} \leq \rho\lambda_{(k)}\|\boldsymbol{r}_{(k)}\|^2 \leq \rho\frac{\|\boldsymbol{g}_{(k)}\|^2}{36\lambda_{(k)}}$$

We can upper bound the regularization by the max of two terms that depend on the size of the gradient norm relative to $\delta$.

$$\lambda_{(k)} \leq \left(\sqrt{M+1}\right)\max\{\|\boldsymbol{g}_{(k)}\|^{1/2}, \delta^{1/2}\}$$

By assumption, however, we have that $\delta \leq 1$, which yields the overall bound (recalling that $\rho$ is negative):

$$f_{(k+1)} - f_{(k)} \leq \frac{\rho}{36\sqrt{M+1}}\|\boldsymbol{g}_{(k)}\|^{3/2}$$

Applying assumption 6 along with convexity gives the following final bound:

$$f_{(k+1)} - f_{(k)} \leq \frac{\rho}{36\sqrt{M+1}D^{3/2}}\left(f_{(k)} - f_*\right)^{3/2}$$

where we note that the constant is negative. ▲

**Proof**

*Proof of Theorem 6.*

By assumption, there exists a $\mu_* > 0$ such that $\mu_*\mathbf{I} \prec \boldsymbol{H}(\boldsymbol{x}_*)$, which provides the following bound on the residual:

$$\|\boldsymbol{r}_{(k)}\| = \|\left(\boldsymbol{H}_{(k)}^2 + \lambda_{(k)}^2\mathbf{I}\right)^{-1/2}\boldsymbol{g}_{(k)}\| \leq \frac{1}{\mu_*}\|\boldsymbol{g}_{(k)}\|$$

for $k \geq k_0$. Substituting this into the lower bound from Lemma 3 gives the following:

$$\|\boldsymbol{g}_{(k+1)}\| \leq \frac{3}{2\mu_*}\lambda_{(k)}\|\boldsymbol{g}_{(k)}\| = \frac{3\sqrt{M}}{2\mu_*}\|\boldsymbol{g}_{(k)}\|^{3/2} + \frac{3\sqrt{\delta}}{2\mu_*}\|\boldsymbol{g}_{(k)}\|$$

where the final inequality follows by substituting the form of the regularization and rearranging. From here, we can see that for $\delta$ sufficiently small... ▲

# Appendix C

**Proof**

*Proof of Theorem 7:*

The noise is zero which trivially satisfies part 3, and we assume part 4 is already satisfied. If we can show the spectral inequality of part 1 holds, then part 2 follows via Lemma 2. To that end, consider the following:

$$\left(\boldsymbol{H}_{(k)}^2 + M_{(k)}\|\boldsymbol{g}_{(k)}\|\mathbf{I}\right)^{1/2} \preceq \frac{1}{\eta_{(k)}}\left(\boldsymbol{H}_{(k)}^2 + \|\boldsymbol{g}_{(k)}\|\mathbf{I}\right)^{1/2} \preceq \sqrt{M_{(k)}}\left(\boldsymbol{H}_{(k)}^2 + \|\boldsymbol{g}_{(k)}\|\mathbf{I}\right)^{1/2}$$

We consider eigenvalues $\mu \in \sigma(\boldsymbol{H}_{(k)})$ and need to show the following inequality holds:

$$\frac{1}{M_{(k)}} \leq \frac{\mu^2 + \|\boldsymbol{g}\|}{\mu^2 + M\|\boldsymbol{g}\|}$$

Which can be validated by looking at the critical point $\mu = 0$ and also the limit $\mu \to \infty$ of the RHS. ▲