Generalization of Quasi-Newton Methods: Application to Robust Symmetric Multisecant Updates

Damien Scieur* DAMIEN.SCIEUR@GMAIL.COM
Samsung - SAIT AI Lab Montreal
Lewis Liu* ALLIS.ALGO2@GMAIL.COM
MILA and DIRO, Université de Montréal
Thomas Pumir PUMIR.THOMAS@GMAIL.COM
Princeton University
Nicolas Boumal NICOLASBOUMAL@GMAIL.COM
EPFL

Abstract

Quasi-Newton techniques approximate the Newton step by estimating the Hessian using the so-called secant equations. Some of these methods compute the Hessian using several secant equations but produce non-symmetric updates. Other quasi-Newton schemes, such as BFGS, enforce symmetry but cannot satisfy more than one secant equation. We propose a new type of quasi-Newton symmetric update using several secant equations in a least-squares sense. Our approach generalizes and unifies the design of quasi-Newton updates and satisfies provable robustness guarantees.

1. Introduction

We consider second-order methods for unconstrained minimization of a smooth, possibly non-convex function $f\colon \mathbb{R}^d \to \mathbb{R}$. Despite a locally quadratic convergence rate, the well-known Newton method iterate

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\left[\nabla^2 f(\mathbf{x}_k)\right]^{-1}\nabla f(\mathbf{x}_k). \tag{1}$$

is not suitable for large-scale problems, in part because it requires solving a $d \times d$ linear system involving the Hessian at every iteration. To address this issue, quasi-Newton algorithms replace the update rule (1) by

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k - h_k\mathbf{B}_k^{-1}\nabla f(\mathbf{x}_k) \quad \text{or}\\ \mathbf{x}_{k+1} &= \mathbf{x}_k - h_k\mathbf{H}_k \ \ \nabla f(\mathbf{x}_k),\end{aligned} \tag{2}$$

where $\mathbf{B}_k \approx \nabla^2 f(\mathbf{x}_k)$ and $\mathbf{H}_k \approx \left[\nabla^2 f(\mathbf{x}_k)\right]^{-1}$ are approximations of the Hessian and its inverse (respectively) at $\mathbf{x}_k$. Choosing the right approximation for $\mathbf{H}_k$ and $\mathbf{B}_k$ has drawn considerable attention in the optimization literature. We can cite the DFP update [10], Broyden method [6], SR1 update [8] or the well-known BFGS method [7], [16], [17] [31]. In general, those methods estimate a matrix $\mathbf{B}_k$ or $\mathbf{H}_k$ satisfying the *secant* equation

$$\begin{aligned}(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})) &= \mathbf{B}_k(\mathbf{x}_k - \mathbf{x}_{k-1}) \text{ or}\\ \mathbf{H}_k(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})) &= \quad (\mathbf{x}_k - \mathbf{x}_{k-1}),\end{aligned} \tag{3}$$

0. *Equal contribution

then perform the quasi-Newton step (2). It is also possible to satisfy *several* secant equations. For instance, the multisecant Type-I and Type-II Broyden methods [15] find a matrix $\mathbf{B}_k$ or $\mathbf{H}_k$ satisfying a block of secants: for a memory size m and for $i = k - m + 1 \dots k$,

$$[\nabla f(\mathbf{x}_i) - \nabla f(\mathbf{x}_{i-1})] = \mathbf{B}_k[\mathbf{x}_i - \mathbf{x}_{i-1}] \text{ or}$$
$$\mathbf{H}_k[\nabla f(\mathbf{x}_i) - \nabla f(\mathbf{x}_{i-1})] = \quad [\mathbf{x}_i - \mathbf{x}_{i-1}],$$

Such multisecant updates are called *block Broyden methods*, and have been considered by Fang and Saad [15], who showed an equivalence between Broyden updates and Anderson acceleration [2]. Unfortunately, the resulting approximations of the Hessian are not symmetric.

By contrast, other methods like BFGS and DFP enforce the symmetry of the update. Their main drawback is that, for generic objectives, they can only satisfy *one* secant equation. The major limitation with single-secant update is the high dependence in the step size[28]. Indeed, while BFGS and DFP enjoy an optimal convergence rate on quadratics when using an exact line-search [27], Powell [28] showed that with *unitary* step size, theses updates converge particularly slowly on a simple quadratic function with just two variables. Moreover, it was also observed that BFGS updates are sensitive to gradient noise, and designing quasi-Newton methods for stochastic algorithm is still a challenge [3–5, 9].

Unfortunately, except for quadratic functions [29], it is usually impossible to find a symmetric matrix that satisfies more than one secant equation. Moreover, line search has been shown to be computationally expensive. Finally, stabilisation procedure for stochastic BFGS usually requires a growing batch size to reduce the gradient noise, making it unpractical in many applications.

1.1. Notation

We use boldface small letters, like $\mathbf{x}$, to refer to vectors and boldface capital letters, like $\mathbf{A}$, for matrices. We use d to refer to the *dimension* of the problem, and m for the *memory* of the algorithm (we will see later that m is the number of secant equations). For a function $f\colon \mathbb{R}^d \to \mathbb{R}$, its gradient and Hessian at x are denoted by $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$ respectively. Consistently with the notations in the literature, we use $\mathbf{H}$ to denote an approximation of the *inverse* of the Hessian, while we use $\mathbf{B}$ to denote an approximation of the Hessian. We denote the usual *Frobenius* norm as $\|\cdot\|$. Moreover, for any square matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ and any positive definite matrix $\mathbf{W} \in \mathbb{R}^{d\times d}$, we define the norm $\|\mathbf{A}\|_{\mathbf{W}}$ as

$$\|\mathbf{A}\|_{\mathbf{W}} = \|\mathbf{W}^{\frac{1}{2}}\mathbf{A}\mathbf{W}^{\frac{1}{2}}\|. \tag{4}$$

We often use the matrices $\mathbf{X} \in \mathbb{R}^{d\times m}$ and $\mathbf{G}$, that concatenates the iterates and their gradients as follow,

$$\mathbf{X} = [\mathbf{x}_i, \dots, \mathbf{x}_{i+m}], \quad \mathbf{G} = [\nabla f(\mathbf{x}_i), \dots, \nabla f(\mathbf{x}_{i+m})].$$

Also, we define $\mathbf{C}$, and $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ as

$$\Delta\mathbf{X} = \mathbf{X}\mathbf{C}, \qquad \Delta\mathbf{G} = \mathbf{G}\mathbf{C},$$

where $\mathbf{C} \in \mathbb{R}^{m+1\times m}$ is a matrix of rank $m-1$ such that $\mathbf{1}_{m+1}^T\mathbf{C} = 0$, $\mathbf{1}_{m+1}$ being a vector of size $m+1$ full of ones. Typically, $\mathbf{C}$ is the column-difference matrix

$$\mathbf{C} = \begin{bmatrix} -1 & 0 & 0 & \dots \\ 1 & -1 & 0 & \dots \\ 0 & 1 & 1 & \dots \\ & \ddots & \ddots & \\ & & 1 & -1 \\ & & 0 & 1 \end{bmatrix} \Rightarrow \Delta\mathbf{X} = \begin{bmatrix} \mathbf{x}_{i+1} - \mathbf{x}_i \\ \vdots \\ \mathbf{x}_{i+m} - \mathbf{x}_{i+m-1} \end{bmatrix},$$

and similarly for $\Delta\mathbf{G}$. See Section A for backgrounds on single-secant and multi-secant updates.

1.2. Contributions

The previous section shows that quasi-Newton methods approximate the Hessian. However, the two methods do it in very different ways that seems incompatible given the work of Schnabel [29]. Despite their difference, they shares similarities, such as the idea of secant equations. This leads to the following questions:

Is it possible to design a generalized framework for quasi-Newton updates, including for instance Broyden's, DFP and BFGS schemes? Can Symmetric and Multisecant techniques be combined into a symmetric multisecant update?

Our work proposes a positive answer to these questions trough the following contributions. (1) We propose a general framework that models and generalizes previous quasi-Newton updates. (2) We derive new quasi-Newton update rules (Algorithm 1), which are symmetric and take into account *several secant equations*. The bottleneck is a (economic size) Singular Value Decomposition (SVD), whose complexity is linear in the dimension of the problem, therefore comparable to other quasi-Newton methods. (3) We show the optimality of the convergence rate of any multisecant quasi-Newton update build using our framework, on quadratic functions *without line search*. This improves over the BFGS and DFP updates as they are inefficient with unitary step size on quadratics [28], and suboptimal if exact line-search is not used. (4) We introduce novel *robust updates*, that provably reduce the sensitivity to the noise of our quasi-Newton schemes. This robustness property is a direct consequence of considering several secant equations at once.

2. Generalization of Quasi-Newton

We have seen in the previous section two different qN updates, one that focuses on the *symmetry* of the estimate, the other on the number of satisfied *secant equations*. In this section, we propose an unified framework to design existing and new qN schemes.

2.1. Generalized (Multi-)Secant Equations

The central part of qN methods is the secant equation. The idea follows from the linearization of the gradient of the objective function. Indeed, consider the function $f(\mathbf{x})$, supposed to be smooth, strongly convex and twice differentiable. In such case, the linearization of its gradient around the minimum $\mathbf{x}_\star$ reads

$$\nabla f(\mathbf{x}) \approx \underbrace{\nabla f(\mathbf{x}_\star)}_{=0} + \nabla^2 f(\mathbf{x}_\star)(\mathbf{x} - \mathbf{x}_\star). \tag{5}$$

Therefore, like in the case of the Newton step, assuming this approximation equal we perform the step

$$\mathbf{x} - [\nabla^2 f(\mathbf{x}^\star)]^{-1} \nabla f(\mathbf{x}) \approx \mathbf{x}_\star.$$

Unfortunately, we do not have access to the matrix $[\nabla^2 f(\mathbf{x}^\star)]$ as we do not know $\mathbf{x}_\star$. Moreover, solving the linear system $[\nabla^2 f(\mathbf{x}^\star)]^{-1} \nabla f(\mathbf{x})$ may be costly in the case where d is big.

To avoid such problems, consider a sequence $\{\mathbf{x}_0, \dots, \mathbf{x}_m\}$ of points for which we have computed their gradient. In such case, (5) reads

$$\mathbf{G} = \nabla^2 f(\mathbf{x}_\star)(\mathbf{X} - \mathbf{X}_\star),$$

where $\mathbf{X}_\star = \mathbf{x}_\star \mathbf{1}_{m+1}^T$, i.e., the matrix concatenating $m+1$ copies of the vector $\mathbf{x}_\star$. Matrices $\mathbf{X}$ and $\mathbf{G}$ are defined in section 1.1.

Ideally, the estimate $\mathbf{B}$ of the Hessian, or the estimate of its inverse $\mathbf{H}$, has to satisfy the condition

$$\mathbf{G} = \mathbf{B}(\mathbf{X} - \mathbf{X}_\star), \quad \text{or} \quad \mathbf{HG} = (\mathbf{X} - \mathbf{X}_\star).$$

However, the dependency in $\mathbf{x}_\star$ makes the problem of estimating $\mathbf{B}$ or $\mathbf{H}$ intractable. To remove this problematic dependency, consider a matrix $\mathbf{C} \in \mathbb{R}^{m+1 \times m}$ of rank m such that $\mathbf{1}_{m+1}^T \mathbf{C} = 0$ (see Section 1.1 for an example). After multiplying on the right, we simplify $\mathbf{X}_\star \mathbf{C} = 0$ and we obtain the *multisecant equations*

$$\Delta\mathbf{G} = \mathbf{B}\Delta\mathbf{X}, \quad \text{or} \quad \mathbf{H}\Delta\mathbf{G} = \Delta\mathbf{X}, \tag{6}$$

where $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ are defined in Section 1.1. In the specific case where we have only one secant equation, (6) corresponds exactly to the standard secant equation in (11). In the case where $\mathbf{C}$ is the column-difference operator, we obtain the multisecant equations usually used in multisecant Broyden methods.

2.2. Regularization and Constraints

The matrix $\mathbf{B}$ (Broyden Type-I and DFP updates) and $\mathbf{H}$ (Broyden Type-II and BFGS) are selected so as to minimize its distance w.r.t. a reference matrix, called $\mathbf{B}_{\text{ref}}/\mathbf{H}_{\text{ref}}$, as shown in (13). In the case where there is only a sequence of single secant equations, the reference matrix is taken as being the previous estimate, with an arbitrary initialization. In the case of multisecant update, the reference matrix is arbitrary. Moreover, in the case of DFP and BFGS, we have in addition a *symmetry* constraint, restraining even more the search space for the estimate of the Hessian. For simplicity, we will consider only the type-I update here, i.e., the estimate $\mathbf{B}$. The formulation for estimate $\mathbf{H}$ can be easily derived by swapping $\Delta\mathbf{G}$ and $\Delta\mathbf{X}$.

The intuition behind the regularization term is due to the number of degrees of freedom in the problem. Let us recall the secant equation,

$$\mathbf{B}\Delta\mathbf{X} = \Delta\mathbf{G}$$

This secant equation defines the behavior of the operator $\mathbf{B}$, mapping from $\mathbf{span}\{\Delta\mathbf{G}\}$ to $\mathbf{span}\{\Delta\mathbf{X}\}$. However, the dimension of these two spans is as most $m < d$. This means we have to define the behavior of $\mathbf{B}$ *outside* of $\mathbf{span}\{\Delta\mathbf{X}\}$ and $\mathbf{span}\{\Delta\mathbf{G}\}$, i.e., from $\mathbf{span}\{\Delta\mathbf{G}\}^\perp$ to $\mathbf{span}\{\Delta\mathbf{X}\}^\perp$.

Since $\mathbf{B}$ outside the span is not driven by the secant equations, we have to define an operator $\mathbf{B}_{\text{ref}}$, defining the default behavior of $\mathbf{B}$ outside the span of secant equations. This means that, in the case where $\mathbf{B}$ satisfies exactly the secant equations, then $\mathbf{B}$ can be written as

$$\begin{aligned}\mathbf{B} =& [\Delta\mathbf{G}\Delta\mathbf{X}^\dagger]\mathbf{P} \\ &+ [\text{depends on } \mathbf{B}_{\text{ref}} \text{ and constraints}](\mathbf{I} - \mathbf{P}),\end{aligned}$$

where $\mathbf{P}$ is the projector to the span of $\Delta\mathbf{X}$, and $\Delta\mathbf{X}^\dagger$ is the Moore-Penrose pseudo-inverse of $\Delta\mathbf{X}$. Indeed, in this case

$$\mathbf{B}\Delta\mathbf{X} = [\Delta\mathbf{G}\Delta\mathbf{X}^\dagger]\mathbf{P}\Delta\mathbf{X} + [\ldots](\mathbf{I} - \mathbf{P})\Delta\mathbf{X}.$$

Since $\mathbf{P}\Delta\mathbf{X} = \Delta\mathbf{X}$, thus $(\mathbf{I} - \mathbf{P})\Delta\mathbf{X} = 0$ (by construction of $\mathbf{P}$), and $\Delta\mathbf{G}\Delta\mathbf{X}^\dagger\Delta\mathbf{X} = \Delta\mathbf{G}$ by definition of the Moore-Penrose pseudo-inverse and because $\Delta\mathbf{X}$ is assumed to be full column rank, we have that $\mathbf{B}$ satisfies the secant equation.

The way $\mathbf{B}$ behaves outside the span is thus driven by the constraints and initialization on the matrix. To make a parallel with machine learning problem, this term can be seen as the "generalization" term.

There are two common ways to define $\mathbf{B}$ outside the span of the secant equations, trough regularization and constraints. Consider the regularisation function $\mathcal{R}(\cdot, \mathbf{B}_{\text{ref}})$, assumed to be strictly-convex and with minimum attained at $\mathbf{B}_{\text{ref}}$, and the convex set of constraints $\mathcal{C}$. We can thus write the qN update estimation problem as

$$\min_{\mathbf{B}\in\mathcal{C}} \mathcal{R}(\mathbf{B}, \mathbf{B}_{\text{ref}}) \quad \text{subject to } \mathbf{B}\Delta\mathbf{X} = \Delta\mathbf{G}. \tag{7}$$

This approach generalizes the way we define qN updates. Indeed, for instance, we recover DFP by setting $\mathcal{R} = \|\mathbf{B} - \mathbf{B}_{\text{ref}}\|_{\mathbf{W}^{-1}}$, $\mathcal{C} = \mathbb{S}^{d\times d}$ (the set of symmetric matrices), $m = 1$ and $\mathbf{B}_{\text{ref}} = \mathbf{B}_{k-1}$ in (7). We also recover the Type-I Broyden method by setting $\mathcal{R} = \|\mathbf{B} - \mathbf{B}_{\text{ref}}\|_F$ and $\mathcal{C} = \mathbb{R}^{d\times d}$.

2.3. Generalized QN Update

A natural extension, given the updates of DFP/BFGS and multisecant Broyden, would be the symmetric multi-secant update. This update would read, for an arbitrary regularization function,

$$\min_{\mathbf{B}\in\mathbb{S}^{d\times d}} \mathcal{R}(\mathbf{B}, \mathbf{B}_{\text{ref}}) \quad \text{subject to } \mathbf{B}\Delta\mathbf{X} = \Delta\mathbf{G}.$$

In the case where $m > 1$, this multisecant technique seems promising as it combines the advantages of multisecant Broyden and symmetric updates.

Unfortunately, the system of equations and the constraints in problem (7) are of the form $\mathbf{B}\Delta\mathbf{X} = \Delta\mathbf{G}$. Assuming $\Delta\mathbf{X}, \Delta\mathbf{G}$ have full column rank, these equations always have a solution $\mathbf{B}$, there exists a *symmetric* solution *if and only if* $\Delta\mathbf{X}^T\Delta\mathbf{G}$ is symmetric [21, 29].

When $\Delta\mathbf{X}^T\Delta\mathbf{G}$ is symmetric, Schnabel [29] derived a Multisecant BFGS-type update rule. This assumption indeed holds for quadratic objectives, but not for general objective functions when $m \geq 2$, that is, when we consider more than one secant condition [29, Example 3.1]. Hence, a naive extension of symmetric quasi-Newton update leads to unfeasible problems.

To tackle the problem of unfeasible updates, we can relax the constraint on the secant equations by a *loss function* $\mathcal{L}(\cdot, \Delta\mathbf{X}, \Delta\mathbf{G})$. We finally end up with the *generalized (type-I and type-II) qN update*

$$\mathbf{B}_k = \lim_{\lambda\to 0} \arg\min_{\mathbf{B}\in\mathcal{C}} \mathcal{L}(\mathbf{B}, \Delta\mathbf{X}, \Delta\mathbf{G}) + \frac{\lambda}{2}\mathcal{R}(\mathbf{B}, \mathbf{B}_{\text{ref}}) \tag{GQN-I}$$

$$\mathbf{H}_k = \lim_{\lambda\to 0} \arg\min_{\mathbf{H}\in\mathcal{C}} \mathcal{L}(\mathbf{H}, \Delta\mathbf{G}, \Delta\mathbf{X}) + \frac{\lambda}{2}\mathcal{R}(\mathbf{H}, \mathbf{H}_{\text{ref}}) \tag{GQN-II}$$

where we assume that $\mathcal{L}$ and $\mathcal{R}$ strictly convex, and sufficiently simple to have an explicit formula. The limits here simply states that we first minimize the loss function, then with the remaining degree of freedom we minimize the regularization term. In the case where the update (7) is feasible, then (GQN-I)/(GQN-II) and (7) are equivalent.

2.4. Preconditioning

As shown for instance in DFP and BFGS, it is common to use a preconditionner to reduce the dependence of the update to the units of the problem. We give here the example for type-II update. The type-I follows immediately, as it suffices to consider $\mathbf{W}^{-1}$ instead of $\mathbf{W}$.

The idea of preconditioning is, instead of considering $\mathbf{H}$, we set

$$\mathbf{M} = \mathbf{W}^{(1-\alpha)}\mathbf{H}\mathbf{W}^{\alpha},$$

where $\mathbf{W}$ is a symmetric, positive definite matrix, which ideally has the same units as the *Hessian* of the function f. For example, in BFGS, $\mathbf{W}$ is *any* matrix such that $\mathbf{W}\Delta\mathbf{X} = \Delta\mathbf{G}$, which always

exists in the case where $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ are vectors. Ideally, the preconditionner cancels the units in the update rules, i.e., $\mathbf{W}$ has to have the same units as the Hessian.

In the case where we consider a preconditioner,

$$\mathbf{M}\mathbf{W}^{-\alpha}\Delta\mathbf{X} = \mathbf{W}^{1-\alpha}\Delta\mathbf{G}, \quad \mathbf{M}_{\text{ref}} = \mathbf{W}^{\alpha-1}\mathbf{H}_{\text{ref}}\mathbf{W}^{-\alpha}.$$

Now, the problems becomes the *type-II Preconditioned Generalized Quasi-Newton* update

$$\underset{\mathbf{M}\in\tilde{\mathcal{C}}}{\arg\min} \quad \mathcal{L}(\mathbf{M}, \mathbf{W}^{-\alpha}\Delta\mathbf{X}, \mathbf{W}^{(1-\alpha)}\Delta\mathbf{G}) + \frac{\lambda}{2}\mathcal{R}(\mathbf{M}, \mathbf{M}_{\text{ref}}) \tag{PGQN-II}$$

where $\tilde{\mathcal{C}} = \mathbf{W}^{(1-\alpha)}\mathcal{C}\mathbf{W}^{\alpha}$, i.e., the image of the constraint after application of the preconditionner. To retrieve the update $\mathbf{H}$, it suffices to solve

$$\mathbf{H} = \mathbf{W}^{-(1-\alpha)}\mathbf{M}\mathbf{W}^{-\alpha}.$$

2.5. Rate of Convergence on Quadratics

The generalized qN methods (GQN-I) and (GQN-II) are optimal on quadratics under mild assumptions, in the sense that their performance is comparable to conjugate gradient descent, as shown in the theorem below. The only requirement is that the loss, regularization, initialization and constraint set of problems (PGQN-II) and (PGQN-II) create updates that are regular enough.

Theorem 1 *Consider any multisecant quasi-Newton method (GQN-I) or (GQN-II) with unit step-size and infinite memory of the form*

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{H}_k\nabla f(\boldsymbol{x}_k), \text{ or } \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{B}_k^{-1}\nabla f(\boldsymbol{x}_k) \tag{8}$$

where f is the quadratic form $(\boldsymbol{x}-\boldsymbol{x}_\star)^T\frac{\boldsymbol{Q}}{2}(\boldsymbol{x}-\boldsymbol{x}_\star)$ for some $Q \succ 0$, and $\boldsymbol{B}$ ($\boldsymbol{H}$) satisfies exactly the secant equations. If the update (8) is a Preconditioned first-order method, i.e.,

$$\boldsymbol{x}_{k+1} \in \boldsymbol{x}_0 + \tilde{\boldsymbol{B}}^{-1}\,(\tilde{\boldsymbol{H}})\ \mathbf{span}\{\nabla f(\boldsymbol{x}_0), \dots, \nabla f(\boldsymbol{x}_k)\}$$

where $\tilde{\boldsymbol{B}}^{-1}$ ($\tilde{\boldsymbol{H}}$) is non-singular, then $\boldsymbol{x}_k = \boldsymbol{x}_\star$ if $k \geq d+1$, otherwise the method converges at rate

$$\|\nabla f(\boldsymbol{x}_k)\| \leq \textit{constant} \cdot \left(\tfrac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^k \|\nabla f(\boldsymbol{x}_0)\|,$$

Where κ is the inverse of the condition number of $\boldsymbol{Q}$. The constant is usually smaller when $\boldsymbol{B}_{ref}$ ($\boldsymbol{H}_{ref}$) is a good approximation of $\boldsymbol{Q}$ ($\boldsymbol{Q}^{-1}$).

Notice that the multisecant Broyden update (13) satisfies the assumptions of Theorem (1) if $\mathbf{B}_{\text{ref}}$ or $\mathbf{H}_{\text{ref}}$ are scaled identity. We do the example for the Broyden Type-II method. Indeed,

$$\mathbf{H} = \Delta\mathbf{X}\Delta\mathbf{G}^\dagger + \mathbf{H}_{\text{ref}}(\mathbf{I} - \Delta\mathbf{G}\Delta\mathbf{G}^\dagger).$$

After multiplication by the gradient $\boldsymbol{g}_k$, if $c \overset{\text{def}}{=} \Delta\mathbf{G}^\dagger\boldsymbol{g}_k$ and $\mathbf{H}_{\text{ref}} = \beta\mathbf{I}$,

$$\mathbf{H}\boldsymbol{g}_k = \underbrace{\Delta\mathbf{X}c}_{\in\mathbf{span}} + \beta\underbrace{(\boldsymbol{g}_k - \Delta\mathbf{G}c)}_{\in\mathbf{span}} \in \mathbf{span}\{\nabla f(x_0), \dots\}$$

The same hold for the multisecant DFP or BFGS for quadratic functions if we use the update from [29].

We have now a generic form of qN update, but it raises some important questions. What practical losses and regularization functions should we use, and What happen if λ does not go to zero? The next section addresses the first point by giving an example that extends (limited memory) DFP and multi-secant Broyden methods. Then, we analyse the robustness of the method when λ is non-zero.

3. Robust Symmetric Multisecant Updates

We now extend the BFGS and multisecant Broyden method into the type-II Symmetric Multisecant Update (SMU-II) below, solving the problem (PGQN-II) in the special case where the loss and the regularization are Frobenius norms. For simplicity, we do not consider any preconditionner here. The method reads

$$\mathbf{H}_k = \underset{\mathbf{H}=\mathbf{H}^T}{\arg\min} \|\mathbf{H}\Delta\mathbf{X} - \Delta\mathbf{G}\|_F^2 + \frac{\lambda}{2}\|\mathbf{H} - \mathbf{H}_{\text{ref}}\|_F^2, \tag{SMU-II}$$

3.1. Explicit Formula

This section addresses the problem of solving (SMU-II) efficiently. This problem is an extension of the *symmetric Procrusted problem* from [23]. Indeed, [23] solves the problem

$$\min_{\mathbf{Z}=\mathbf{Z}^T} \|\mathbf{Z}\mathbf{A} - \mathbf{D}\|_F,$$

where $\mathbf{A}$ and $\mathbf{D}$ are $\mathbb{R}^{n\times m}$ matrices, where $m > n$. In our case, we have $m \ll n$, and an extra regularization term, that makes the update formula more complicated. Fortunately, the matrix-vector multiplication $\mathbf{Z}\mathbf{v}$ can be done efficiently even in our case, the bottleneck being the computation of the SVD of a thin matrix.

The next theorem details the explicit formula to compute $\mathbf{M}_k$ (and its inverse if one wants to use a type-I method).

Theorem 2 *Consider the Regularized Symmetric Procrustes (RSP) problem*

$$\boldsymbol{Z}_\star = \underset{\boldsymbol{Z}=\boldsymbol{Z}^T}{\arg\min} \|\boldsymbol{Z}\boldsymbol{A} - \boldsymbol{D}\|_F^2 + \frac{\lambda}{2}\|\boldsymbol{Z} - \boldsymbol{Z}_{ref}\|_F^2, \tag{RSP}$$

where $\boldsymbol{Z}_{ref}$ is symmetric (otherwise, take the symmetric part of $\boldsymbol{Z}_{ref}$), $\boldsymbol{Z}$, $\boldsymbol{Z}_{ref} \in \mathbb{R}^{d\times d}$, and $\boldsymbol{A}$, $\boldsymbol{D} \in \mathbb{R}^{d\times m}$, $m \leq d$. Then, the solution $\boldsymbol{Z}_\star$ is given by

$$\boldsymbol{Z}_\star = \boldsymbol{V}_1\boldsymbol{Z}_1\boldsymbol{V}_1^T + \boldsymbol{V}_1\boldsymbol{Z}_2 + \boldsymbol{Z}_2^T\boldsymbol{V}_1^T + (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Z}_{ref}(\boldsymbol{I} - \boldsymbol{P}) \tag{Sol-RSP}$$

where

$$\begin{aligned}
[\boldsymbol{U}, \Sigma, \boldsymbol{V}_1] &= \boldsymbol{SVD}(\boldsymbol{A}^T, \texttt{'econ'}),\\
\boldsymbol{Z}_1 &= \boldsymbol{S} \odot \left[\boldsymbol{V}_1^T\left(\boldsymbol{A}\boldsymbol{D}^T + \boldsymbol{D}\boldsymbol{A}^T + \lambda\boldsymbol{Z}_{ref}\right)\boldsymbol{V}_1\right],\\
\boldsymbol{S} &= \frac{1}{\Sigma^2\boldsymbol{1}\boldsymbol{1}^T + \boldsymbol{1}\boldsymbol{1}^T\Sigma^2 + \lambda\boldsymbol{1}\boldsymbol{1}^T},\\
\boldsymbol{P} &= \boldsymbol{V}_1\boldsymbol{V}_1^T,\\
\boldsymbol{Z}_2 &= (\Sigma^2 + \lambda\boldsymbol{I})^{-1}\boldsymbol{V}_1^T(\boldsymbol{A}\boldsymbol{D}^T + \lambda\boldsymbol{Z}_{ref})(\boldsymbol{I} - \boldsymbol{P})
\end{aligned}$$

The fraction in $\boldsymbol{S}$ stands for the element-wise inversion (Hadamard inverse). The inverse $\boldsymbol{Z}_\star^{-1}$ reads

$$\begin{aligned}
\boldsymbol{Z}_\star^{-1} &= \boldsymbol{E}\left(\boldsymbol{Z}_1 - \boldsymbol{Z}_2\boldsymbol{Z}_{ref}^{-1}\boldsymbol{Z}_2^T\right)^{-1}\boldsymbol{E}^T + (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Z}_{ref}^{-1}(\boldsymbol{I} - \boldsymbol{P})\\
\boldsymbol{E} &= \boldsymbol{V}_1 - (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Z}_{ref}^{-1}\boldsymbol{Z}_2^T.
\end{aligned} \tag{Inv-RSP}$$

The type-I update uses the matrix $\mathbf{Z}_\star^{-1}$, using $\mathbf{A} = \Delta\mathbf{X}$ and $\mathbf{D} = \Delta\mathbf{G}$. The type-II uses instead $\mathbf{Z}_\star$, with $\mathbf{A} = \Delta\mathbf{G}$ and $\mathbf{D} = \Delta\mathbf{X}$. the next proposition shows the complexity of performing one matrix-vector multiplication with $\mathbf{Z}_\star$ and its inverse. The bottleneck of the method is the SVD of a $\mathbb{R}^{m\times d}$ matrix, whose complexity is $O(m^2 d)$, thus linear in the dimension.

Proposition 3 *The complexity of evaluating $\mathbf{Z}_\star \boldsymbol{v}$ and $\mathbf{Z}_\star^{-1}\boldsymbol{v}$ is $O(m^2 d)$, assuming $m \ll d$ and that the complexity of $\mathbf{Z}_{ref}\boldsymbol{v}$ and $\mathbf{Z}_{ref}^{-1}\boldsymbol{v}$ is at most $O(m^2 d)$.*

3.2. Robustness

The symmetric multisecant update can be used in two different modes, one that let $\lambda \to 0$, the other, biased but more robust, that set $\lambda > 0$.

The update formula is slightly simpler when $\lambda = 0$. However, due to the presence of matrix inversion, this may leads to instability problem is some cases, similarly to the BFGS method when

$$(\mathbf{x}_{k+1} - \mathbf{x}_k)^T(\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)) \approx 0,$$

i.e., when the step and difference of gradients are close to be orthogonal. In BFGS, such problem is tackled by a filtering step, discarding the update if the scalar product goes below some threshold. Unfortunately, when the gradient is corrupted by some noise, the impact on the BFGS update can be huge.

In the case where $\lambda > 0$, we can show that our update is robust when $\mathbf{A}$ and $\mathbf{D}$ are corrupted.

Proposition 4 *Let $\mathbf{Z}_\star(\lambda)$ be defined as the solution of* (Sol-RSP) *for some λ, and $\mathbf{Z}_\star(\lambda) = \lim_{\lambda\to 0} \mathbf{Z}_\lambda$. Let $\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{C}}$ be a corrupted version of $\boldsymbol{A}$ and $\boldsymbol{C}$ where*

$$\|\boldsymbol{A} - \tilde{\boldsymbol{A}}\| \leq \delta_{\boldsymbol{A}}, \quad \|\boldsymbol{D} - \tilde{\boldsymbol{D}}\| \leq \delta_{\boldsymbol{D}}.$$

Finally, let $\tilde{\mathbf{Z}}_\star(\lambda)$ be the solution of (Sol-RSP) *using $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{C}}$. Then, we have*

$$\|\tilde{\mathbf{Z}}_\star(\lambda) - \mathbf{Z}_\star(0)\| \leq \underbrace{\|\mathbf{Z}_\star(\lambda) - \mathbf{Z}_\star(0)\|}_{\textit{Bias}} + \underbrace{\|\tilde{\mathbf{Z}}_\star(\lambda) - \mathbf{Z}_\star(\lambda)\|}_{\textit{Stability}},$$

where

$$\|\mathbf{Z}_\star(\lambda) - \mathbf{Z}_\star(0)\| \leq \frac{\lambda\|\mathbf{Z}_\star(0) - \mathbf{Z}_{ref}\|}{\sigma_{\min}^2(\boldsymbol{A}) + \lambda}, \tag{9}$$

$$\|\tilde{\mathbf{Z}}_\star(\lambda) - \mathbf{Z}_\star(\lambda)\| \leq \mathcal{O}\left(\frac{1}{\lambda}(\|\tilde{\boldsymbol{A}}\| + \|\tilde{\boldsymbol{D}}\|)^2\right). \tag{10}$$

This suggests that λ should satisfy a trade-off to achieve the best performing approximation. Notice that when $\lambda = 0$ in the noise-less case, we recover the optimal $\mathbf{Z}_\star$, and when $\lambda \to \infty$, we have $\mathbf{Z}_\star = \mathbf{Z}_{\text{ref}}$.

Our result is called *robust* as we can bound the maximum perturbation in any scenario, as long as $\lambda > 0$. This is *not* the case in the analysis of [23], whose main assumption is $\delta_{\mathbf{A}} \leq \sigma_{\min}(\mathbf{A})$, where $\sigma_{\min}$ is the smallest non-zero singular value of $\mathbf{A}$. This condition is extremely restrictive, as we obverse that $\sigma_{\min}$ is of the order of the epsilon machine even for reasonable m (like $m = 5$).

Since the singular values of $\mathbf{A}$ are, in practice, often small, it is always recommended to set a small λ. We will show latter, in the numerical experiments, that even for quadratic function (i.e., in the "perturbation-free regime"), a small value of λ drastically change the final result, as this makes the method robust to numerical noise.

Scaling of λ. The parameter λ has to be scaled w.r.t. the problem input. It is clear, from Theorem 12, that the role of λ is to regularize the matrix inversion by lower-bounding the eigenvalues of the inverted matrix. Therefore, we advice to set $\lambda = \bar{\lambda}\|\mathbf{A}^T\mathbf{A}\|_2$, i.e., proportional to $\|\mathbf{A}^T\mathbf{A}\|_2$. This way, assuming $\sigma_{\min}$ small, the conditioning of $(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}$ is upper-bounded by $1 + 1/\bar{\lambda}$.

References

[1] N. Agarwal, B. Bullins, and E. Hazan. Second-order Stochastic Optimization for Machine Learning in Linear Time. *J. Mach. Learn. Res.*, 18(1):4148–4187, January 2017. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=3122009.3176860.

[2] Donald G Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560, 1965.

[3] Albert S Berahas, Raghu Bollapragada, and Jorge Nocedal. An investigation of newton-sketch and subsampled newton methods. *Optimization Methods and Software*, pages 1–20, 2020.

[4] Raghu Bollapragada, Dheevatsa Mudigere, Jorge Nocedal, Hao-Jun Michael Shi, and Ping Tak Peter Tang. A progressive batching l-bfgs method for machine learning. *arXiv preprint arXiv:1802.05374*, 2018.

[5] Raghu Bollapragada, Richard H Byrd, and Jorge Nocedal. Exact and inexact subsampled newton methods for optimization. *IMA Journal of Numerical Analysis*, 39(2):545–578, 2019.

[6] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19(92):577–593, 1965.

[7] C. G. Broyden. The Convergence of a Class of Double-Rank Minimization Algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 09 1970. doi: 10.1093/imamat/6.3.222.

[8] Richard H. Byrd, Humaid Fayez Khalfan, and Robert B. Schnabel. Analysis of a symmetric rank-one trust region method. *SIAM J. on Optimization*, 6(4):1025–1039, April 1996. ISSN 1052-6234. doi: 10.1137/S1052623493252985. URL https://doi.org/10.1137/S1052623493252985.

[9] Richard H. Byrd, S. L. Hansen, Jorge Nocedal, and Yoram Singer. A Stochastic Quasi-Newton Method for Large-scale Optimization. *SIAM Journal on Optimization*, 26:1008–1031, 2016.

[10] W.C. Davidon. Variable metric method for minimization. *Technical Report ANL 5990 (revised), Argonne National Laboratory, Argonne, Il*, 1959.

[11] W.C. Davidon. Variable metric method for minimization. *SIAM Journal on Optimization*, 1:1–17, 1991.

[12] R. Dembo, S. Eisenstat, and T. Steihaug. Inexact Newton Methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982. doi: 10.1137/0719025. URL https://doi.org/10.1137/0719025.

[13] Ron S. Dembo and Trond Steihaug. Truncated-Newton algorithms for large-scale unconstrained optimization. *Mathematical Programming*, 26(2):190–212, Jun 1983. ISSN 1436-4646. doi: 10.1007/BF02592055. URL https://doi.org/10.1007/BF02592055.

[14] Murat A. Erdogdu and Andrea Montanari. Convergence rates of sub-sampled newton methods. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 3052–3060, Cambridge, MA, USA, 2015. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969442.2969580.

[15] Haw-ren Fang and Yousef Saad. Two classes of multisecant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications*, 16(3):197–221, 2009.

[16] R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317?322, 1970.

[17] D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109):23?26, 1970.

[18] G.H. Golub and C.F. Van Loan. *Matrix computations*, volume 3 of *Johns Hopkins Studies in the Mathematical Sciences*. Johns Hopkins University Press, 4th edition, 2012. doi: 10.1137/0720042.

[19] Robert M. Gower, Donald Goldfarb, and Peter Richtárik. Stochastic Block BFGS: Squeezing More Curvature out of Data. In *ICML*, 2016.

[20] Robert Mansel Gower and Jacek Gondzio. Action constrained quasi-newton methods. *arXiv preprint arXiv:1412.8045*, 2014.

[21] F.J. Henk Don. On the symmetric solutions of a linear matrix equation. *Linear Algebra and its Applications*, 93:1–7, 07 1987. doi: 10.1016/S0024-3795(87)90308-9.

[22] P. Hennig. Probabilistic interpretation of linear solvers. *SIAM Journal on Optimization*, 25(1):234–260, 2015. doi: 10.1137/140955501. URL https://doi.org/10.1137/140955501.

[23] N. J. Higham. The symmetric Procrustes problem. *BIT*, 28, 03 1988. doi: 10.1007/BF01934701.

[24] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, Aug 1989. ISSN 1436-4646. doi: 10.1007/BF01589116. URL https://doi.org/10.1007/BF01589116.

[25] Aryan Mokhtari and Alejandro Ribeiro. Global Convergence of Online Limited Memory BFGS. *Journal of Machine Learning Research*, 16:3151–3181, 2015. URL http://jmlr.org/papers/v16/mokhtari15a.html.

[26] Philipp Moritz, Robert Nishihara, and Michael Jordan. A Linearly-Convergent Stochastic L-BFGS Algorithm. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 249–258, Cadiz, Spain, 09–11 May 2016. PMLR. URL http://proceedings.mlr.press/v51/moritz16.html.

[27] J. Nocedal and S.J. Wright. *Numerical optimization, Second Edition*. Springer Verlag, 1999.

[28] M. J. Powell. How bad are the BFGS and DFP methods when the objective function is quadratic? *Math. Program.*, 34:34–47, 1986.

[29] Robert B Schnabel. Quasi-newton methods using multiple secant equations. Technical report, University of Colorado Boulder, Computer Science Department, 1983.

[30] Nicol N. Schraudolph, Jin Yu, and Simon Gunter. A Stochastic Quasi-Newton Method for Online Convex Optimization. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 436–443, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR. URL http://proceedings.mlr.press/v2/schraudolph07a.html.

[31] D.F. Shanno. Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computing*, 24:647–656, 07 1970. doi: 10.1090/S0025-5718-1970-0274029-X.

[32] Max. A. Woodbury. Inverting modified matrices. *Memorandum Rept. 42, Statistical Research Group, Princeton University, Princeton, NJ*, 1950.

[33] Peng Xu, Jiyan Yang, Farbod Roosta-Khorasani, Christopher Ré, and Michael W Mahoney. Sub-sampled newton methods with non-uniform sampling. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3000–3008. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6037-sub-sampled-newton-methods-with-non-uniform-sampling.pdf.

Appendix A. Related work

The idea of iteratively updating an approximation of the Hessian or its inverse can be traced back to Davidon [10, 11] with the DFP update. Several type of updates, such as the Broyden method [6] or the BFGS method [7, 16, 17, 31] have been proposed since then. Notably, Dembo et al. [12], Dembo and Steihaug [13] proposed to approximately invert the Hessian using a Conjugate Gradient method. Limited memory BFGS (L-BFGS) [24], where a limited number of vectors are stored for the approximation of the Hessian, has proven to be one of the most powerful type of quasi-Newton method. Last, the use of multisecant equations has also been used in a different context by Fang and Saad [15], Gower and Gondzio [20] and Hennig [22].

To scale up second-order methods, recent works focus on stochastic quasi-Newton methods. The use of stochastic quasi-Newton updates has been investigated by Schraudolph et al. [30], Mokhtari and Ribeiro [25], Moritz et al. [26], Byrd et al. [9] and Gower et al. [19], while approximating the Hessian through sampling methods has been proposed by Erdogdu and Montanari [14], Xu et al. [33] and Agarwal et al. [1], among others.

We now present two popular quasi-Newton updates: the BFGS method, and the multi-secant Broyden method. They will serve as a basis to motivate the needs of generalization of quasi-Newton updates.

A.1. Single secant DFP/BFGS updates

The BFGS update finds a symmetric matrix H_k that satisfies the secant equation (3). Among the many possible solutions, it selects the one closest to H_{k-1} in a weighted Frobenius norm (4), specifically:

$$\begin{aligned} \mathbf{H}_k = &\underset{\mathbf{H}=\mathbf{H}^T}{\operatorname{argmin}} \|\mathbf{H} - \mathbf{H}_{k-1}\|_{\mathbf{W}} \\ &\text{s.t. } \mathbf{H}(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})) = \mathbf{x}_k - \mathbf{x}_{k-1}. \end{aligned} \tag{11}$$

where $\mathbf{W}$ is *any* positive definite matrix such that $\mathbf{W}(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})) = \mathbf{x}_k - \mathbf{x}_{k-1}$ [27, §8.1] — a similar claim holds for the update formula of $\mathbf{B}_k$, known as DFP, whose update reads

$$\begin{aligned} \mathbf{H}_k = &\underset{\mathbf{B}=\mathbf{B}^T}{\operatorname{argmin}} \|\mathbf{B} - \mathbf{B}_{k-1}\|_{\mathbf{W}^{-1}} \\ &\text{s.t. } \mathbf{B}(\mathbf{x}_k - \mathbf{x}_{k-1}) = \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1}). \end{aligned} \tag{12}$$

The matrix is then inverted using the Woodbury matrix identity. In the two update rules, the matrices $\mathbf{W}$ and $\mathbf{W}^{-1}$ are used implicitly, i.e., we do not need to form $\mathbf{W}$ to evaluate $\mathbf{H}_k$ nor $\mathbf{B}_k$.

Solving (11) repeatedly, BFGS builds a sequence $\mathbf{H}_1, \mathbf{H}_2, \ldots$ of matrices such that each $\mathbf{H}_k$ satisfies the kth secant equation. While it may satisfy them approximately, the update rule offers no such guarantees. The same holds for the DFP update.

A.2. Multi-secant Broyden updates

In the case of Broyden update, we seek for a matrix $\mathbf{B}$ for the type-I, or $\mathbf{H}$ for the type-II, that satisfies the secant equations only, without any restriction on the symmetric of the estimate. The update of the standard Broyden method is simpler than BFGS or DFP, and reads

$$\begin{aligned} \mathbf{B}_k &= \underset{\mathbf{B}}{\operatorname{argmin}} \|\mathbf{B} - \mathbf{B}_{k-1}\| \quad \text{s.t. } \mathbf{B}(\mathbf{x}_k - \mathbf{x}_{k-1}) = \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1}), \\ \mathbf{H}_k &= \underset{\mathbf{H}}{\operatorname{argmin}} \|\mathbf{H} - \mathbf{H}_{k-1}\| \quad \text{s.t. } \mathbf{H}(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})) = \mathbf{x}_k - \mathbf{x}_{k-1}. \end{aligned} \tag{13}$$

As for the DFP update, the matrix $\mathbf{B}_k$ can also be inverted cheaply. In [15], the authors show how to extend this update to the case where we want to satisfy more than one secant equation. However, its solution is generally not symmetric.

Appendix B. Algorithm

Algorithm 1 Type-I Symmetric Multisecant step

Input: Function f and gradient ∇f, initial approximation of the Hessian $\mathbf{B}_0$, maximum memory m (can be ∞), relative regularization parameter $\bar{\lambda}$.

1: Compute $g_0 = \nabla f(x_0)$ and perform the initial step

$$\mathbf{x}_1 = \mathbf{x}_0 - \mathbf{B}_0^{-1} \boldsymbol{g}_0$$

2: **for** $t = 1, 2, \ldots$ **do**

3: Form the matrices $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ using the m last pairs $(x_i, \nabla f(x_i))$.

4: Compute the qN direction $\mathbf{d}$ as

$$\mathbf{d}_t = -\mathbf{Z}_\star^{-1} g_t,$$

see (Inv-RSP) with $\mathbf{A} = \Delta\mathbf{X}$, $\mathbf{D} = \Delta\mathbf{G}$, $\mathbf{Z}_{\text{ref}} = \mathbf{B}_0$, $\lambda = \bar{\lambda}\|\mathbf{A}\|$.

5: Perform an approximate-line search

$$\mathbf{x}_{t+1} = \mathbf{x}_t + h_t \mathbf{d}_t, \quad h_t \approx \arg\min_h f\big(\mathbf{x}_t + h_t \mathbf{d}_t\big).$$

6: **end for**

Appendix C. Convergence Analysis for Quadratics

C.1. Convergence analysis for minimizing quadratic functions

In this section we analyse the convergence rate of quasi-Newton methods when f is the quadratic function

$$f(x) = \frac{1}{2}(x - x^*)^T Q (x - x^*), \quad Q \succ 0.$$

For such function, the gradient is written

$$\nabla f(x) = Q(x - x^*).$$

In this case, there is a strong link between the matrices $\Delta\mathbf{X}$ and $\Delta\mathbf{G}$ since

$$\Delta\mathbf{X} = Q\Delta\mathbf{G}, \qquad \Leftrightarrow Q^{-1}\Delta\mathbf{X} = \Delta\mathbf{G}. \tag{14}$$

In this section, we consider any method of the form

$$x_{k+1} = x_k - H_k \nabla f(x_k) \tag{15}$$

where H_k satisfies *exactly* the secants conditions

$$H_k \Delta\mathbf{G} = \Delta\mathbf{X}.$$

We show that this family of method has the optimal rate of convergence

$$\|\nabla f(x_k)\|_M = \min_{p \in \mathcal{P}_k^{(1)}} \|p(Q)\nabla f(x_0)\|_M,$$

where $\mathcal{P}_k^{(1)}$ is the set of polynomials with degree at most k whose coefficients sum to one, and

$$\|v\|_M = \sqrt{v^T M v}, \quad M \succ 0.$$

The next proposition shows the structure of the matrix H_k, if it satifies the secant conditions.

Proposition 5 *If the function is quadratic, the matrix H_k is written*

$$H_k = \Delta \boldsymbol{X}(\Delta \boldsymbol{G})^\dagger + \tilde{H}(I - \Delta \boldsymbol{G}(\Delta \boldsymbol{G})^\dagger)$$

where $(\Delta \boldsymbol{G})^\dagger$ is a pseudo inverse of $\Delta \boldsymbol{G}$ satisfying

$$\Delta \boldsymbol{G}(\Delta \boldsymbol{G})^\dagger \Delta \boldsymbol{G} = \Delta \boldsymbol{G}.$$

Proof Since H_k satisfies the secant conditions, H_k can be written as

$$H_k = \Delta \mathbf{X}(\Delta \mathbf{G})^\dagger$$

where $(\Delta \mathbf{G})^\dagger$ is a pseudo inverse of $\Delta \mathbf{G}$ satisfying

$$\Delta \mathbf{G}(\Delta \mathbf{G})^\dagger \Delta \mathbf{G} = \Delta \mathbf{G}.$$

In this case, it satisfies the secant conditions since $(I - \Delta \mathbf{G}(\Delta \mathbf{G})^\dagger)\Delta \mathbf{G} = 0$ and

$$H_k \Delta \mathbf{G} = \Delta \mathbf{X}(\Delta \mathbf{G})^\dagger \Delta \mathbf{G} = Q^{-1} \Delta \mathbf{G}(\Delta \mathbf{G})^\dagger \Delta \mathbf{G} = Q^{-1} \Delta \mathbf{G} = \Delta \mathbf{X}.$$

■

C.2. Generalized qN step

We introduce the generalized qN step, written

$$x_{k+1} = (X - H_k G)c \tag{16}$$

where c is a vector whose entries sum to one, and

$$X = [x_0, \ldots, x_k], \quad G = [\nabla f(x_0), \ldots, \nabla f(x_k)].$$

The qN update (15) can be seen as a special case of (16) where $c = [0, \ldots, 0, 1]^T$. The nest proposition shows that for any c, the step (16) is identical.

Proposition 6 *For any c, c', the generalized qN step (16) produce the same x_k, i.e.,*

$$(X - H_k G)(c - c') = 0.$$

Proof With proposition 5, we have

$$H_k = \Delta \mathbf{X}(\Delta \mathbf{G})^\dagger.$$

Thus, the difference between two generalized qN step is written

$$(X - \Delta \mathbf{X}(\Delta \mathbf{G})^\dagger G + \tilde{H}(I - \Delta \mathbf{G}(\Delta \mathbf{G})^\dagger)G)(c - c').$$

Since c and c' sum to one, $(c - c')$ sum to zero. Consider the $k \times k - 1$ matrix

$$C = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ 0 & -1 & 1 & \\ & & \ddots & \ddots \end{bmatrix}.$$

Then, it is easy to show that C is full column rank and

$$\Delta \mathbf{X} = XC, \qquad \Delta \mathbf{G} = GC.$$

In addition, for any vector w with $k - 1$ entries that sum to zero, there exists a vector v such that

$$w = Cv.$$

In particular, we consider the vector v that gives

$$Cv = (c - c').$$

We thus have

$$\tilde{H}(I - \Delta\mathbf{G}(\Delta\mathbf{G})^\dagger)G(c - c') = \tilde{H}(I - \Delta\mathbf{G}(\Delta\mathbf{G})^\dagger)\Delta\mathbf{G}v = 0$$

and

$$\begin{aligned}(X - \Delta\mathbf{X}(\Delta\mathbf{G})^\dagger G)(c - c') &= (X - \Delta\mathbf{X}(\Delta\mathbf{G})^\dagger G)Cv, \\ &= (\Delta\mathbf{X} - \Delta\mathbf{X}(\Delta\mathbf{G})^\dagger\Delta\mathbf{G})v, \\ &= Q^{-1}(\Delta\mathbf{G} - \Delta\mathbf{G}(\Delta\mathbf{G})^\dagger\Delta\mathbf{G})v \\ &= Q^{-1}(\Delta\mathbf{G} - \Delta\mathbf{G})v, \\ &= 0.\end{aligned}$$

■

C.3. Accuracy of the generalized qN step

In this section, we consider the generalized qN step, where $\Delta\mathbf{G}$ and $\Delta\mathbf{X}$ are full column rank. At the end of this section we will consider the case where $\Delta\mathbf{G}$ and $\Delta\mathbf{X}$ are not full rank. The next proposition gives the expression of the gradient of the generalized qN step.

Proposition 7 *The gradient of x_{k+1}, generated by the generalized qN iteration (16) is written*

$$\nabla f(x_{k+1}) = (I - Q\tilde{H})(I - P)Gc \qquad \forall c : c^T 1 = 1.$$

where $P = \Delta\boldsymbol{G}(\Delta\boldsymbol{G})\dagger$.

Proof The gradient of x_{k+1} is written

$$\nabla f(x_{k+1}) = Q(x_{k+1} - x^*) = Q((X - H_k G)c - x^*) \qquad \forall c : c^T\mathbf{1} = 1,$$

which is valid for all c that sum to one thanks to Proposition 6. If we writte

$$X^* = [x^* x^* \ldots] = x^*\mathbf{1}^T,$$

then, because $\mathbf{1}^T c = 1$ we have

$$Q(x_{k+1} - x^*) = Q((X - H_k G)c - x^*) = Q(x_{k+1} - x^*) = Q((X - X^* - H_k G)c).$$

In addition, it is easy to show that $Q(X - X^*) = G$. Thus,

$$Q(X - X^* - H_k G)c = (G - QH_k G)c.$$

Finally, using the definition of H_k from Proposition (5),

$$\begin{aligned}\nabla f(x_{k+1}) &= (G - Q\Delta\mathbf{X}(\Delta\mathbf{G})^\dagger G + Q\tilde{H}(I - \Delta\mathbf{G}(\Delta\mathbf{G})^\dagger)G)c, \\ &= (I - \Delta\mathbf{G}(\Delta\mathbf{G})^\dagger - Q\tilde{H}(I - \Delta\mathbf{G}(\Delta\mathbf{G})^\dagger))Gc.\end{aligned}$$

Writting $P = \Delta\mathbf{G}(\Delta\mathbf{G})^\dagger$, we finally have

$$\nabla f(x_{k+1}) = (I - P - Q\tilde{H}(I - P)Gc = (I - Q\tilde{H})(I - P)Gc.$$

■

Intuitively, this means the gradient of the point x_{k+1} is equal to zero in the space generated by P, and in its orthogonal space the better $\tilde{H}$ approximates Q^{-1}, the smaller the norm of the gradient is.

Before going further, we need to prove the following lemma.

Lemma 8 *There exist a c^* such that $\boldsymbol{I}^T(c^*) = 1$ and*

$$(I - P)Gc^* = Gc^*.$$

Moreover, if $G_{k+1} = [G,\ \nabla f(x_{k+1})]$ is also full rank then the last coefficient c^*_{k+1} is nonzero.

Proof If we want $(I - P)Gc^* = Gc^*$, it is sufficient to find c^* such that

$$PGc^* = 0, \qquad \Leftrightarrow \quad c \in \ker(PG).$$

However, the matrix PG of size $k \times k+1$ can be at most of rank k. Thus, the dimension of its kernel it at least one. Now, we will show by contradiction that, if the vector c^* sum to 0, then it cannot be a non-trivial solution of the system.

Indeed, if $c^T\mathbf{1} = 0$ then there exists a vector v such that $Gc = \Delta\mathbf{G}v$ (see proof of Proposition 6). In this case,

$$PGc^* = P\Delta\mathbf{G}v = \Delta\mathbf{G}(\Delta\mathbf{G})^\dagger\Delta\mathbf{G}v = \Delta\mathbf{G}v = Gc^*.$$

However, because the matrix G is full column rank, it is impossible to find a non-zero c^* such that $Gc^* = 0$.

In conclusion, there exists a solution c^* such that $c^* \in \ker(PG)$ and $\mathbf{1}^T c^* = 1$.

Now, we need to show that its last coefficient is nonzero if $\nabla f(x_{k+1})$ is linearly independant of previous gradients, i.e.,

$$\mathbf{rank}([G, \nabla f(x_{k+1})]) = k + 2 \quad \Rightarrow \quad c^*_{k+1} \neq 0.$$

We now use the optimal c^* to write the gradient $\nabla f(x_{k+1})$. Indeed, combined with Proposition 7, we have

$$\nabla f(x_{k+1}) = (I - Q\tilde{H})Gc^*.$$

Writing $c^{(i)}$ the optimal vector obtained for the gradient $\nabla f(x_i)$, we have

$$G_{1\ldots k+1} = [\nabla f(x_1), \ldots, \nabla f(x_{k+1})] = (I - Q\tilde{H})G \begin{bmatrix} c^{(1)} & c^{(2)} & \ldots & c^{(k)} & c^{(k+1)} \\ 0_{k\times 1} & 0_{(k-1)\times 1} & \ldots & 0_{1\times 1} & \end{bmatrix}.$$

Clearly, the matrix of c's is upper-triangular. Thus, if $[G, \nabla f(x_{k+1})]$ is full rank, $G_{1\ldots k+1}$ is also full rank. Thus, it is necessary to the matrix of c's to have non-zero elements in the diagonnal, so $c^{(k+1)}_{k+1} = c^*_{k+1} \neq 0$. ■

With this lemma we can now show that qN can be seen as Krylov methods, under some conditions on the matrix $\tilde{H}$.

Proposition 9 *Assume G full column rank. Then, $\nabla f(x_k)$ can be written*

$$\nabla f(x_{k+1}) = p_{k+1}(I - Q\tilde{H})\nabla f(x_0)$$

where p_{k+1} is a polynomial of degree $k+1$ (i.e. its leading coefficient is nonzero) and its coefficients sum to one $(i.e. p(1) = 1)$.

Proof Using Lemma 8 and Proposition 7, the gradient can be written

$$\nabla f(x_{k+1}) = (I - Q\tilde{H})Gc^*$$

for some c such that $\mathbf{1}^T c^* = 1$ and $c^*_{k+1} \neq 0$ (i.e, it sums to one and the last coefficient is nonzero). We show by recursion that

$$\nabla f(x_{k+1}) = p_{k+1}(I - Q\tilde{H})\nabla f(x_0)$$

where $p_{k+1} \in \mathcal{P}^{(1)}_{k+1}$, the set of polynomial whose coefficients sum to one with nonzero leading coefficients. Of couse, the first element satisfies this condition since

$$\nabla f(x_0) = (I - Q\tilde{H})^0 \nabla f(x_0) = p_0(I - Q\tilde{H})\nabla f(x_0),$$

for the particular polynomial $p_0(z) = 1$. Now assume this is true for all $\nabla f(x_i)$ up to $i = k$. In this case,

$$\begin{aligned}
\nabla f(x_{k+1}) &= (I - Q\tilde{H})Gc^* \\
&= (I - Q\tilde{H}) \sum_{i=1}^{k+1} c_i^* \nabla f(x_{i-1}) \\
&= \underbrace{(I - Q\tilde{H}) \sum_{i=1}^{k+1} c_i^* p_{i-1}(I - Q\tilde{H})}_{=p_{k+1}(I-Q\tilde{H})} \nabla f(x_0)
\end{aligned}$$

Clearly, $p_{k+1}(I - Q\tilde{H})$ is a polynonial of degree at most $k + 1$ since it corresponds to a linear combination of polynomials of degree at most k, then multiplied by $(I - Q\tilde{H})$. It is easy to see that the coefficients of p_{k+1} sum to one, since

$$p_{k+1}(1) = (1) \sum_{i=1}^{k+1} c_i^* p_{i-1}(1)$$

By recurtion, all $p_{i-1}(1) = 1$ and by assumption $\mathbf{1}^T c_i^* = 1$. Now we need to show that the leading coefficient is nonzero. The highest degree polynomial is the following,

$$c_{k+1}^*(I - Q\tilde{H})p_k(I - Q\tilde{H})$$

By recursion, the degree of p_k is *exactly* k, thus its leading coefficient is nonzero. Moreover, it comes with a non-zero contribution since c_{k+1}^* is non-zero by Lemma (8). This means that p_{k+1} has degree *exactly* $k + 1$. ■

This proposition shows us that we are iteratively building a basis of polynomials. This is a crucial point in our proof, as now we are abble to show that the rate of convergence of multisecants qN method is similar to the rate of conjugate gradients or GMRES.

Theorem 10 *If we use a multisecant qN method, then for all $M \succ 0$,*

$$\|\nabla f(x_k)\|_M \leq \|I - M^{1/2}Q\tilde{H}M^{-1/2}\| \min_{p \in \mathcal{P}_k^{(1)}} \|p(I - Q\tilde{H})\nabla f(x_0)\|_M \tag{17}$$

Proof We start with the result of Proposition 7,

$$\nabla f(x_{k+1}) = (I - Q\tilde{H})(I - P)Gc \qquad \forall c : c^T 1 = 1.$$

First, we consider the projector

$$(I - \tilde{P}) = (I - Q\tilde{H})(I - P)(I - Q\tilde{H})^{-1}.$$

Since the formula is valid *for all* c that sum to one, we can pick c such that

$$c^{opt} = \arg\min c : c^T \mathbf{1} = 1 \|(I - Q\tilde{H})(I - P)Gc\|_M$$

for a positive definite matrix M. In this case,

$$\begin{aligned} &\|\nabla f(x_{k+1})\|_M && (18)\\ &= \|(I - Q\tilde{H}(I-P)Gc^{opt}\|_M && (19)\\ &= \min_{c:\mathbf{1}^T c=1} \|(I - Q\tilde{H})(I-P)Gc\|_M && (20)\\ &\leq \|M^{1/2}(I - Q\tilde{H})M^{-1/2}\|_2 \|M^{1/2}(I-P)M^{-1/2}\|_2 \min_{c:\mathbf{1}^T c=1} \|Gc\|_M && (21)\end{aligned}$$

We have that $M^{1/2}(I-P)M^{-1/2}$ is also a projector, thus its norm is bounded by one. By consequence,

$$\|\nabla f(x_{k+1})\|_M \leq \|M^{1/2}(I - Q\tilde{H})M^{-1/2}\|_2 \min_{c:\mathbf{1}^T c=1} \|Gc\|_M$$

By Proposition 9, we have that the $i-th$ column of G represent a polynomial of degree *exactly* $i-1$, whose coefficients sum to one. Thus, by combining the $k+1$ columns of G with coefficients c that also sum to one, we can build any polynomial of $\mathcal{P}_k^{(1)}$. This means

$$\min_{c:\mathbf{1}^T c=1} \|(I - Q\tilde{H})Gc\|_M = \min_{p\in\mathcal{P}_k^{(1)}} \|(I - Q\tilde{H})p(I - Q\tilde{H})\nabla f(x_0)\|_M$$

This prove the desired result. ■

In the particular case where $0 \preceq I - Q\tilde{H} \preceq 1-\kappa$ and $M = I$, we can show a rate similar to conjugate gradients method.

Corollary 11 *Let ζ the degree of the minimal polynomial of $(I - Q\tilde{H})$, and assume $Q\tilde{H}$ invertible. If $(I - Q\tilde{H})$ is symmetric, $0 \preceq I - Q\tilde{H} \preceq 1-\kappa \prec I$ and $M = I$,*

$$\|\nabla f(x_{k+1})\|_2 \leq \begin{cases} 2(1-\kappa)\left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^k \|\nabla f(x_{k+1})\|_2 & \text{if } k \leq \zeta \\ 0 & \text{if } k > \zeta \end{cases}$$

Proof It suffices to split de norm in Theorem 10 as follow,

$$\|\nabla f(x_{k+1})\|_2 \leq \underbrace{\|I - Q\tilde{H}\|_2}_{\leq 1-\kappa} \|\nabla f(x_0)\|_2 \min_{p\in\mathcal{P}_k^{(1)}} \|p(I - Q\tilde{H})\|_2$$

Then, using classical results for minimal polynomial (see for instance [18]), we have that, if $k \leq \zeta$,

$$\min_{p\in\mathcal{P}_k^{(1)}} \|p(I - Q\tilde{H})\|_2 \leq \min_{p\in\mathcal{P}_k^{(1)}} \max_{A:0\preceq A\preceq 1-\kappa} \|p(A)\|_2 \leq 2\left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^k.$$

Otherwise, consider q the minimal polynomial of $I - Q\tilde{H}$. Since $Q\tilde{H}$ is invertible, the matrix $I - Q\tilde{H}$ does not have 1 as eigenvalue, thus $q(1) \neq 0$. In this case, $p = \frac{q}{q(1)}$ is a feasible solution of (17), so

$$\min_{p\in\mathcal{P}_k^{(1)}} \|p(I - Q\tilde{H})\nabla f(x_0)\|_M \leq \|q(I - Q\tilde{H})\nabla f(x_0)\|_M = 0$$

by definition of the minimal polynomial. ■

Appendix D. Symmetric Procrustes Problem

Consider the following problem, known as Symmetric Procrustes.

Theorem 12 *Consider the problem*

$$\boldsymbol{Z}_\lambda = \min_{\boldsymbol{Z}=\boldsymbol{Z}^T\in\mathbb{R}^{d\times d}} \|\boldsymbol{Z}\boldsymbol{A}-\boldsymbol{D}\|_F^2 + \frac{\lambda}{2}\|\boldsymbol{Z}-\boldsymbol{Z}_{ref}\|_F^2, \tag{22}$$

where A, $C\in\mathbb{R}^{d\times m}$, and $Z_0\in\mathbb{R}^{d\times d}$ is a symmetric matrix. Assume $m<d$ and ***rank****(A)* **=** *m. The solution of (22) is given by*

$$\boldsymbol{Z}_\lambda = \boldsymbol{V}_1 Z_1 \boldsymbol{V}_1^T + (\boldsymbol{I}-\boldsymbol{P})\boldsymbol{Z}_D^T + \boldsymbol{Z}_D(\boldsymbol{I}-\boldsymbol{P}) + (\boldsymbol{I}-\boldsymbol{P})^T\boldsymbol{Z}_0(\boldsymbol{I}-\boldsymbol{P}), \tag{23}$$

where

$$[U,\Sigma,V_1] = \boldsymbol{SVD}(A^T,\texttt{'econ'}), \tag{24}$$

$$\boldsymbol{Z}_1 = \left(\frac{1}{\Sigma^2\boldsymbol{1}\boldsymbol{1}^T+\boldsymbol{1}\boldsymbol{1}^T\Sigma^2+2\lambda\boldsymbol{1}\boldsymbol{1}^T}\right)\odot V_1^T\left(AC^T+CA^T+\lambda(Z_0+Z_0^T)\right)V_1, \tag{25}$$

$$P = V_1V_1^T,$$

$$\boldsymbol{Z}_D = \boldsymbol{V}_1\left(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}+2\lambda\boldsymbol{I}\right)^{-1}\boldsymbol{V}_1^T\left(\boldsymbol{A}\boldsymbol{D}^T+2\lambda\boldsymbol{Z}_{ref}\right)(\boldsymbol{I}-\boldsymbol{P}), \qquad A^\dagger = V_1\Sigma^{-1}U^T,$$

where in (25) we used the element-wise division and $\odot$ is the element-wise multiplication (Hadamard product). Assuming the matrix Z_0 is invertible, the inverse Z_*^{-1} is given by

$$Z_\lambda^{-1} = QMQ^T + (I-P)Z_0^{-1}(I-P) \tag{26}$$

where

$$M = \left(Z_1 - Z_DZ_0^{-1}Z_D^T\right)^{-1} \qquad \text{and} \qquad Q = V_1 - (I-P)Z_0^{-1}Z_D^T.$$

Finally, both the SVD in (24) and the matrix-matrix multiplication Z_*D or $Z_*^{-1}D$ have a complexity of the order of $O(m^2d)$, for any matrix $D\in\mathbb{R}^{d\times m}$.

Proof We begin by deriving the solution of (22). By taking the transposition of the matrices inside the Frobenius norm of the first term in (22), we obtain the equivalent problem

$$\min_{Z=Z^T\in\mathbb{R}^{d\times d}} \|A^TZ-C^T\|^2 + \lambda\|Z-Z_0\|_F^2. \tag{27}$$

We write the SVD of $\mathbf{A}^T$ as $\mathbf{U}[\mathbf{\Sigma},0]\mathbf{V}^T$, where $\mathbf{U}\in\mathbb{R}^{m\times m}$, $\mathbf{V}\in\mathbb{R}^{d\times d}$ are orthogonal matrices, and $\mathbf{\Sigma}\in\mathbb{R}^{m\times m}$ is a diagonal matrix with nonnegative entries. Thus, we obtain another problem equivalent to (22) as follows

$$\min_{Z=Z^T\in\mathbb{R}^{d\times d}} \|[\Sigma,0]V^TZV - U^TC^TV\|_F^2 + \lambda\|Z-Z_0\|_F^2.$$

By defining $\tilde{Z} = V^TZV$, (D) can be further written as

$$\begin{aligned}&\min_{\tilde{Z}=\tilde{Z}^T\in\mathbb{R}^{d\times d}} \|[\Sigma,0]\tilde{Z} - U^TC^TV\|_F^2 + \lambda\|\tilde{Z}-V^TZ_0V\|_F^2\\ &= \min_{\tilde{Z}=\tilde{Z}^T\in\mathbb{R}^{d\times d}} \|[\Sigma,0]\tilde{Z}-\tilde{C}\|_F^2 + \lambda\|\tilde{Z}-V^TZ_0V\|_F^2,\end{aligned} \tag{28}$$

where

$$\mathbf{V} = \begin{bmatrix}\mathbf{V}_1 & \mathbf{V}_2\end{bmatrix},\ \tilde{\mathbf{Z}} = \begin{bmatrix}\tilde{\mathbf{Z}}_1 & \tilde{\mathbf{Z}}_D\\ (\tilde{\mathbf{Z}}_D)^T & \tilde{\mathbf{Z}}_2\end{bmatrix},$$

$$\tilde{\mathbf{D}} = \mathbf{U}^T\mathbf{D}^T\mathbf{V} = \begin{bmatrix}\tilde{\mathbf{D}}_1 & \tilde{\mathbf{D}}_2\end{bmatrix},\ \tilde{\mathbf{Z}}_0 = \mathbf{V}^T\mathbf{Z}_0\mathbf{V} = \begin{bmatrix}(\tilde{\mathbf{Z}}_0)_1 & (\tilde{\mathbf{Z}}_0)_D\\ (\tilde{\mathbf{Z}}_0)_D^T & (\tilde{\mathbf{Z}}_0)_2\end{bmatrix}. \tag{29}$$

Here $\tilde{\mathbf{Z}}_1 = (\tilde{\mathbf{Z}}_1)^T \in \mathbb{R}^{m\times m}$, $\tilde{\mathbf{Z}}_2 = \tilde{\mathbf{Z}}_2^T \in \mathbb{R}^{(d-m)\times(d-m)}$, $\tilde{\mathbf{Z}}_D \in \mathbb{R}^{m\times(d-m)}$, $\tilde{\mathbf{D}}_1 \in \mathbb{R}^{m\times m}$, $\tilde{\mathbf{D}}_2 \in \mathbb{R}^{m\times(d-m)}$, $\mathbf{V}_1 \in \mathbb{R}^{d\times m}$, and $\mathbf{V}_2 \in \mathbb{R}^{d\times(d-m)}$. Hence, we can write the objective of (28) as

$$\begin{aligned}
&\|[\boldsymbol{\Sigma}\tilde{\mathbf{Z}}_1, \boldsymbol{\Sigma}\tilde{\mathbf{Z}}_D] - \mathbf{U}^T\mathbf{D}^T\mathbf{V}\|^2 + \lambda\|\tilde{\mathbf{Z}} - \tilde{\mathbf{Z}}_0\|_F^2 \\
&\quad = \|[\boldsymbol{\Sigma}\tilde{\mathbf{Z}}_1, \boldsymbol{\Sigma}\tilde{\mathbf{Z}}_D] - \tilde{\mathbf{D}}\|^2 + \lambda\|\tilde{\mathbf{Z}} - \tilde{\mathbf{Z}}_0\|_F^2 \\
&\quad = \|[\boldsymbol{\Sigma}\tilde{\mathbf{Z}}_1, \boldsymbol{\Sigma}\tilde{\mathbf{Z}}_D] - [\tilde{\mathbf{D}}_1, \tilde{\mathbf{D}}_2]\|^2 + \lambda\|\tilde{\mathbf{Z}} - \tilde{\mathbf{Z}}_0\|_F^2 \\
&\quad = \|\boldsymbol{\Sigma}\tilde{\mathbf{Z}}_1 - \tilde{\mathbf{D}}_1\|^2 + \|\boldsymbol{\Sigma}\tilde{\mathbf{Z}}_D - \tilde{\mathbf{D}}_2\|^2 + \lambda\|\tilde{\mathbf{Z}} - \tilde{\mathbf{Z}}_0\|_F^2 \\
&\quad = \underbrace{\|\boldsymbol{\Sigma}\tilde{\mathbf{Z}}_1 - \tilde{\mathbf{D}}_1\|^2 + \lambda\|\tilde{\mathbf{Z}}_1 - (\tilde{\mathbf{Z}}_0)_1\|^2}_{\text{(i)}} + \underbrace{\|\boldsymbol{\Sigma}\tilde{\mathbf{Z}}_D - \tilde{\mathbf{D}}_2\|^2 + 2\lambda\|\tilde{\mathbf{Z}}_D - (\tilde{\mathbf{Z}}_0)_D\|^2}_{\text{(ii)}} \\
&\qquad + \underbrace{\lambda\|\tilde{\mathbf{Z}}_2 - (\tilde{\mathbf{Z}}_0)_2\|^2}_{\text{(iii)}}.
\end{aligned}$$

Hence, we derive the solution to (22) by minimizing three independent terms as below.

Term (iii): The term

$$\operatorname*{argmin}_{\tilde{\mathbf{Z}}_2=(\tilde{\mathbf{Z}}_2)^T\in\mathbb{R}^{d-m\times d-m}} \lambda\|\tilde{\mathbf{Z}}_2 - (\tilde{\mathbf{Z}}_0)_2\|^2 \tag{30}$$

imposes the constraint $\tilde{\mathbf{Z}}_2 = (\tilde{\mathbf{Z}}_0)_2$. In other words, we have

$$\tilde{\mathbf{Z}}_2 = \mathbf{V}_2^T\mathbf{Z}_0\mathbf{V}_2 \tag{31}$$

Term (i): In what follows, we solve the problem

$$\min_{\tilde{\mathbf{Z}}_1=(\tilde{\mathbf{Z}}_1)^T\in\mathbb{R}^{m\times m}} \|\boldsymbol{\Sigma}\tilde{\mathbf{Z}}_1 - \tilde{\mathbf{D}}_1\|^2 + \lambda\|\tilde{\mathbf{Z}}_1 - (\tilde{\mathbf{Z}}_0)_1\|^2, \tag{32}$$

By writing $\tilde{Z}_1 = (\tilde{z}_{ij}^{(1)})_{i,j\in[d]}$, $\tilde{C}_1 = (c_{ij})_{i,j\in[d]}$, and noting that $\tilde{Z}_1$ is symmetric, we rewrite the optimization problems in terms of the entries in Z as below.

$$\begin{aligned}
\min_{\tilde{Z}\in\mathbb{R}^{d\times d}} &\sum_{i=1}^{m}(\sigma_i\tilde{z}_{ii}^{(1)} - c_{ii})^2 + \big((\sigma_i z_{ij}^{(1)} - c_{ij})^2 + (\sigma_j z_{ij}^{(1)} - c_{ji})^2\big) \\
&+ \lambda\Big(\sum_{i=1}^{m}(\tilde{z}_{ii}^{(1)} - z_{ii})^2 + \sum_{j>i}\big((\tilde{z}_{ij}^{(1)} - z_{ij})^2 + (\tilde{z}_{ij}^{(1)} - z_{ij})^2\big)\Big).
\end{aligned} \tag{33}$$

By setting the derivative w.r.t. z_{ij}, we obtain for $\lambda > 0$

$$\tilde{z}_{ij}^{(1)} = \frac{\sigma_i c_{ij} + \sigma_j c_{ji} + \lambda(z_{ij} + z_{ji})}{\sigma_i^2 + \sigma_j^2 + 2\lambda}, \tag{34}$$

where we do not need to deal with the case $\sigma_i^2 + \sigma_j^2 = 0$ as in the setting $\lambda \to 0$. We can equivalently write

$$\tilde{\mathbf{Z}}_1 = \left(\frac{1}{\Sigma^2\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T\Sigma^2 + 2\lambda\mathbf{1}\mathbf{1}^T}\right) \odot \mathbf{V}_1^T\left(\mathbf{A}\mathbf{D}^T + \mathbf{D}\mathbf{A}^T + \lambda(\mathbf{Z}_{\text{ref}} + \mathbf{Z}_{\text{ref}}^T)\right)\mathbf{V}_1, \tag{35}$$

$$\tag{36}$$

where $\odot$ is the Hadamard product computing the product element-wise.

Term (ii): In addition, the problem

$$\min_{\tilde{Z}_D\in\mathbb{R}^{m\times d-m}} \|\Sigma\tilde{Z}_D - \tilde{C}_2\|^2 + 2\lambda\|\tilde{Z}_D - (\tilde{Z}_0)_D\|^2 \tag{37}$$

has a closed form solution given by

$$\tilde{\mathbf{Z}}_D = \left(\boldsymbol{\Sigma}^\top \boldsymbol{\Sigma} + 2\lambda \mathbf{I}\right)^{-1} \left(\boldsymbol{\Sigma}^\top \tilde{\mathbf{D}}_2 + 2\lambda (\tilde{\mathbf{Z}}_0)_D\right). \tag{38}$$

$$\begin{aligned}\mathbf{Z}_D &= \mathbf{V}_1 \tilde{\mathbf{Z}}_D \mathbf{V}_2^T \\ &= \mathbf{V}_1 \left(\boldsymbol{\Sigma}^\top \boldsymbol{\Sigma} + 2\lambda \mathbf{I}\right)^{-1} \left(\boldsymbol{\Sigma}^\top \tilde{\mathbf{D}}_2 + 2\lambda (\tilde{\mathbf{Z}}_0)_D\right) \mathbf{V}_2^T \qquad (39) \\ &= \mathbf{V}_1 \left(\boldsymbol{\Sigma}^\top \boldsymbol{\Sigma} + 2\lambda \mathbf{I}\right)^{-1} \left(\boldsymbol{\Sigma}^\top \mathbf{U}^T \mathbf{D}^T \mathbf{V}_2 + 2\lambda \mathbf{V}_1^T (\mathbf{Z}_0)_D \mathbf{V}_2\right) \mathbf{V}_2^T. \qquad (40)\end{aligned}$$

Since we have $\mathbf{V}_2 \mathbf{V}_2^T = \mathbf{I} - \mathbf{V}_1 \mathbf{V}_1^T = \mathbf{I} - \mathbf{P}$,

$$\begin{aligned}\mathbf{Z}_D &= \mathbf{V}_1 \left(\boldsymbol{\Sigma}^\top \boldsymbol{\Sigma} + 2\lambda \mathbf{I}\right)^{-1} \left(\boldsymbol{\Sigma}^\top \mathbf{U}^T \mathbf{D}^T + 2\lambda \mathbf{V}_1^T \mathbf{Z}_{\text{ref}}\right)(\mathbf{I} - \mathbf{P}) \\ &= \mathbf{V}_1 \left(\boldsymbol{\Sigma}^\top \boldsymbol{\Sigma} + 2\lambda \mathbf{I}\right)^{-1} \mathbf{V}_1^T \left(\mathbf{A}\mathbf{D}^T + 2\lambda \mathbf{Z}_{\text{ref}}\right)(\mathbf{I} - \mathbf{P}) \qquad (41)\end{aligned}$$

In the case where Z_0 is a diagonal matrix, and since $V_1^T(I - P) = 0$,

$$Z_D = \mathbf{V}_1 \left(\boldsymbol{\Sigma}^T \boldsymbol{\Sigma} + 2\lambda \mathbf{I}\right)^{-1} \mathbf{V}_1^T \left(\mathbf{A}\mathbf{D}^T\right)(\mathbf{I} - \mathbf{P}).$$

Therefore, the solution can be written as

$$\begin{aligned}\mathbf{Z}_\lambda &= \begin{bmatrix}\mathbf{V}_1 & \mathbf{V}_2\end{bmatrix} \begin{bmatrix}\tilde{\mathbf{Z}}_1 & \tilde{\mathbf{Z}}_D \\ (\tilde{\mathbf{Z}}_D)^T & \tilde{\mathbf{Z}}_2\end{bmatrix} \begin{bmatrix}\mathbf{V}_1 & \mathbf{V}_2\end{bmatrix}^T \\ &= \mathbf{V}_1 \tilde{\mathbf{Z}}_1 \mathbf{V}_1^T + \mathbf{V}_1 \tilde{\mathbf{Z}}_D \mathbf{V}_2^T + \mathbf{V}_2 \tilde{\mathbf{Z}}_D^T \mathbf{V}_1^T + \mathbf{V}_2 \tilde{\mathbf{Z}}_2 \mathbf{V}_2^T \\ &= \mathbf{Z}_1 + \mathbf{Z}_D + \mathbf{Z}_D^T + (\mathbf{I} - \mathbf{P})\mathbf{Z}_{\text{ref}}(\mathbf{I} - \mathbf{P}), \qquad (42)\end{aligned}$$

where $\mathbf{P} = \mathbf{V}_1 \mathbf{V}_1^T = \mathbf{I} - \mathbf{V}_2 \mathbf{V}_2^T$ and $\mathbf{Z}_D = \mathbf{V}_1 \left(\boldsymbol{\Sigma}^T \boldsymbol{\Sigma} + 2\lambda \mathbf{I}\right)^{-1} \mathbf{V}_1^T \left(\mathbf{A}\mathbf{D}^T + 2\lambda (\mathbf{Z}_0)_D\right)(\mathbf{I} - \mathbf{P})$, and $\mathbf{Z}_1 = \mathbf{V}_1 \tilde{\mathbf{Z}}_1 \mathbf{V}_1^T$.

Below we compute the inverse of $\mathbf{Z}_*$. Since

$$\begin{aligned}\mathbf{Z}_* &= \mathbf{V} \begin{bmatrix}\tilde{\mathbf{Z}}_1 & \tilde{\mathbf{Z}_D} \\ \tilde{\mathbf{Z}_D}^T & \tilde{\mathbf{Z}}_2\end{bmatrix} \mathbf{V}^T \\ &= \mathbf{V} \tilde{\mathbf{Z}} \mathbf{V}^T,\end{aligned}$$

we can write

$$\mathbf{Z}_*^{-1} = \mathbf{V} \tilde{\mathbf{Z}}^{-1} \mathbf{V}^T.$$

By the Woodbury matrix identity [32] and (29), we have

$$\tilde{\mathbf{Z}}^{-1} = \begin{bmatrix}\mathbf{M}_1 & -\mathbf{M}_1 \tilde{\mathbf{Z}}_D \tilde{\mathbf{Z}}_2^{-1} \\ -\tilde{\mathbf{Z}}_2^{-1} \tilde{\mathbf{Z}}_D^T \mathbf{M}_1 & \tilde{\mathbf{Z}}_2^{-1} + \tilde{\mathbf{Z}}_2^{-1} \tilde{\mathbf{Z}}_D^T \mathbf{M}_1 \tilde{\mathbf{Z}}_D \tilde{\mathbf{Z}}_2^{-1}\end{bmatrix}, \tag{43}$$

with $\mathbf{M}_1 = (\tilde{\mathbf{Z}_1} - \tilde{\mathbf{Z}_D} \tilde{\mathbf{Z}_2}^{-1} \tilde{\mathbf{Z}}_D^T)^{-1}$. Hence $\mathbf{Z}_*^{-1} = \mathbf{V} \tilde{\mathbf{Z}}^{-1} \mathbf{V}^T$ can be rewritten as

$$\begin{aligned}\mathbf{Z}_*^{-1} &= \mathbf{V}_1 \mathbf{M}_1 \mathbf{V}_1^T + \mathbf{V}_2 \tilde{\mathbf{Z}_2}^{-1} \tilde{\mathbf{Z}}_D^T \mathbf{M}_1 \tilde{\mathbf{Z}_D} \tilde{\mathbf{Z}_2}^{-1} \mathbf{V}_2^T \\ &\quad + \mathbf{V}_2 \tilde{\mathbf{Z}_2}^{-1} \mathbf{V}_2^T - \mathbf{V}_1 \mathbf{M}_1 \tilde{\mathbf{Z}_D} \tilde{\mathbf{Z}_2}^{-1} \mathbf{V}_2^T - \mathbf{V}_2 \tilde{\mathbf{Z}_2}^{-1} \tilde{\mathbf{Z}_D}^T \mathbf{M}_1 \mathbf{V}_1^T \qquad (44) \\ &= \mathbf{Q}\mathbf{M}\mathbf{Q}^T + (\mathbf{I} - \mathbf{P})\mathbf{Z}_0^{-1}(\mathbf{I} - \mathbf{P}),\end{aligned}$$

where $M = \left(Z_1 - Z_D Z_0^{-1} Z_D^T\right)^{-1}$ and $Q = \left(V_1 - (I - P) Z_0^{-1} Z_D^T\right)$. For a proof that for any matrix $\mathbf{D} \in \mathbb{R}^{d \times m}$. ■

Appendix E. Numerical Experiment

In this section, we compare our symmetric multisecant algorithm to existing methods in the literature in different settings. We first compare the quality of the estimate of the Hessian (and its inverse). Then, we compare the speed of convergence when using this estimate to estimate the Newton-step.

E.1. Hessian Recovery

We analyse the following problem, consisting in the recovery of the inverse of a symmetric Hessian $\mathbf{Q}^{-1}$ of a quadratic function, that satisfies

$$\mathbf{Q}^{-1}\Delta\mathbf{G} = \Delta\mathbf{X}, \quad \mathbf{Q} = \mathbf{Q}^T.$$

However, we have only access to $\tilde{\Delta\mathbf{G}}$, a corrupted version of $\Delta\mathbf{G}$. Such case happens in the case where the oracle is a stochastic gradient, for example.

In our case, we consider the worst-case ℓ_2 corruption

$$\tilde{\Delta\mathbf{G}} = \mathbf{U}_{\Delta\mathbf{G}} \max\{\Sigma_{\Delta\mathbf{G}} - \epsilon \cdot \sigma_1(\Delta\mathbf{G}),\, 0\}\mathbf{V}_{\Delta\mathbf{G}}^T,$$

where $\mathbf{U}_{\Delta\mathbf{G}}\Sigma_{\Delta\mathbf{G}}\mathbf{V}_{\Delta\mathbf{G}}^T$ is the SVD of $\Delta\mathbf{G}$, and ϵ is the relative perturbation intensity. When $\epsilon = 1$, the matrix $\tilde{\Delta\mathbf{G}}$ is full of zero.

We estimate $\mathbf{Q}^{-1}$ using different techniques, that we compare using the relative residual error

$$\text{error}(\mathbf{Q}_{\text{est}}^{-1}) = \|\mathbf{Q}_{\text{est}}^{-1}\Delta\mathbf{G} - \Delta\mathbf{X}\|_F / \|\Delta\mathbf{X}\|_F.$$

Note that, in our error function, we use the noise-free version of $\Delta\mathbf{G}$.

Our baseline is the diagonal estimate, corresponding to the inverse of the Lipchitz constant of $\mathbf{Q}$, typically used as a step-size in the gradient method. We also use ℓ-BFGS and our Type-1 and Type-2 multisecant algorithms, solving respectively (Inv-RSP) and (Sol-RSP) with $\mathbf{A} = \tilde{\Delta\mathbf{G}}$, $\mathbf{D} = \Delta\mathbf{X}$, $\mathbf{B}_0 = \mathbf{H}_0^{-1} = \|\mathbf{Q}\|$, and $\lambda = 10^{-10}$ for their regularized version. The number of secant equations is 50. The results are reported in Figure 2.

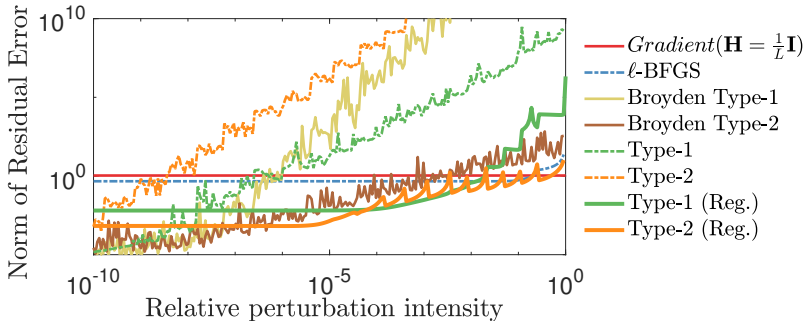

Figure 1: Comparison of different methods to estimate a symmetric matrix. We clearly see that multisecant method perform well in a small-noise regime, but quickly gets out of control for larger perturbation. This is not the case for their regularized counterpart ($\lambda = 10^{-10}$), whose clearly show a more stable behavior. The performance of BFGS are quite poor compared to multisecant algorithms, since it can only satisfy one secant equation at a time. Finally, the type-II multisecant Broyden method seems stable, but does not recover a symmetric matrix.

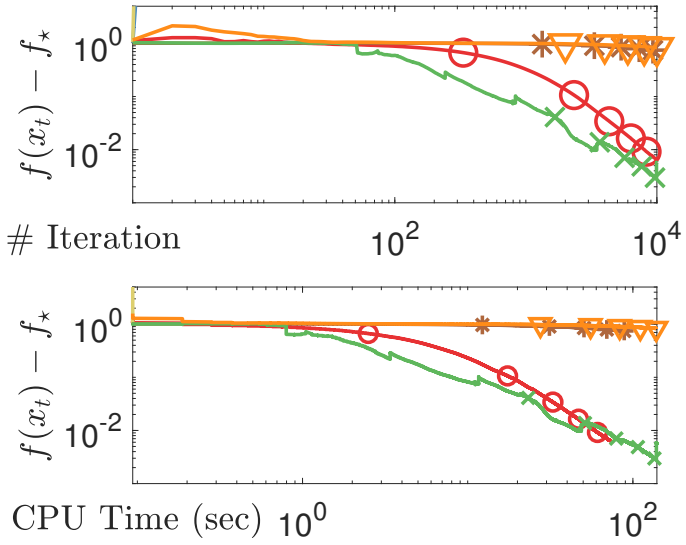

Figure 2: Comparison of the stability of qN methods with stochastic gradients on Madelon dataset. We report the function value of the average of the iterates. The batch size is 10% of the number of points. Since the function is stochastic, we used only unitary stepsize. The memory is 25, and the relative regularization $\bar{\lambda} = 10^{-4}$. The condition number is 10^3. The $\ell-$BFGS and the multisecant Broyden of type 1 are divergent in this situation. With unitary stepsize, the regularized symmetric multisecant Type-I method is slightly faster than stochastic gradient *with no extra tuning*.