Regularized Saddle-Free Newton: Saddle Avoidance and Efficient Implementation

by

Cooper R. Simpson

B.S., University of Colorado Boulder, 2019

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Masters of Science

Department of Applied Mathematics

2022

Committee Members:

Stephen Becker, Chair

Rafael Frongillo

Emilliano Dall'Anese

Simpson, Cooper R. (M.S., Applied Mathematics)

Regularized Saddle-Free Newton: Saddle Avoidance and Efficient Implementation

Thesis directed by Prof. Stephen Becker

We present a new second-order method for unconstrained non-convex optimization, which we dub Regularized Saddle-Free Newton (R-SFN). This work builds upon a number of recent ideas related to improving the practical performance of the classic Newton's method. In particular, we develop a nonlinear transformation to the Hessian which ensures it is positive definite at each iteration by approximating the matrix absolute value and regularizing with a scaled gradient norm. While our method applies to C^2 objectives with Lipschitz Hessian, our analysis will require the existence of a third continuous derivative. Given this, we show that with an appropriately random initialization our method avoids saddle points almost surely. Furthermore, 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.

Contents

Tables

Table

Figures

Figure

Chapter 1

Introduction

We consider the following unconstrained optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \tag{1.1}$$

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

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha\Big(\big(\nabla^2 f(\mathbf{x}^{(k)})\big)^2 + \lambda^{(k)}\mathbf{I}\Big)^{-1/2}\nabla f(\mathbf{x}^{(k)}) \tag{1.2}$$

which we dub Regularized Saddle-Free Newton (R-SFN). The step-size α is a constant scalar, and for now we only require that $\lambda^{(k)}$ is a scalar that could possibly depend on $\mathbf{x}^{(k)}$, but we leave the form otherwise unspecified.

Under a few further assumptions, and with a specific form of regularization, we will show that if a sequence given by eq. (1.2) converges, then it almost surely converges to a second-order minimizer. We will also provide details on an efficient implementation of eq. (1.2) by computing the update via Krylov based quadrature, and we will investigate its performance experimentally.

A key tool for our analysis, and the fundamental equation behind our efficient implementation is given by the following integral representation of the matrix inverse square root:

$$\mathbf{A}^{-1/2} = \frac{2}{\pi}\int_0^\infty \big(t^2\mathbf{I} + \mathbf{A}\big)^{-1}\, dt \tag{1.3}$$

which holds for $\mathbf{A} \in \mathbb{R}^{n\times n}$ with no eigenvalues on $\mathbb{R}_-$. This is easily derived from the integral representation of the matrix square root [7].

1.1 A Motivating Example

Consider the following two dimensional quadratic:

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

This function is unbounded below, and so has no minima, but it does have a saddle point at $(0, 0)$. The gradient and the Hessian are easily computed, so we can write out the form of the Newton update as follows:

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

Thus we 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 this is not a minimum. Well, there is no minimum, but we would at least like to see behaviour that reduces the function value. If we make a minor modification and take the absolute value of the Hessian eigenvalues, we instead get the following result:

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

In this case, the saddle point is avoided if the initial iterate is not chosen on the $(x_1, 0)$ plane, and we can observe that a fast rate of "convergence" is preserved.

This example highlights the extremes of Newton's method, and seems to suggest that a simple modification could eliminate its issues. It also makes clear the necessity of an almost sure type argument, or the introduction of other mechanics to the iteration to avoid cases like an initialization of $(x_1, 0)$.

1.2 Related Work

Second order optimization methods have long been a focus of research due to their potential for fast convergence. The canonical example is Newton's method, which under mild conditions enjoys local quadratic convergence, but for which there is no known global convergence result. A number of effective

Newton variants have been established, aimed at improving different aspects of the standard version. Trust-Region Newton and Cubic Newton are two notable methods which both achieve fast global convergence. Recent work by [11] and [3] showed that an appropriately regularized Newton's method will converge at least sub-linearly for a convex objective function from any global initialization.

A naive application of Newton's method may prove to be quite ineffective. In particular, for non-convex objectives, the Hessian is no longer positive definite. Thus, the Newton update is not a descent direction, which can result in convergence to saddle points. It has been noted in the literature for a long time (see [14]) that this issue may be mitigated by taking the absolute value of the Hessian, i.e. taking the absolute value of the Hessian eigenvalues. The work of [2] popularized this idea for deep learning, and dubbed their method Saddle-Free Newton (SFN). While this work showed promising empirical results and gave some intuition as to why this approach may be valid, there was still no solid theory backing it up.

Continuing to employ the absolute value of the Hessian, [17] 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 scenarios. This technique then allows them to show avoidance of saddle points and global convergence. The works of [9] and [16] established almost sure avoidance of saddle points for gradient descent using the stable manifold theorem. We will employ this same method in our analysis. Perhaps the most similar to our work here is the unpublished report of [22]. There, the authors consider what is essentially Saddle-Free Newton with a randomized regularization term, and attempt to show saddle avoidance using the stable manifold theorem. Our work differs in a number of ways. First, our methods are fundamentally different despite having similar inspiration, and thus they require different analysis. 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.

Another issue with Newton's method and its variants comes in their implementation. In the standard Newton's method one is required to invert a potentially huge matrix, and things only get worse in the saddle-free variant. In order to apply the absolute value, most methods 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 method introduced in [15] attempts to circumvent this issue by using a low-rank approximation to the Hessian. A matrix-free technique is given by [1], where they compute the absolute value as the square root of the squared matrix via a specific ODE.

Broadly, one may consider an update of the following form:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha^{(k)} \big(\mathbf{B}^{(k)}\big)^{-1} \nabla f(\mathbf{x}^{(k)}) \tag{1.4}$$

where $\mathbf{B}_k$ is a matrix that depends on the point $\mathbf{x}_k$.

$\mathbf{B}^{(k)}$	Method	Details
$\nabla^2 f(\mathbf{x}^{(k)})$	Newton	N/A
$\nabla^2 f(\mathbf{x}^{(k)}) + \lambda^{(k)}\mathbf{I}$	Regularized Newton [11], [3]	Convex objective
$\lvert\nabla^2 f(\mathbf{x}^{(k)})\rvert$	Saddle-Free Newton [2]	N/A
$\lvert\nabla^2 f(\mathbf{x}^{(k)})\rvert_r + \gamma\mathbf{I}$	Low Rank Saddle-Free Newton [15]	Rank-r approximation
$\big\lvert\nabla^2 f(\mathbf{x}^{(k)})\rvert_m$	Non-Convex Newton [17]	Small eigenvalues replaced by m
$\Big(\big(\nabla^2 f(\mathbf{x}^{(k)})\big)^2 + \lambda^{(k)}\mathbf{I}\Big)^{1/2}$	Regularized Saddle-Free Newton (Ours)	N/A

Table 1.1: Newton variants

Our method can be seen as a combination of regularized Newton and saddle-free Newton, although from this one might expect $\mathbf{B}^{(k)} = |\nabla^2 f(\mathbf{x}^{(k)})| + \lambda^{(k)}\mathbf{I}$. Indeed in some sense this would be ideal, but the necessity for our approximation to this will be made apparent in chapter 3 when we discuss efficient computation. Among the saddle-free methods one may note that the all bound the smallest magnitude eigenvalue in some manner, whether this is via regularization or truncation.

Chapter 2

Analysis

In this chapter we will give a theoretical analysis of R-SFN's saddle avoidance properties. Here, in section 2.1, we will begin by introducing some necessary background, discuss notation, and state our assumptions. Section 2.2 will then detail the main result of this paper: almost sure avoidance of saddle points.

2.1 Preliminaries

Throughout, we will use lowercase bold letters to denote vectors, and uppercase bold letters to denote matrices or matrix valued operators. A parenthetic superscript will indicate an iteration count. To make things somewhat easier to parse we will also employ the following notation

- $\mathbf{g} = \mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x})$, with $\mathbf{g}^{(k)} = \mathbf{g}(\mathbf{x}^{(k)})$
- $\mathbf{H} = \mathbf{H}(\mathbf{x}) = \nabla^2 f(\mathbf{x})$, with $\mathbf{H}^{(k)} = \mathbf{H}(\mathbf{x}^{(k)})$

Unless otherwise specified, the norm we employ is the spectral norm, denoted as $\|\cdot\|$. As well, we will denote the Jacobian operator as D, so that the following holds for a function $\phi : \mathbb{R}^n \to \mathbb{R}^m$:

$$[D\phi(x)]_{ij} = \frac{\partial \phi_i}{\partial x_j}$$

for $i = 1 \ldots, m$ and $j = 1, \ldots, n$ – given all such partial derivatives exist. We note that in order for a ϕ 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 [10].

Our saddle avoidance analysis will require the following assumptions:

Assumption 1

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

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

Assumption 2

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

Assumption 3

The regularization has the following form:

$$\lambda = 2M\|\mathbf{g}\| + \epsilon$$

where $\epsilon > 0$ is arbitrary.

It should be noted that the regularization in assumption 2 has an implicit dependence on $\mathbf{x}$ via the gradient. Also, this assumption is not too strong of a requirement beyond assumption 1, as the latter implies almost everywhere differentiability to begin with. Assumption 3 may look a bit strange, but we will see that the extra ϵ term is quite necessary for our analysis.

Because the Hessian is a real symmetric matrix, it is orthogonally diagonalizable, so we may write the following decomposition:

$$\mathbf{H} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T \tag{2.1}$$

where $\mathbf{V}$ is orthonormal, and $\mathbf{\Sigma}$ is a diagonal matrix consisting of the eigenvalues of $\mathbf{H}$. 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. (1.1), the goal is to show convergence to the following:

Definition 1: Second-Order Stationary Point

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

Where the notation $\mathbf{B} \preceq \mathbf{A}$, for symmetric matrices $\mathbf{A}$ and $\mathbf{B}$, indicates $\mathbf{A} - \mathbf{B}$ is positive semi-definite. For a general saddle point, the smallest eigenvalue may be zero, in which case it is also a second-order stationary point, so to distinguish between the two, we introduce the following:

Definition 2: Strict Saddle Point

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

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

We will define the map associated with the R-SFN update rule (eq. (1.2)) as follows:

$$\Phi(\mathbf{x}; \alpha) = \mathbf{x} - \alpha \underbrace{\left(\mathbf{H}^2 + \lambda \mathbf{I}\right)^{-1/2} \mathbf{g}}_{\mathbf{F}(\mathbf{x})} = (\mathbf{I} - \alpha \mathbf{F})(\mathbf{x}) \tag{2.2}$$

where we note that the fixed points of Φ are exactly the critical points of f. This form of our method will be useful for the forthcoming analysis. When the step-size is clear from the context or unspecified, we will just write $\Phi(\mathbf{x})$.

2.2 Saddle Avoidance

The motivation behind using the absolute value of the Hessian is that it allows one to retain the "appropriate" scaling of Newton's method, while preventing the possibility for convergence to saddle points. Our method uses an approximation to the absolute value, but in this section we will show that this saddle avoidance behaviour indeed holds almost surely. Our analysis will follow that of [9] and [16] by employing the following result:

Theorem 1: Stable Manifold [19, III.7]

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

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

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

To apply theorem 1, and for a few other key results, it will be necessary to have an explicit form for the derivative $D\Phi$, so we will derive this before proceeding any further. We refer the reader to [10] for references on matrix calculus. We begin by using eq. (1.3) to write the R-SFN map (eq. (2.2)) as follows:

$$\Phi(\mathbf{x}) = \mathbf{x} - \frac{2\alpha}{\pi} \int_0^\infty \Big((t^2 + \lambda)\mathbf{I} + \mathbf{H}^2 \Big)^{-1} \mathbf{g} \, dt$$

Now let $\mathbf{T}(\mathbf{x}) = \big((\lambda + t^2)\mathbf{I} + \mathbf{H}^2 \big)$ and consider the following:

$$D\Phi = \frac{\partial \Phi}{\partial \mathbf{x}^T} = \mathbf{I} - \frac{2\alpha}{\pi} \int_0^\infty \frac{\partial}{\partial \mathbf{x}^T} \Big[\Big((t^2 + \lambda)\mathbf{I} + \mathbf{H}^2 \Big)^{-1} \mathbf{g} \Big] \, dt$$

where we have swapped the integral and the derivative in the second term using the dominated convergence theorem. From appendix A we have the following result:

$$\frac{\partial \Big((t^2 + \lambda)\mathbf{I} + \mathbf{H}^2 \Big)^{-1} \mathbf{g}}{\partial \mathbf{x}^T} = \big(\mathbf{g}^T \otimes \mathbf{I} \big) \frac{\partial \text{vec}(\mathbf{T}^{-1})}{\partial \mathbf{x}^T} + \big(1 \otimes \mathbf{T}^{-1}(\mathbf{x}) \big) \frac{\partial \text{vec}(\mathbf{g})}{\partial \mathbf{x}^T}$$

The first term simplifies according to the result in appendix A, and it is easy to see that the second term is given by $\mathbf{T}^{-1}\mathbf{H}$. Combining all of this together we get the following:

$$D\Phi(\mathbf{x}) = \mathbf{I} - \frac{2\alpha}{\pi} \int_0^\infty \mathbf{T}^{-1} \big(\mathbf{I} - c\mathbf{T}^{-1}\mathbf{g}\mathbf{g}^T \big)\mathbf{H} - \big(\mathbf{g}^T \mathbf{T}^{-1} \mathbf{H} \otimes \mathbf{T}^{-1} + \mathbf{g}^T \mathbf{T}^{-1} \otimes \mathbf{T}^{-1} \mathbf{H} \big) \frac{\partial \text{vec}(\mathbf{H})}{\partial \mathbf{x}^T} \, dt \qquad (2.3)$$

where $c = 2M\|\mathbf{g}\|^{-1}$.

First, we will use eq. (2.3) to prove the following lemma. This will be key to using the stable manifold theorem in a helpful way.

Lemma 1

For $\mathbf{x}_s$ a strict saddle point, $D\Phi(\mathbf{x}_s)$ 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. (2.3) reduces to the following:

$$D\Phi(\mathbf{x}) = \mathbf{I} - \frac{2\alpha}{\pi}\int_0^\infty \mathbf{T}^{-1}\mathbf{H}\, dt = \mathbf{I} - \mathbf{H}\big(\mathbf{H}^2 + \epsilon\mathbf{I}\big)^{-1/2}$$

The three matrices involved are simultaneously diagonalizable, so we may write the following:

$$D\Phi(\mathbf{x}_s) = \mathbf{V}\Big(\mathbf{I} - \alpha\mathbf{\Sigma}\big(\mathbf{\Sigma}^2 + \epsilon\mathbf{I}\big)^{-1/2}\Big)\mathbf{V}^T$$

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

$$1 - \frac{\mu_i\alpha}{\sqrt{\mu_i^2 + \epsilon}}$$

By assumption, at least μ_n is negative, which implies the following:

$$1 - \frac{\mu_n\alpha}{\sqrt{\mu_n^2 + \epsilon}} > 1$$

Thus the result holds. ▲

Next, we will use eq. (2.3) to prove the lemma below.

Lemma 2

The nonlinear operator $\mathbf{F}$, from eq. (2.2), satisfies the following for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$:

$$\|\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})\| < 2\|\mathbf{x} - \mathbf{y}\|$$

Proof

Our approach will be to show that the norm of $D\mathbf{F}$, which was derived for eq. (2.3) is bounded by 2, which immediately implies the result.

$$\begin{aligned}\|D\mathbf{F}\| &\leq \frac{2}{\pi}\int_0^\infty \|\mathbf{T}^{-1}\big(\mathbf{I} - c\mathbf{T}^{-1}\mathbf{g}\mathbf{g}^T\big)\mathbf{H} - \big(\mathbf{g}^T\mathbf{T}^{-1}\mathbf{H}\otimes\mathbf{T}^{-1} + \mathbf{g}^T\mathbf{T}^{-1}\otimes\mathbf{T}^{-1}\mathbf{H}\big)\frac{\partial \mathrm{vec}(\mathbf{H})}{\partial \mathbf{x}^T}\|\, dt \\ &\leq \frac{2}{\pi}\int_0^\infty \|\mathbf{T}^{-1}\big(\mathbf{I} - c\mathbf{T}^{-1}\mathbf{g}\mathbf{g}^T\big)\mathbf{H}\| + \|\big(\mathbf{g}^T\mathbf{T}^{-1}\mathbf{H}\otimes\mathbf{T}^{-1} + \mathbf{g}^T\mathbf{T}^{-1}\otimes\mathbf{T}^{-1}\mathbf{H}\big)\frac{\partial \mathrm{vec}(\mathbf{H})}{\partial \mathbf{x}^T}\|\, dt\end{aligned}$$

We can bound the second term in the integrand as follows (see appendix A):

$$\|\big(\mathbf{g}^T\mathbf{T}^{-1}\mathbf{H}\otimes\mathbf{T}^{-1} + \mathbf{g}^T\mathbf{T}^{-1}\otimes\mathbf{T}^{-1}\mathbf{H}\big)\frac{\partial \mathrm{vec}(\mathbf{H})}{\partial \mathbf{x}^T}\| < \frac{\mu\lambda}{(\mu^2+t^2+\lambda)(\mu_{\min}^2+t^2+\lambda)}$$

where μ is defined as follows:

$$\mu = \max_{\mu_i}\frac{\mu_i}{\mu_i^2+t^2+\lambda}$$

In other words, μ is the eigenvalue that maximizes the given fraction. Next, we consider the first term in the integrand, which we rewrite as $\|\mathbf{T}^{-2}\big(\mathbf{T} - c\mathbf{g}\mathbf{g}^T\big)\mathbf{H}\|$. Then we write the following:

$$\|\mathbf{T}^{-2}\big(\mathbf{T} - c\mathbf{g}\mathbf{g}^T\big)\mathbf{H}\|^2 = \|\mathbf{T}^{-2}\big(\mathbf{T} - c\mathbf{g}\mathbf{g}^T\big)\mathbf{H}^2\big(\mathbf{T} - c\mathbf{g}\mathbf{g}^T\big)\mathbf{T}^{-2}\|$$

This quantity is bounded above by $||\mathbf{T}^{-1}\mathbf{H}^2\mathbf{T}^{-1}|| = ||\mathbf{T}^{-1}\mathbf{H}||^2$ if the following holds:

$$\mathbf{T}^{-2}(\mathbf{T} - c\mathbf{g}\mathbf{g}^T)\mathbf{H}^2(\mathbf{T} - c\mathbf{g}\mathbf{g}^T) \preceq \mathbf{T}\mathbf{H}^2\mathbf{T}$$

Rearranging the LHS and cancelling terms we end up with the following:

$$0 \preceq \mathbf{g}\mathbf{g}^T\mathbf{H}^2\mathbf{T} + \big(\mathbf{T} - c\mathbf{g}\mathbf{g}^T\big)\mathbf{H}^2\mathbf{g}\mathbf{g}^T$$

which holds because $\|c\mathbf{g}\mathbf{g}^T\| = \lambda$ and all the eigenvalues of $\mathbf{T}$ are of the form $\mu_i^2 + t^2 + \lambda$. Putting our two pieces together we have the following bound:

$$\|D\mathbf{F}\| < \frac{2}{\pi}\int_0^\infty \frac{\mu}{\mu^2+t^2+\lambda} + \frac{\mu\lambda}{(\mu^2+t^2+\lambda)(\mu_{\min}^2+t^2+\lambda)}\, dt \leq \frac{2}{\pi}\int_0^\infty \frac{2\mu}{\mu^2+t^2+\lambda}\, dt$$

Carrying out the integration we get our final result:

$$\|D\mathbf{F}\| < \frac{2\mu}{\sqrt{\mu^2+\lambda}} < 2$$

▲

This immediately implies the following two corollaries which will allow us to apply the stable manifold theorem, and extend it to a global statement.

Corollary 1

The R-SFN map with step-size 1/2 is a local diffeomorphism.

Proof

Via the Inverse Function Theorem [21] the map Φ is a local diffeomorphism if the derivative at a point $\mathbf{x}$, $D\Phi(\mathbf{x})$, is a linear isomorphism. The first criterion is easily satisfied because $D\Phi(\mathbf{x})$ is a matrix. From lemma 2 we know that the $\frac{1}{2}||D\mathbf{F}(\mathbf{x})|| < 1$, so we can conclude that the eigenvalues of $D\Phi = \mathbf{I} - \frac{1}{2}\mathbf{F}$ are strictly larger than 0. Thus, $D\Phi$ is invertible, so Φ is a local diffeomorphism. ▲

Corollary 2

The R-SFN map with step-size 1/2 is injective.

Proof

Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ be given such that $\mathbf{x} \neq \mathbf{y}$. Now assume that $\Phi(\mathbf{x}) = \Phi(\mathbf{y})$, which implies the following:

$$\mathbf{x} - \frac{1}{2}\mathbf{F}(\mathbf{x}) = \mathbf{y} - \frac{1}{2}\mathbf{F}(\mathbf{y}) \implies \|\mathbf{x} - \mathbf{y}\| = \frac{1}{2}\|\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})\|$$

Using the inequality from lemma 2 we can write the following:

$$\|\mathbf{x} - \mathbf{y}\| < \|\mathbf{x} - \mathbf{y}\|$$

which is clearly a contradiction. Thus we conclude that $\mathbf{x} = \mathbf{y}$, and the result holds. ▲

We will also state the following theorem as it is key to our analysis, and somewhat less known.

Theorem 2: Alexandroff [5, 1.12.8]

If $\phi : X \to Y$ is a surjective open mapping from a separable locally compact metric space to a separable metric space such that $\left|\phi^{-1}(y)\right| \leq \aleph_0$ for all $y \in Y$, then $\dim(X) = \dim(Y)$.

Now we move to the the main results of this section:

Theorem 3: Saddle Avoidance

The set of points in $\mathbb{R}^n$ that converge to a strict saddle point of f under the R-SFN map (eq. (2.2)), with step-size $1/2$, is of Lebesgue measure zero.

Proof

Let $\mathbf{x}_s$ be a strict saddle point. Corollary 1 gives $\Phi : U \to \mathbb{R}^n$ as a C^1 local diffeomorphism for U a neighborhood of $\mathbf{x}_s$, so applying theorem 1 yields the C^1 manifold $W_s = W(\mathbf{x}_s)$. From lemma 1 we know that there is at least one eigenvalue of $D\Phi(\mathbf{x}_s)$ that is strictly larger then 1, so we conclude that $\dim(W_s) < n$.

For each strict saddle point $\mathbf{x}_s$, define B_s to be the ball given by theorem 1. Via Lindelöf's lemma [8], we can find a countable set of these balls such that the following holds:

$$\bigcup_{m=1}^{\infty} B_{s_m} = \bigcup_{s} B_s$$

i.e. a countable sub-cover for the union of the balls B_s. Now suppose that the iteration eq. (1.2) converges to a strict saddle point from the starting point $\mathbf{x}^{(0)}$. This implies that there exists a K such that $\mathbf{x}^{(k)} = \Phi^k(\mathbf{x}^{(0)}) \in B_{s_m}$ for all $k \geq K$ and some $m \in \mathbb{N}$. Theorem 1 further implies that $\Phi^K(\mathbf{x}^{(0)}) \in W_s$, so it follows that $\mathbf{x}^{(0)} \in \Phi^{-K}(W_s)$.

Now, we claim that $\Phi^{-K}(W_s)$ has measure zero. From corollary 2 we have that Φ is injective, and given it is also continuous, the Invariance of Domain theorem [13] guarantees that it is an open map. Let $A = \Phi^{-1}(W_s)$ and define the map $\tilde{\Phi} : A \to \Phi(A) \subset W_s$ as the restriction of Φ to A, which is surjective by construction. Observe that A and $\Phi(A)$ are separable, because they are subsets of $\mathbb{R}^n$, and A is locally compact because it is the continuous pre-image of a disc. Theorem 2 implies that $\dim(A) = \dim(\Phi(A)) < n$. Proceeding via induction guarantees that $\Phi^{-K}(W_s)$ has dimension strictly less than n, and thus it has Lebesgue measure zero.

From here it holds that the following set has measure zero:

$$S = \bigcup_{m=1}^{\infty} \bigcup_{k=0}^{\infty} \Phi^{-k}(W_{s_m})$$

which holds because countable unions of sets of measure zero are measure zero. The set S is precisely the set of points in $\mathbb{R}^n$ that converge to a strict saddle point of the objective function f. Thus, the result holds. ▲

The proof above uses an argument similar to [16], which is itself a generalization of the argument in [9]. It is important, however, to point out how our approach differs. In particular, this comes in how we show that the composed inverse map Φ^{-1} of the embedded disc W is measure zero. The argument used in the aforementioned work relies on the fact that their map is a diffeomorphism, thus the inverse is locally Lipschitz, and such maps are null-set preserving. In contrast, we used the fact that our map was injective to apply theorem 2 (Alexandroff's), which is an approach unique to our work. In order to prove that this was the case in corollary 2 we used the result of lemma 2, which relied on the existence of a third continuous derivative of f. However, in some sense our approach is more general, because if one can show lemma 2 holds without using differentiability, then we need only assume f is C^3 in a neighborhood around the saddle points (to satisfy theorem 1). This can actually be taken a step further by noting the conditions for Alexandroff's theorem, which are more general than we have required. In fact, a variety of similar results can be found in [5] and may allow for different proof strategies.

Theorem 3 implies the following corollary:

Corollary 3

Assume that the starting point $\mathbf{x}^{(0)}$ is chosen according to an absolutely continuous probability density. If the sequence of iterates $(\mathbf{x}^{(k)})$ given by the R-SFN iteration (eq. (1.2)) with step-size $1/2$ converges, then this sequence converges to a second-order stationary point almost surely.

Proof

Let $\mathbf{x}^{(\infty)} = \lim_{k\to\infty} \mathbf{x}^{(k)}$ and note that it must be a critical point of f. Theorem 3 gives the Lebesgue measure of the set of points that converge to a strict saddle point as 0. Thus the probability that $\mathbf{x}^{(\infty)}$ is a strict saddle is 0, and if it is not a strict saddle it must be a second-order stationary point. Thus we conclude that $\mathbf{x}^{(\infty)}$ is a second-order stationary point with probability 1. ▲

Note that we require the sequence to be convergent for this result to hold, which we have not guaranteed. Given the update is a descent direction, it is hopeful that some result can be established here, and that is discussed further in chapter 4. It is also worthwhile to restate the fact that convergence to a second-order stationary point is equivalent to convergence to a local minimizer if all of the saddle points are strict.

The main results of this section, theorem 3 and corollary 3, have required a step-size of 1/2. This requirement ultimately stems from the bound in lemma 2, but there is a question as to whether a sharper bound can be found. For example, in the case of a 1d or quadratic objective function, the following bound can be shown to hold:

$$\|\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})\| < \|\mathbf{x} - \mathbf{y}\|$$

and thus a step-size of 1 yields the saddle avoidance results. Why it is that such a step-size is desirable is discussed in chapter 4, and the 1d case is presented in appendix B. The quadratic case is quite simplified by the fact that the third derivative is zero, so eq. (2.3) reduces considerably, and a similar proof to that of lemma 2 shows the bound in question. Obviously these results do not prove it in general, but they do seem to point toward the possibility.

Chapter 3

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.

To start, we can avoid an eigen-decomposition by using the integral form for the matrix square root (eq. (1.3)). Applying this to the update portion of R-SFN (eq. (1.2)) we get the following:

$$\Big(\big(\mathbf{H}^{(k)}\big)^2 + \lambda^{(k)}\mathbf{I}\Big)^{-1/2}\mathbf{g}^{(k)} = \frac{2\alpha}{\pi}\int_0^\infty \Big(\big(t^2 + \lambda^{(k)}\big)\mathbf{I} + \big(\mathbf{H}^{(k)}\big)^2\Big)^{-1}\mathbf{g}^{(k)}\ dt \tag{3.1}$$

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:

$$\Big(\big(\mathbf{H}^{(k)}\big)^2 + \lambda^{(k)}\mathbf{I}\Big)^{-1/2}\mathbf{g}^{(k)} \approx \frac{2\alpha}{\pi}\sum_{i=1}^{N} w_i\Big(\big(t_i^2 + \lambda^{(k)}\big)\mathbf{I} + \big(\mathbf{H}^{(k)}\big)^2\Big)^{-1}\mathbf{g}^{(k)} \tag{3.2}$$

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 chapter 2 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. (1.2) 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. (3.1) satisfies this, but we need positive weights for our approximation (eq. (3.2)) to as well.

In general, the summand of eq. (3.2) can be computed efficiently using a Krylov subspace method. In particular, taking inspiration from [4], 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 $\mathbf{A}\mathbf{x} = \mathbf{b}$, the associated order r Krylov subspace is given as follows:

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

It can then be shown that $\mathcal{K}_r(\mathbf{A}, \mathbf{b}) = \mathcal{K}_r(\mathbf{A} + \lambda\mathbf{I}, \mathbf{b})$, and although we will not state the full algorithm, this feature means one only need construct the Krylov subspace once [6]. 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. (3.2) 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 following algorithm:

Algorithm 1: R-SFN Update

Data: Objective function f, Current iterate $\mathbf{x}$, Quadrature order N, Hessian Lipschitz constant M, Step-size α, Krylov solver tolerance τ

$\mathbf{g} \leftarrow \nabla f(\mathbf{x})$

$\mathbf{H}^2 \leftarrow \left(\nabla^2 f(\mathbf{x})\right)^2$ `// Could be a matrix-free operator`

$\mathbf{t}, \mathbf{w} \leftarrow \text{Quadrature}(N)$ `// Compute quadrature nodes and weights`

$s_i \leftarrow t_i^2 + \lambda$ `// Compute shifts, agnostic to form of` λ

$\mathbf{Y} \leftarrow \text{ShiftedKrylov}(\mathbf{H}^2, \mathbf{g}, \mathbf{s}, \tau)$ `// Compute solutions to linear systems`

for $i = 1 : N$ **do**

  $\mathbf{x} = \mathbf{x} - \alpha w_i \mathbf{Y}_i$

end

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.

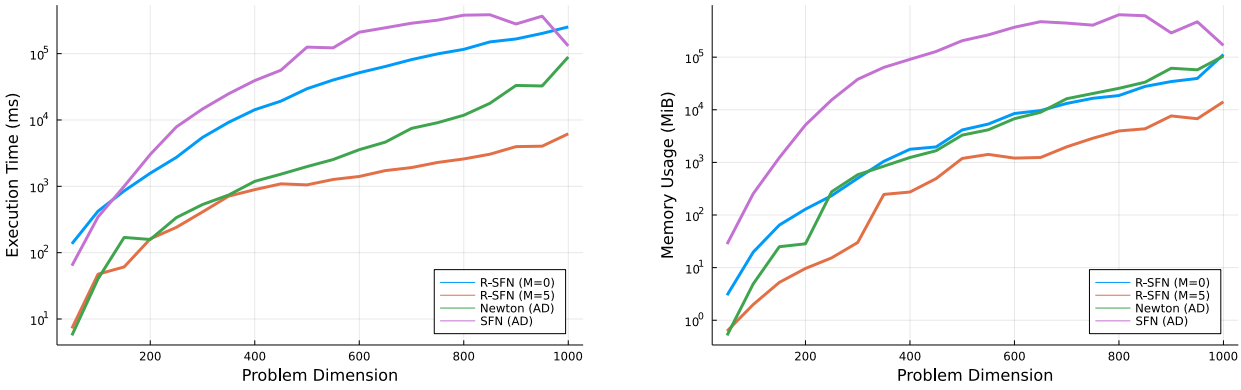
3.1 Numerical Experiments

Our goal in this section is to present a proof-of-concept for R-SFN as a practical saddle-free Newton type method for optimization. To that end, we will explore its efficiency and its convergence, although we will only do so in a limited capacity. Certainly there is more to be investigated, and that will be discussed as future work in chapter 4.

The following experiments were conducted in the Julia programming language, the code for which can be found in the GitHub repository [20]. 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 [12] from the Julia Smooth Optimizers organization, where we use a matrix free operator for Hessian-vector products generated via mixed forward over backward AD. We note that in all of our experiments a step-size of $1/2$ is used for R-SFN, and a step-size of 1 is used for Newton's method unless otherwise specified. Lastly, the regularization is computed according to the theory provided in section 2.2, the form of which is given by assumption 3. In particular, we recall that M is the Hessian Lipschitz constant, as this will be an important parameter considered in our experiments.

3.1.1 Efficiency

Here, we consider the execution time and memory consumption for a single update step of the form eq. (1.4). We use a synthetic binary logistic regression problem, where we vary the number of features, and use 10 times that many observations. We compare the standard Newton's method, saddle-free Newton (SFN), and R-SFN. For all methods we use automatic differentiation (AD), but in different ways. For R-SFN and Newton's method we use it to construct a matrix free operator, but for SFN we use it to construct the full Hessian so that we can perform an eigen-decomposition. We will point out that this problem admits an analytic form for the Hessian that we could have used for Newton's method and SFN. Indeed, doing so would improve the performance of those approaches, at least for smaller problems, but it is more realistic to expect one will not have access to such an analytic form in general.



(a) Update step run time

(b) Update step memory consumption

Figure 3.1: Comparison of execution time and memory consumption for Newton type methods.

The most immediate observation one may make is that the choice of M has a large impact on the run-time performance of R-SFN. It is somewhat unclear why this is, but a possible explanation is that regularization plays a large role in the solvability of the linear systems. In the Krylov solver we are asking for a high degree of accuracy, so with no regularization (i.e. $M = 0$) it is possible this is much harder to obtain. In the case of non-zero regularization, we see that R-SFN performs at least an order of magnitude better than SFN in both metrics. This is notable because with zero regularization, R-SFN and SFN are essentially the same method, differing only in how the update step is computed. It should also be noted that

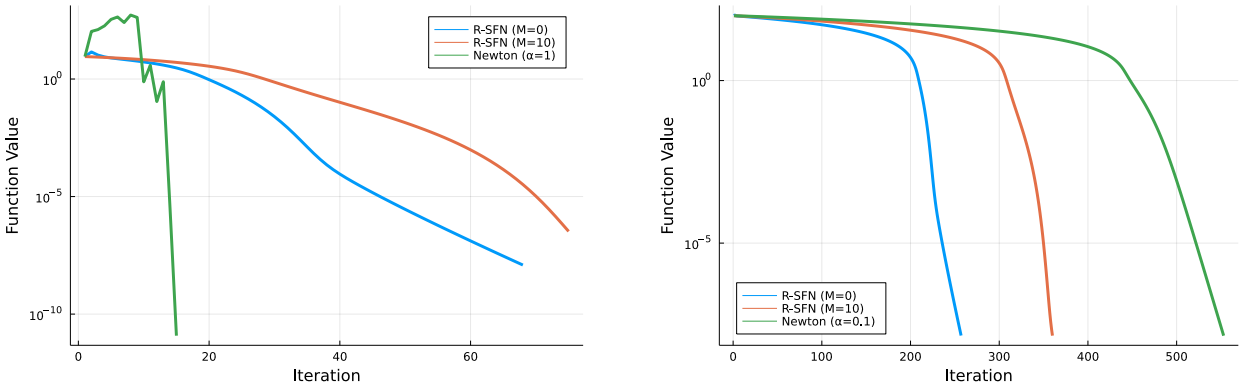
the problem dimensions we are considering here are not necessarily that large, so stronger differences may be noted in other regimes.

3.1.2 Convergence

Here, we consider a similar experiment to that of [15]. The d-dimensional Rosenbrock function is a non-convex function often used for benchmarking, and is given as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{d-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$$

It has a global minimum of 0 which is achieved by the unique minimizer $\mathbf{1}$. This problem does not have a Lipschitz Hessian, so we pick two values for M in our experiments, and we use the maximum number of quadrature nodes (197) that can be computed under machine precision using Gauss-Laguerre. We start at the initial point $\mathbf{x}^{(0)} = \mathbf{0}$ and apply R-SFN, with Newton's method serving as a comparison.



(a) 10-dimensional

(b) 100-dimensional

Figure 3.2: Minimization of Rosenbrock function.

In the 10-dimensional case we see that Newton's method exhibits some erratic behaviour at the beginning, but quickly converges to the minimum. Both versions of R-SFN also converge to the minimum, but do so in much more of a steady manner. Without regularization, i.e. $M = 0$, we see R-SFN does briefly increase the function, but with sufficient regularization R-SFN appears to always descend, albeit at a slower rate. This perhaps points towards regularization also being necessary for convergence, but in a trade-off with speed. In the 100-dimensional case we can see that R-SFN actually outperforms Newton's method. This is

likely due to the smaller step-size, without which Newton's method does not converge. Again, we also see that less regularization results in faster convergence.

Chapter 4

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 avoids saddle points. We also presented an efficient implementation of the method which allows for fast matrix-free updates. Much recent work has been carried out towards developing Newton type methods that avoid saddle points, yield strong global convergence, or can be easily applied in practice. We view R-SFN as a strong step towards combining all of these features. Perhaps more than anything, the work presented here points towards many promising directions for further investigation.

The step-size of 1/2 required for the saddle avoidance results may initially appear sub-optimal, but it is unclear if that is the case. In the standard Newton's method, a step-size of 1 is the ideal quantity, and smaller steps are introduced only to guarantee convergence results. However, our method isn't derived in the same way, so the importance of a unit step-size is somewhat unclear. One reason that it may be desirable stems from the Dennis-Moré condition for local super-linear convergence [14], although it is certainly possible a global convergence result could still be shown regardless. The fact that it is a constant is really a key feature, as no added computation is required. If the objective were convex, the steps with step-size 1 would actually be larger than that of regularized Newton. In the end, what really matters is the effect of the step-size on convergence, and its associated rate. This leads to the two most significant open questions: can a global guarantee of convergence to a critical point can be obtained, and can a rate of convergence be obtained?

Many questions are raised on the implementation side of things. In particular, how do the approxima-

tions due to the quadrature and Krylov solutions impact the theory? What is even the best quadrature rule to use? Often in practice, especially in machine learning regimes, we use mini-batches of data. This results in sub-sampled derivative information, which introduces yet another source of randomness that must be integrated into the analysis. It is also unlikely that one would have access to the Hessian Lipschitz constant. Thus, an appropriate line search procedure would be needed for a general algorithm. Even with all of this in hand, to really understand R-SFN, many more numerical experiments are needed.

Bibliography

[1] M. Arjovsky, Saddle-free hessian-free optimization, arXiv preprint arXiv:1506.00059, (2015).

[2] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, Identifying and attacking the saddle point problem in high-dimensional non-convex optimization, Advances in neural information processing systems, 27 (2014).

[3] N. Doikov and Y. Nesterov, Gradient regularization of newton method with bregman distances, arXiv preprint arXiv:2112.02952, (2021).

[4] J.-P. Dussault and D. Orban, Scalable adaptive cubic regularization methods, arXiv preprint arXiv:2103.16659, (2021).

[5] R. Engelking, ed., Dimension Theory of Separable Metric Spaces, vol. 19, PWN-Polish Scientific Publishers - Warszawa, 1979.

[6] A. Frommer and P. Maass, Fast cg-based methods for tikhonov–phillips regularization, SIAM Journal on Scientific Computing, 20 (1999), pp. 1831–1850.

[7] N. J. Higham, Functions of matrices: theory and computation, SIAM, 2008.

[8] J. L. Kelley, General Topology, Springer, 1955.

[9] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, Gradient descent only converges to minimizers, in Conference on learning theory, PMLR, 2016, pp. 1246–1257.

[10] J. R. Magnus and H. Neudecker, Matrix Differential Calculus with Applications in Statistics and Econometrics, John Wiley & Sons, 2019.

[11] K. Mishchenko, Regularized newton method with global $o(1/k^2)$ convergence, arXiv preprint arXiv:2112.02089, (2021).

[12] A. Montoison, D. Orban, and contributors, Krylov.jl: A Julia basket of hand-picked Krylov methods. https://github.com/JuliaSmoothOptimizers/Krylov.jl, June 2020.

[13] J. R. Munkres, Elements of algebraic topology, CRC press, 2018.

[14] J. Nocedal and S. J. Wright, Numerical Optimization, Springer, 2e ed., 2006.

[15] T. O'Leary-Roseberry, N. Alger, and O. Ghattas, Low rank saddle free newton: A scalable method for stochastic nonconvex optimization, arXiv preprint arXiv:2002.02881, (2020).

[16] I. Panageas and G. 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.

[17] S. Paternain, A. Mokhtari, and A. Ribeiro, A newton-based method for nonconvex optimization with fast evasion of saddle points, SIAM Journal on Optimization, 29 (2019), pp. 343–368.

[18] A. Rainer, Perturbation theory for normal operators, Transactions of the American Mathematical Society, 365 (2013), pp. 5545–5577.

[19] M. Shub, Global Stability of Dynamical Systems, Springer, 1987.

[20] C. Simpson, R-SFN. `https://github.com/RS-Coop/R-SFN`, 2022. A Julia implementation of the regularized saddle-free Newton method.

[21] M. Spivak, Calculus On Manifolds, Addison-Wesley, 1965.

[22] T. T. Truong, T. D. To, T. H. Nguyen, T. H. Nguyen, H. P. Nguyen, and M. Helmy, A fast and simple modification of newton's method helping to avoid saddle points, arXiv preprint arXiv:2006.01512, (2020).

Appendix A

Matrix Derivative Results

We will show the following result [10, exercise 9.14.6] holds for $\mathbf{A}(\mathbf{x}) \in \mathbb{R}^{m\times p}$, $\mathbf{B}(\mathbf{x}) \in \mathbb{R}^{p\times r}$, and $\mathbf{x} \in \mathbb{R}^n$:

$$\frac{\partial}{\partial \mathbf{x}^T}\big[\mathbf{A}(\mathbf{x})\mathbf{B}(\mathbf{x})\big] = \big(\mathbf{B}(\mathbf{x})^T \otimes \mathbf{I}_m\big)\frac{\partial \mathrm{vec}\big(\mathbf{A}(\mathbf{x})\big)}{\partial \mathbf{x}^T} + \big(\mathbf{I}_r \otimes \mathbf{A}(\mathbf{x})\big)\frac{\partial \mathrm{vec}\big(\mathbf{B}(\mathbf{x})\big)}{\partial \mathbf{x}^T}$$

where the subscript on the identity indicates its size.

Proof

We begin by computing the differential of $\mathbf{A}(\mathbf{x})\mathbf{B}(\mathbf{x})$:

$$\partial\big(\mathbf{A}(\mathbf{x})\mathbf{B}(\mathbf{x})\big) = \partial\big(\mathbf{A}(\mathbf{x})\big)\mathbf{B}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\partial\big(\mathbf{B}(\mathbf{x})\big)$$

Then apply the vec operator to get the following result:

$$\mathrm{vec}\big(\mathbf{A}(\mathbf{x})\mathbf{B}(\mathbf{x})\big) = \big(\mathbf{B}(\mathbf{x})^T \otimes \mathbf{I}_m\big)\partial\mathrm{vec}\big(\mathbf{A}(\mathbf{x})\big) + \big(\mathbf{I}_r \otimes \mathbf{A}(\mathbf{x})\big)\partial\mathrm{vec}\big(\mathbf{B}(\mathbf{x})\big)$$

From here we simply read of the result. ▲

With $\mathbf{T} = (t^2 + \lambda)I + \mathbf{H}^2$ and λ as in assumption 3 we will show the following:

$$\big(\mathbf{g}^T \otimes \mathbf{I}\big)\frac{\partial \mathrm{vec}(\mathbf{T}^{-1})}{\partial \mathbf{x}^T} = -c\mathbf{T}^{-2}\mathbf{g}\mathbf{g}^T\mathbf{H} - \big(\mathbf{g}^T\mathbf{T}^{-1}\mathbf{H} \otimes \mathbf{T}^{-1} + \mathbf{g}^T\mathbf{T}^{-1} \otimes \mathbf{T}^{-1}\mathbf{H}\big)\frac{\partial \mathrm{vec}(\mathbf{H})}{\partial \mathbf{x}^T}$$

where $c = 2M\|\mathbf{g}\|^{-1}$:

Proof

We start with the following differentials:

$$\partial\text{vec}(\mathbf{T}^{-1}) = -\big(\mathbf{T}^{-1} \otimes \mathbf{T}^{-1}\big)\partial\text{vec}(\mathbf{T})$$

$$\partial\text{vec}(\mathbf{T}) = \partial\text{vec}(\lambda\mathbf{I}) + \partial\text{vec}\big(\mathbf{H}^2\big) = \partial\text{vec}(\lambda\mathbf{I}) + \big(\mathbf{H} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{H}\big)\partial\text{vec}(\mathbf{H})$$

Putting these together then implies the result below:

$$\frac{\partial\text{vec}(\mathbf{T}^{-1})}{\partial\mathbf{x}^T} = -\big(\mathbf{T}^{-1} \otimes \mathbf{T}^{-1}\big)\frac{\partial\lambda\mathbf{I}}{\partial\mathbf{x}^T} + \big(\mathbf{T}^{-1}\mathbf{H} \otimes \mathbf{T}^{-1} + \mathbf{T}^{-1} \otimes \mathbf{T}^{-1}\mathbf{H}\big)\frac{\partial\text{vec}(\mathbf{H})}{\partial\mathbf{x}^T}$$

Multiplying through by $\big(\mathbf{g}^T \otimes \mathbf{I}\big)$ gives the last term of the result in question. Next, we note the following:

$$\big(\mathbf{g}^T \otimes \mathbf{I}\big)\big(\mathbf{T}^{-1} \otimes \mathbf{T}^{-1}\big) = \begin{bmatrix} \big(\mathbf{g}^T\mathbf{T}_1^{-1}\mathbf{T}^{-1}\big) & \cdots & \big(\mathbf{g}^T\mathbf{T}_n^{-1}\mathbf{T}^{-1}\big) \end{bmatrix}$$

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

where the subscript on $\mathbf{T}$ indicates the column. The i, j element of their product is given by the following expression:

$$c\sum_{k=1}^{n}\big(\mathbf{g}^T\mathbf{T}^{-1}\big)_k\mathbf{T}_{ik}^{-1}\big(\mathbf{g}^T\mathbf{H}\big)_j = c\big(\mathbf{T}^{-1}\mathbf{T}^{-1}\mathbf{g}\mathbf{g}^T\mathbf{H}\big)_{ij}$$

Thus we see that the result holds. ▲

We will show the following:

$$\|\big(\mathbf{g}^T\mathbf{T}^{-1}\mathbf{H} \otimes \mathbf{T}^{-1} + \mathbf{g}^T\mathbf{T}^{-1} \otimes \mathbf{T}^{-1}\mathbf{H}\big)\frac{\partial\text{vec}(\mathbf{H})}{\partial\mathbf{x}^T}\| < \frac{\mu\lambda}{(\mu^2 + t^2 + \lambda)(\mu_{\min} + t^2 + \lambda)}$$

where μ is defined as follows:

$$\mu = \max_{\mu_i}\frac{\mu_i}{\mu_i^2 + t^2 + \lambda}$$

Proof

We first note that $\mathbf{T}^{-1} \otimes \mathbf{T}^{-1}\mathbf{H}$ and $\mathbf{T}^{-1}\mathbf{H} \otimes \mathbf{T}^{-1}$ have eigenvalues of the following form:

$$\frac{\mu_i}{(\mu_i^2 + t^2 + \lambda)(\mu_j^2 + t^2 + \lambda)} \leq \frac{\mu}{(\mu^2 + t^2 + \lambda)(\mu_{\min}^2 + t^2 + \lambda)}$$

which is a well known property of Kronecker products [10]). Note that the eigenvalue μ is defined as

above. We can then write the following:

$$\|\left(\mathbf{g}^T\mathbf{T}^{-1}\mathbf{H}\otimes\mathbf{T}^{-1}+\mathbf{g}^T\mathbf{T}^{-1}\otimes\mathbf{T}^{-1}\mathbf{H}\right)\frac{\partial \text{vec}(\mathbf{H})}{\partial \mathbf{x}^T}\| < \frac{\mu 2M||\mathbf{g}||}{(\mu^2+t^2+\lambda)(\mu_{\min}+t^2+\lambda)}$$

which follows because the norm of $\frac{\partial \text{vec}(\mathbf{H})}{\partial \mathbf{x}^T}$ is bounded above by M due to assumption 1. Recognizing $\lambda - \epsilon$ in the numerator gives the final result. ▲

Appendix B

1d Lipschitz Bound

We will show that for a 1d objective function, the following holds for all $x, y \in \mathbb{R}$:

$$\|F(x) - F(y)\| < \|x - y\|$$

where $F = \big((f'')^2 + \lambda\big)^{-1/2} f'$ is defined in eq. (2.2).

Proof

Our approach will be to bound the derivative of F, so to that end we write the following:

$$\frac{d}{dx}F = \frac{f''}{\big((f'')^2 + \lambda\big)^{1/2}} - \frac{f' \cdot (2f''f''' + \lambda')}{2\big((f'')^2 + \lambda\big)^{3/2}}$$

Combining the two fractions yields the following:

$$\frac{d}{dx}F = \frac{2(f'')^3 + 2\lambda f'' - 2f'f''f''' - \lambda' f'}{2\big((f'')^2 + \lambda\big)^{3/2}}$$

In this setting we have $\lambda = 2M|f'| + \epsilon$, so this gives $\lambda' = 2M\text{sign}(f')f''$. Plugging this into the equation above and recognizing that $\text{sign}(f')f' = |f'|$ results in the following:

$$\frac{d}{dx}F = \frac{2(f'')^3 + (2M|f'| + 2\epsilon)f'' - 2f'f''f'''}{2\big((f'')^2 + \lambda\big)^{3/2}}$$

From here, we take the absolute value, apply the triangle inequality and simplify:

$$\Big|\frac{d}{dx}F\Big| < |f''|\frac{|f''|^2 + 2M|f'| + \epsilon}{\big((f'')^2 + \lambda\big)^{3/2}} = \frac{|f''|}{\big((f'')^2 + \lambda\big)^{1/2}} \leq 1$$

where we have also used the fact that $|f'''| \leq M$ (by assumption) for the inequality, and the form of λ for the equality. Thus, the result holds. ▲