TECHNISCHE UNIVERSITÄT
CHEMNITZ

Technische Universität Chemnitz
Fakultät für Mathematik
Professur Numerische Mathematik
(Partielle Differentialgleichungen)

bachelor thesis

The Riemannian BFGS Method and its Implementation in Julia

submitted in fulfilment of the requirements for the degree of

Bachelor of Science

submitted by	Tom-Christian Riemer
student number	444019
1. referee	Referee A
2. referee	Referee B

date of submission In the Future, last compiled September 12, 2020

Contents

1 The BFGS-Method For The Euclidean Case

1.1 Preliminaries

$$\min f(x), \quad x \in \mathbb{R}^n \tag{1.1.1}$$

Algorithm 1 General descent method

1: $x_0 \in \mathbb{R}^n,\ k = 0$
2: **while** x_k does not satisfy any stopping criterion **do**
3: Determine a descent direction d_k of f in x_k.
4: Determine a stepsize $\alpha_k > 0$ with $f(x_k + \alpha_k d_k) < f(x_k)$.
5: Set $x_{k+1} = x_k + \alpha_k d_k$ and $k = k + 1$.
6: **end while**
7: **return** x_k

Algorithm 2 Local Newton's method

1: $x_0 \in \mathbb{R}^n,\ 0 \leq \epsilon < 1,\ k = 0$
2: **while** $\|\nabla f(x_k)\| > \epsilon$ **do**
3: Determine $d_k \in \mathbb{R}^n$ by solving

$$\nabla^2 f(x_k) d = -\nabla f(x_k).$$

4: Set $x_{k+1} = x_k + d_k$ and $k = k + 1$.
5: **end while**
6: **return** x_k

Wolfe conditions:

A popular inexact line search condition is the so called Armijo condition or Armijo rule. It stipulates a stepsize α_k that should first of all give a sufficient decrease in the objective function f, as measured by the following inequality:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^{\mathrm{T}} d_k \tag{1.1.2}$$

for some constant $c_1 \in (0, 1)$. In other words, the reduction in f should be proportional to the stepsize α_k and the directional derivative $\nabla f(x_k)^{\mathrm{T}} d_k$.
Eq. (1.1.2) is not enough by itself to ensure that the algorithm makes reasonable progress because it is

satisfied for all sufficiently small values of α_k. To rule out unacceptably short steps we introduce a second requirement, which requires α_k to satisfy

$$\nabla f(x_k + \alpha_k d_k)^{\mathrm{T}} d_k \geq c_2 \nabla f(x_k)^{\mathrm{T}} d_k \tag{1.1.3}$$

for some constant $c_2 \in (c_1, 1)$. The geometric interpretation of Eq. (1.1.3) is that the slope $\nabla f(x_k + \alpha_k d_k)^{\mathrm{T}} d_k$, which is simply the derivative of $f(x_k + \alpha_k d_k)$, at the acceptable point must be greater than or equal to some multiple c_2 of the initial slope $\nabla f(x_k)$. This makes sense because if the slope $f(x_k + \alpha d_k)$ is strongly negative, we have an indication that we can reduce f significantly by moving further along the chosen direction. On the other hand, if $f(x_k + \alpha_k d_k)$ is only slightly negative or even positive, it is a sign that we cannot expect much more decrease in f in this direction, so it makes sense to terminate the line search.
Eq. (1.1.2) and Eq. (1.1.3) are known as the Wolfe-Powell inexact line search rule, Wolfe-Powell rule or just Wolfe conditions with $0 < c_1 < c_2 < 1$. The requirement $0 < c_1 < c_2 < 1$ is necessary, such that there exists stepsize α_k satisfying the Wolfe-Powell rule (see Sun, Yuan, 2006, p. 104-105).
It should point out that Eq. (1.1.3) is an approximation of the orthogonal condition $\nabla f(x_{k+1})^{\mathrm{T}} d_k = 0$. However, unfortunately, one possible disadvantage of Eq. (1.1.3) is that it does not reduce to an exact line search in the limit $c_2 \to 0$. In addition, a stepsize may satisfy the Wolfe conditions without being particularly close to a minimizer of $f(x_k + \alpha d_k)$. We can, however, modify Eq. (1.1.3) to force α_k to lie in at least a broad neighborhood of a local minimizer or stationary point of $f(x_k + \alpha d_k)$. The strong Wolfe conditions (or strong Wolfe-Powell rule) require α_k to satisfy Eq. (1.1.2) and

$$|\nabla f(x_k + \alpha_k d_k)^{\mathrm{T}} d_k| \geq c_2 |\nabla f(x_k)^{\mathrm{T}} d_k| \tag{1.1.4}$$

with $0 < c_1 < c_2 < 1$. The only difference with the Wolfe conditions is that we no longer allow the derivative $\nabla f(x_k + \alpha_k d_k)^{\mathrm{T}} d_k$ to be too positive. Hence, we exclude points that are far from stationary points of $f(x_k + \alpha d_k)$.
In general, the smaller the value c_2, the more exact the line search. Normally, taking $c_2 = 0.1$ gives a fairly accurate line search, whereas the value $c_2 = 0.9$ gives a weak line search. However, taking too small c_2 may be unwise, because the smaller the value c_2, the more expensive the computing effort. Usually, $c_1 = 0.1$ and $c_2 = 0.4$ are suitable, and it depends on the specific problem.
It is not difficult to prove that there exist stepsizes that satisfy the Wolfe conditions for every function f that is smooth and bounded below.

Lemma 1.1.1 (Nocedal, Wright, 2006, Lemma 3.1)**.** *Suppose that $f\colon \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. Let d_k be a descent direction at x_k, and assume that f is bounded below along the ray $\{x_k + \alpha d_k | \alpha > 0\}$. Then if $0 < c_1 < c_2 < 1$, there exist intervals of stepsizes satisfying the Wolfe conditions and the strong Wolfe conditions.*

The Wolfe conditions are scale-invariant in a broad sense: Multiplying the objective function by a constant or making an affine change of variables does not alter them. They can be used in most line search methods, and are particularly important in the implementation of quasi-Newton methods Nocedal, Wright, 2006, p. 33-35.

The question now arises of how the stepsize is determined. We therefore introduce the concept of a stepsize strategy:

Definition 1.1.2 (Geiger, Kanzow, 1999, p. 27). *A map T of $\mathbb{R}^n \times \mathbb{R}^n$ into the power set of positive real numbers $\mathcal{P}((0,\infty))$, i.e. a map that assigns a subset $T(x,d)$ of $\mathbb{R}$ to each pair (x,d), is called stepsize strategy or stepsize rule. We call such a stepsize rule (under certain conditions) well-defined, if (under these conditions) the set $T(x,d)$ for each pair (x,d) with $\nabla f(x)^{\mathrm{T}} d < 0$ is not empty.*

Among this set of illustrations there is a certain subset, which includes one that meets the Wolfe conditions. We talk about the efficient stepsize strategies introduced by Warth, Werner, 1977, Definition 0.1. But we use the following definition for reasons of comprehensibility:

Definition 1.1.3 (Geiger, Kanzow, 1999, Definition 4.5). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, $x \in \mathbb{R}^n$ and $d \in \mathbb{R}^n$ a descent direction from f in x. A stepsize strategy T is called efficient if there is a constant $\theta > 0$ independent of x and d with*

$$f(x + \alpha d) \leq f(x) - \theta \left(\frac{\nabla f(x)^{\mathrm{T}} d}{\|d\|} \right)^2$$

for all $\alpha \in T(x,d)$.

The stepsize strategy introduced in Powell, 1976 (see. Werner, 1978/79, Definition 2.2.) meets the Wolfe conditions. It can be shown that this is efficient. Because of the introduction in Powell, 1976, it is often called Wolfe-Powell stepsize strategy, in brief, the Wolfe-Powell rule.
These stepsize strategies are mainly used in theory. In practical applications, algorithms are used to obtain stepsizes that meet the respective conditions. There are different algorithms that find a stepsize that meets the (strong) Wolfe conditions. In practice (mainly because of convergence reasons) the stepsize $\alpha_k = 1$ is preferred if it fits.

Theorem 1.1.4 (Sun, Yuan, 2006, Theorem 1.2.15). *Let $A \in \mathbb{R}^{n\times n}$ be nonsingular and $u, v \in \mathbb{R}^n$ be arbitrary. If*

$$1 + v^{\mathrm{T}} A^{-1} u \neq 0,$$

then the rank-one update $A + uv^{\mathrm{T}}$ of A is nonsingular, and its inverse is represented by

$$(A + uv^{\mathrm{T}})^{-1} = A^{-1} - \frac{A^{-1} u v^{\mathrm{T}} A^{-1}}{1 + v^{\mathrm{T}} A^{-1} u}.$$

Theorem 1.1.5 (Sherman-Morrison-Woodbury Theorem Sun, Yuan, 2006, Theorem 1.2.16). *Let $A \in \mathbb{K}^{n\times n}$ be a nonsingular matrix, $U, V \in \mathbb{K}^{n\times m}$. If $I + V^* A^{-1} U$ is invertible, then $A + UV^*$ is invertible and*

$$(A + UV^*)^{-1} = A^{-1} - A^{-1} U (I + V^* A^{-1} U)^{-1} V^* A^{-1}.$$

Note: The Sherman-Morrison-Woodbury formula can be extended to rank R modifications Ulbrich, Ulbrich, 2012, p. 70.

Theorem 1.1.6 (Nocedal, Wright, 2006, Theorem 3.2.). *Consider any iteration of the form $x_{k+1} = x_k + \alpha_k d_k$, where d_k is a descent direction and α_k satisfies the Wolfe conditions Eq. (1.1.2) and Eq. (1.1.3). Suppose that f is bounded below in $\mathbb{R}^n$ and that f is continuously differentiable in an open set $\mathcal{N}$ containing the level set $\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$, where x_0 is the starting point of the iteration. Assume also that the gradient ∇f is Lipschitz continuous on $\mathcal{N}$, that is, there exists a constant $L > 0$ such that*

$$\|\nabla f(x) - \nabla f(\tilde{x})\| \leq L\|x - \tilde{x}\|, \quad \text{for all } x, \tilde{x} \in \mathcal{N}.$$

Then

$$\sum_{k \geq 0} \cos(\theta_k)^2 \|\nabla f(x_k)\|^2 < \infty,$$

where

$$\cos(\theta_k) = -\frac{\nabla f(x_k)^{\mathrm{T}} d_k}{\|\nabla f(x_k)\| \|d_k\|}.$$

Theorem 1.1.6 quantifies the effect of properly chosen stepsizes α_k. It describes how far d_k can deviate from the steepest descent direction $\nabla f(x_k)$ and still produce a globally convergent iteration. Various line search termination conditions can be used to establish this result, but for concreteness we will consider only the Wolfe conditions Eq. (1.1.2) and Eq. (1.1.3), Nocedal, Wright, 2006, p. 38.
The theorem Theorem 1.1.6 implies

$$\lim_{k \to \infty} \cos(\theta_k)^2 \|\nabla f(x_k)\|^2 = 0. \tag{1.1.5}$$

This limit Eq. (1.1.5) can be used in turn to derive global convergence results for line search algorithms. If the method for choosing the search direction d_k ensures that the angle θ_k is bounded away from $90°$, there is a positive constant δ such that

$$\cos(\theta_k) \geq \delta > 0.$$

It follows immediately from Eq. (1.1.5) that

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

In other words, the gradient norms $\|\nabla f(x_k)\|$ converge to zero, provided that the search directions d_k are never too close to orthogonality with the gradient.
In some cases (including the global convergence analysis of the BFGS method), only the weaker result

$$\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0. \tag{1.1.6}$$

can be shown using Eq. (1.1.5). That means, just a subsequence of the gradient norms $\|\nabla f(x_{k_j})\|$ converges to zero, rather than the whole sequence. This can be shown by a contradiction. Suppose that Eq. (1.1.6) does not hold, so that the gradients remain bounded away from zero, that is, there exists $\gamma > 0$ such that

$$\|\nabla f(x_k)\| \geq \gamma, \quad \text{for all } k \text{ sufficiently large.}$$

Then from Eq. (1.1.5) we conclude that

$$\lim_{k\to\infty} \cos(\theta_k) = 0 \tag{1.1.7}$$

that is, the entire sequence $\{\cos(\theta_k)\}_k$ converges to 0. To establish Eq. (1.1.6), therefore, it is enough to show that a subsequence $\{\cos(\theta_{k_j})\}_j$ is bounded away from zero.

Theorem 1.1.7 (Nocedal, Wright, 2006, Theorem 3.6)**.** *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. Consider the iteration $x_{k+1} = x_k + \alpha_k d_k$, where d_k is a descent direction and α_k satisfies the Wolfe conditions with $c_1 \leq \frac{1}{2}$. If the sequence $\{x_k\}_k$ converges to a point x^* such that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite, and if the search direction satisfies*

$$\lim_{k\to\infty} \frac{\|\nabla f(x_k) + \nabla^2 f(x_k) d_k\|}{\|d_k\|} = 0 \tag{1.1.8}$$

then

(i) the stepsize $\alpha_k = 1$ is admissible for all k greater than a certain index k_0 and

(ii) if $\alpha_k = 1$ for all $k > k_0$, $\{x_k\}_k$ converges to x^* superlinearly.

It is easy to see that if $c_1 > \frac{1}{2}$, then the line search would exclude the minimizer of a quadratic, and

unit stepsizes may not be admissible.
If d_k is a quasi-Newton search direction of the form $d_k = -H_k^{-1}\nabla f(x_k)$, where the symmetric and positive definite matrix H_k is updated at every iteration by a quasi-Newton updating formula, then Eq. (1.1.8) is equivalent to

$$\lim_{k\to\infty} \frac{\|(H_k - \nabla^2 f(x^*))d_k\|}{\|d_k\|} = 0 \tag{1.1.9}$$

Hence, we have the surprising (and delightful) result that a superlinear convergence rate can be attained even if the sequence of quasi-Newton matrices H_k does not converge to $\nabla^2 f(x^*)$; it suffices that the H_k become increasingly accurate approximations to $\nabla^2 f(x^*)$ along the search directions d_k. Importantly, condition Eq. (1.1.9) is both necessary and sufficient for the superlinear convergence of quasi-Newton methods Nocedal, Wright, 2006, p. 47.

Corollary 1.1.8 (Geiger, Kanzow, 1999, Lemma 7.9)**.** *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable, $\{H_k\}_k$ a sequence of regular matrices in $\mathbb{R}^{n\times n}$, $x_0 \in \mathbb{R}^n$ and $\{x_k\}_k \subseteq \mathbb{R}^n$ a sequence definded by*

$$x_{k+1} = x_k - H_k^{-1}\nabla f(x_k),\ k = 0,\ 1,\ \cdots$$

with the limit $\lim_{n\to\infty} x_k = x^*$, $x_k \neq x^*$ for all $k \in \mathbb{N}$ and $\nabla^2 f(x^*)$ regular. Then the following statements are equivalent

(i) $\{x_k\}_k \to x^*$ *superlinear and* $\nabla f(x^*) = 0$.

(ii) $\|(\nabla^2 f(x_k) - H_k)(x_{k+1} - x_k)\| = o(\|x_{k+1} - x_k\|)$

(iii) $\|(\nabla^2 f(x^*) - H_k)(x_{k+1} - x_k)\| = o(\|x_{k+1} - x_k\|)$

These conditions are also called Dennis-Moré conditions. They show that for superlinear convergence it is only important that $\nabla^2 f(x_k)(x_{k+1} - x_k)$ and $H_k(x_{k+1} - x_k)$ match sufficiently well. It is therefore not necessary that H_k approximates the entire Hessian matrix $\nabla^2 f(x_k)$ well.

1.2 Quasi-Newton Methods

Quasi-Newton methods are a class of numerical methods for solving nonlinear minimization problems. As the name suggests, these are based on the Newton method, but attempt to minimize the computational effort. The class goes back to the physicist William Davidon of the Argonne National Laboratory, who developed the first algorithm in the mid 1950s.
For the Newton method, both the gradient and the Hessian are calculated in every iteration. Of course, we get useful information about curvature of our function from the Hessian, get local at least superlinear convergence and if we add a method for determining stepsizes, we even get global

convergence. But there are arguments against the Newton method, mainly related to the calculation of the Hessian. For example the calculation could be too costly or not possible at all (which includes the case that the Hessian does not exist). Quasi-Newton methods follow the strategy of not calculating and instead approximating it. Henceforth we call the approximation of the Hessian matrix $\nabla^2 f(x_k)$ used in each iteration H_k.
It is expected that the sequence $\{H_k\}_k$ should posses positive definiteness, $d_k = -H_k^{-1}\nabla f(x_k)$ should be a descent direction and the resulting method should behave like Newton's method in terms of convergence. Of course, the calculation should cost less.
Let $f\colon D \to \mathbb{R}$ be twice continuously differentiable on an open subset $D \subset \mathbb{R}^n$. We consider the quadratic Taylor-approximation of f at x_{k+1}:

$$f(x) \approx m_{k+1}(x) = f(x_{k+1}) + g_{k+1}^{\mathrm{T}}(x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^{\mathrm{T}} G_{k+1}(x - x_{k+1})$$

where $g_{k+1} \triangleq \nabla f(x_{k+1})$ and $G_{k+1} \triangleq \nabla^2 f(x_{k+1})$. For the gradient we obtain

$$\nabla f(x) \approx \nabla m_{k+1}(x) = g_{k+1} + G_{k+1}(x - x_{k+1}).$$

Setting $x = x_k,\ s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k) = g_{k+1} - g_k$, we get

$$G_{k+1}^{-1} y_k \approx s_k.$$

This holds with equality, if f is a quadratic function.
Now one is interested in the fact that an approximation of the Hessian inverse B_{k+1} satisfies this relation for the quasi-Newton method, i.e.,

$$B_{k+1} y_k = s_k \tag{1.2.1}$$

which is called the quasi-Newton equation, quasi-Newton condition or secant equation. A method that uses this condition to generate its symmetric Hessian (inverse) approximations is called a quasi-Newton method.
For quasi-Newton methods we replace the Hessian of our objective function $\nabla^2 f(x_{k+1})$ in the model by an approximation H_{k+1}:

$$m_{k+1}(x) = f(x_{k+1}) + g_{k+1}^{\mathrm{T}}(x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^{\mathrm{T}} H_{k+1}(x - x_{k+1})$$

which satisfies the interpolation conditions:

$$m_{k+1}(x_{k+1}) = f(x_{k+1}) \quad \text{and} \quad \nabla m_{k+1}(x_{k+1}) = \nabla f(x_{k+1}).$$

Unlike the normal Newton method, in which we require that $\nabla^2 m(x_{k+1}) = G_{k+1}$, we want the model to satisfy

$$\nabla m_{k+1}(x_k) = g_k$$

from which follows

$$g_k = g_{k+1} + H_{k+1}(x_k - x_{k+1}).$$

So we have

$$H_{k+1}(x_{k+1} - x_k) = g_{k+1} - g_k \quad \Leftrightarrow \quad H_{k+1}s_k = y_k. \tag{1.2.2}$$

This is also called the quasi-Newton equation, quasi-Newton condition or secant equation but now expressed with the approximation of the Hessian.
Of course, we see immediately that the following relationship holds

$$H_k = B_k^{-1} \quad \text{for all } k \in \mathbb{N}_0 \tag{1.2.3}$$

and vice versa Sun, Yuan, 2006.
We now come to the definition of a single iteration. A model at the current iteration x_k is defined:

$$f(x) \approx m_k(x) = f(x_k) + g_k^{\mathrm{T}}(x - x_k) + \frac{1}{2}(x - x_k)^{\mathrm{T}} H_k (x - x_k)$$

Assuming that H_k is positive definite, we get a quadratic convex model m_k. The minimizer d_k of it, we can write explicitly as

$$d_k = -H_k^{-1} g_k = -B_k g_k \tag{1.2.4}$$

is used as the search direction, and the new iterate is

$$x_{k+1} = x_k + \alpha_k d_k \tag{1.2.5}$$

where the stepsize α_k is chosen to satisfy the Wolfe conditions. This iteration is quite similar to the line search Newton method, also called globalized Newton method. The key difference is that the approximate Hessian H_k is used in place of the true Hessian $G_k = \nabla^2 f(x_k)$, as already mentioned. The quasi-Newton equation requires that the symmetric positive definite matrix H_{k+1} maps s_k to y_k. This will be possible only if s_k and y_k satisfy the curvature condition

$$s_k^{\mathrm{T}} y_k > 0. \tag{1.2.6}$$

This follows from multiplying the quasi-Newton equation by s_k^{T} from the left, because we assume that H_{k+1} is positive definite. If the function f is strongly convex, then this inequality will be satisfied for any two points x_k and x_{k+1}. For nonconvex functions will this condition not always hold. In this scenario we have to impose restrictions on the line search procedure that chooses the stepsize α_k. The curvature condition holds if we impose the (strong) Wolfe conditions on the line search. Setting $s_k = x_{k+1} - x_k = \alpha_k d_k$ and using $\nabla f(x_{k+1})^{\mathrm{T}} s_k \geq c_2 \nabla f(x_k)^{\mathrm{T}} s_k$ leads to:

$$y_k^{\mathrm{T}} s_k \geq (c_2 - 1)\alpha_k g_k^{\mathrm{T}} d_k. \tag{1.2.7}$$

Since $c_2 < 1$ and d_k is a descent direction, the right side is positive and the curvature condition holds. When the curvature condition is satisfied, the quasi-Newton equation has always a solution H_{k+1}. In fact, it admits an infinite number of solutions, since the $n(n+1)/2$ degrees of freedom in a symmetric positive definite matrix exceed the conditions imposed by the quasi-Newton equation. The requirement of positive definiteness imposes n additional inequalities - all principal minors must be positive - but these conditions do not absorb the remaining degrees of freedom Nocedal, Wright, 2006.

That it makes sense that the matrices H_{k+1} satisfy the quasi-Newton equation is indicated by the Corollary Corollary 1.1.8 of Dennis and Moré. Necessary and sufficient for the superlinear convergence of the sequence $\{x_k\}_k$ to a minimizer x^* is the condition:

$$\|(G_k - H_k)(x_{k+1} - x_k)\| = o(\|x_{k+1} - x_k\|). \tag{1.2.8}$$

It can be shown that (Eq. (1.2.8)) is equivalent to

$$\|g_{k+1} - g_k - H_k(x_{k+1} - x_k)\| = o(\|x_{k+1} - x_k\|).$$

This motivates the following requirement on H_{k+1}:

$$H_{k+1}(x_{k+1} - x_k) = g_{k+1} - g_k. \tag{1.2.9}$$

We see immediately that this is the quasi-Newton equation (Eq. (1.2.2)) Geiger, Kanzow, 1999.

quasi-Newton method	Newton method
Only need the function values and gradients	Need the function values, gradients and Hessians
$\{H_k\}_k$ maintains positive definite for several updates	$\{G_k\}_k$ is not sure to be positive definite
Need $O(n^2)$ multiplications in each iteration	Need $O(n^3)$ multiplications in each iteration

Table 1.2.1: Comparison

The current theory is nevertheless sufficient to formulate a general algorithm.

Algorithm 3 General Quasi-Newton Method

1: $x_0 \in \mathbb{R}^n,\ B_0 \in \mathbb{R}^{n\times n}$ approximation of $\nabla^2 f(x_0)^{-1},\ 0 \leq \epsilon < 1,\ k = 0$
2: **while** $\|\nabla f(x_k)\| > \epsilon$ **do**
3: Compute $d_k = -B_k \nabla f(x_k)$.
4: Determine the stepsize $\alpha_k > 0$ by line search.
5: Set $x_{k+1} = x_k + \alpha_k d_k$.
6: Update B_k into B_{k+1} such that the quasi-Newton equation holds.
7: Set $k = k + 1$.
8: **end while**
9: **return** x_k

One commonly starts the algorithm with $B_0 = I$, the identity matrix or set B_0 to be a finite-difference approximation to the inverse Hessian $\nabla^2 f(x_0)^{-1}$. If $B_0 = I$, the first iteration is just a steepest descent iteration. In some cases one uses the direct approximation H_k of the Hessian. In this case we need to solve a system of equations in step 3 to get d_k and we need to update H_k instead of B_k. However, since one generally wants to do without solving a system of equations, this variant is not recommended. The resulting advantages of the quasi-Newton method over the ordinary Newton method are shown in the Table Table 1.2.1.

As Newton's method is a steepest descent method under the norm $\|\cdot\|_{G_k}$, the quasi-Newton method is a steepest descent method under the norm $\|\cdot\|_{H_k}$. In fact, d_k is the solution of the minimization problem

$$\begin{aligned} &\min \quad g_k^{\mathrm{T}} d \\ &\text{s.t.} \quad \|d\|_{H_k} \leq 1. \end{aligned} \tag{1.2.10}$$

It follows from

$$(g_k^{\mathrm{T}} d)^2 \leq (g_k^{\mathrm{T}} H_k^{-1} g_k)(d^{\mathrm{T}} H_k d)$$

that the solution of (Eq. (1.2.10)) is

$$d_k = -H_k^{-1} g_k = -B_k g_k,$$

and $g_k^{\mathrm{T}} d_k$ is the smallest vallue. By the way, since the metric matrices H_k are positive definite and always changed from iteration to iteration, the method is also called the variable metric method Sun, Yuan, 2006.

1.3 The Broyden-Fletcher–Goldfarb-Shanno Formula

We have seen that the search direction in a quasi-Newton method is given by

$$d_k = -B_k g_k = -H_k^{-1} g_k$$

and the new iterate is

$$x_{k+1} = x_k + \alpha_k d_k.$$

This iteration is quite similar to the one of Newton's method. The key difference is that the approximate Hessian H_k is used in place of the true Hessian $\nabla^2 f(x_k)$. Instead of computing H_k afresh at every iteration, Davidon proposed to update it in a simple manner to account for the curvature measured during the most recent step Nocedal, Wright, 2006. The question now is how the matrix H_{k+1} (or B_{k+1}) should be constructed from H_k (or B_k) and other information. Various formulae have been developed for this, some of which are interrelated. In this thesis the main focus is on the Broyden-Fletcher–Goldfarb-Shanno formula or short BFGS formula, which has proven to be the most efficient quasi-Newton method in practice Ulbrich, Ulbrich, 2012. However, all approaches follow the following three important guidelines to create H_{k+1}:

(*i*) H_{k+1} satisfies the quasi-Newton equation.

(*ii*) H_{k+1} is symmetric and positive definite.

(*iii*) H_{k+1} is "near" H_k.

Of course these three characteristics should also hold for the approximation of the inverse B_{k+1}. In the previous subsection was shown that H_{k+1} should satisfy the quasi-Newton equation (Eq. (1.2.2)). The strongest motivation comes from the fact that we approximate our objective function local by a quadratic model and the Hessian of a quadratic function always satisfies the quasi-Newton equation. The fact that the distance between H_{k+1} and H_k should not be too large will be related to the rate of convergence of the resulting method and the uniqueness of the formula. It's obvious that the matrix H_{k+1} should be symmetric, since we want to approximate the Hessian and the Hessian is always symmetric in the case of a twice continuously differentiable function $f \in C^2$. We need positive definiteness for efficiency, numerical stability and global convergence. If the Hessian $\nabla^2 f(x^*)$ is positive definite, the stationary point x^* is a strong minimizer. Hence, we hope the Hessian approximations $\{H_k\}_k$ (or inverse Hessian approximations $\{B_k\}_k$) are positive definite. In addition, if H_k (or B_k) is

positive definite, the local quadratic model of f has a unique local minimizer, and the direction d_k is a descent direction Sun, Yuan, 2006.
Before we get to the BFGS formula, also called BFGS update, let us first look at another one. By exchanging variables, we then get the BFGS formula. It's the so called DFP update, proposed by Davidon Davidon, 1959 and developed later by Fletcher and Powell Fletcher, Powell, 1963. We assume that the matrix B_k approximates $\nabla^2 f(x_k)^{-1}$ sufficiently well. Let us consider a symmetric rank-two update, that means we add two symmetric rank-one matrices to the current matrix

$$B_{k+1} = B_k + auu^{\mathrm{T}} + bvv^{\mathrm{T}}$$

where $u, v \in \mathbb{R}^n$, $a, b \in \mathbb{R}$ are to be determined. From the quasi-Newton equation follows

$$B_{k+1}y_k = B_k y_k + auu^{\mathrm{T}} y_k + bvv^{\mathrm{T}} y_k = s_k.$$

Clearly, u and v can not uniquely be determined. One possible choice is

$$u = s_k, \; v = B_k y_k.$$

Hence we obtain

$$a = \frac{1}{u^{\mathrm{T}} y_k} = \frac{1}{s_k^{\mathrm{T}} y_k}, \; b = -\frac{1}{v^{\mathrm{T}} y_k} = \frac{1}{y_k^{\mathrm{T}} B_k y_k}.$$

Therefore

$$B_{k+1}^{DFP} = B_k^{DFP} + \frac{s_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k} - \frac{B_k^{DFP} y_k y_k^{\mathrm{T}} B_k^{DFP}}{y_k^{\mathrm{T}} B_k^{DFP} y_k}.$$

This is the DFP update, which approximates the inverse of the Hessian $\nabla^2 f(x_k)^{-1}$ in every iteration Sun, Yuan, 2006.

The last two terms in the right-hand-side are symmetric rank-one matrices. This is the fundamental idea of quasi-Newton updating: Instead of recomputing the approximate Hessian (or inverse Hessian) from scratch at every iteration, we apply a simple modification that combines the most recently observed information about the objective function with the existing knowledge embedded in our current Hessian approximation Nocedal, Wright, 2006.

The BFGS formula can be obtained by simple trick: for H_{k+1}^{BFGS} replace the triple (B_k^{DFP}, s_k, y_k) in B_{k+1}^{DFP} by (H_k^{BFGS}, y_k, s_k). Thus, BFGS update is also said to be a complement DFP update. The result is

$$H_{k+1}^{BFGS} = H_k^{BFGS} + \frac{y_k y_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k} - \frac{H_k^{BFGS} s_k s_k^{\mathrm{T}} H_k^{BFGS}}{s_k^{\mathrm{T}} H_k^{BFGS} s_k}. \tag{1.3.1}$$

Sun, Yuan, 2006

This formula was discovered independently by Broyden Broyden, 1967, Fletcher Fletcher, 1970, Goldfarb Goldfarb, 1970 and Shanno Shanno, 1970, which is the reason for the name. All four authors derive the BFGS formula in a slightly different way, which can be seen as a reason why it is superior to the other updating formulae in practice Geiger, Kanzow, 1999. It is presently considered to be the most effective of all quasi-Newton updating formulae Nocedal, Wright, 2006. The DFP update is quite effective, but it was soon superseded by the BFGS formula, which has all good properties of the DFP update Sun, Yuan, 2006.

Since $H_k s_k = -\alpha_k g_k$ and $H_k d_k = -g_k$, this formula can also be written as

$$H_{k+1}^{BFGS} = H_k^{BFGS} + \frac{g_k g_k^{\mathrm{T}}}{g_k^{\mathrm{T}} d_k} + \frac{y_k y_k^{\mathrm{T}}}{\alpha_k y_k^{\mathrm{T}} d_k}.$$

By applying the the Sherman–Morrison–Woodbury formula twice to (Eq. (1.3.1)), we obtain

$$\begin{aligned} B_{k+1}^{BFGS} &= B_k^{BFGS} + \frac{(s_k - B_k^{BFGS} y_k) s_k^{\mathrm{T}} + s_k (s_k - B_k^{BFGS} y_k)^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k} - \frac{(s_k - B_k^{BFGS} y_k)^{\mathrm{T}} y_k s_k s_k^{\mathrm{T}}}{(s_k^{\mathrm{T}} y_k)^2} \\ &= B_k^{BFGS} + \left(\mathrm{I} + \frac{y_k^{\mathrm{T}} B_k^{BFGS} y_k}{s_k^{\mathrm{T}} y_k}\right) \frac{s_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k} - \frac{s_k y_k^{\mathrm{T}} B_k^{BFGS} + B_k^{BFGS} y_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k} \\ &= \left(I - \frac{s_k y_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}\right) B_k^{BFGS} \left(I - \frac{y_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}\right) + \frac{s_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}. \end{aligned} \tag{1.3.2}$$

These are different formulae for the approximation of the Hessian inverse. Furthermore, reference is also made to this with BFGS formula. It is easy to see that (Eq. (1.3.2)) is also a rank-two modification of B_k^{BFGS}. One can easily show that

$$H_{k+1}^{BFGS} B_{k+1}^{BFGS} = B_{k+1}^{BFGS} H_{k+1}^{BFGS} = I.$$

Replacing the triple (B_k^{BFGS}, s_k, y_k) in (Eq. (1.3.2)) by (H_k^{DFP}, y_k, s_k), one would get a formula for H_{k+1}^{DFP}, the direct DFP update. This describes a method for finding its dual update from a given update. For this reason, the DFP and BFGS formulae are sometimes referred to as "dual" updating formulae Sun, Yuan, 2006.
Now it is checked that H_{k+1}^{BFGS} meets the given characteristics. For 1. and 2. there is the following statement:

Theorem 1.3.1 (Ulbrich, Ulbrich, 2012, Theorem 13.4). *(i) If $y_k^T s_k \neq 0$ and $s_k^T H_k^{BFGS} s_k \neq 0$ holds, the matrices $H_{k+1}^{BFGS} \in \mathbb{R}^{n\times n}$ are well defined, symmetric and satisfy the quasi-Newton equation (Eq. (1.2.2)).*

(ii) If H_k^{BFGS} is positive definite and $y_k^T s_k > 0$, then H_{k+1}^{BFGS} is positive definite.

Such an update is also called positive definite update. The same holds of course for the approximation of the inverse B_{k+1}^{BFGS}. In the previous subsection was shown that the curvature condition (Eq. (1.2.6)) must hold. This was achieved by imposing restrictions on the line search method (Eq. (1.2.7)). So the positive definiteness can be guaranteed just by a Wolfe line search strategy. The statement was actually made for Broyden class matrices (the matrices are a convex combination of DFP and BFGS matrices) in Ulbrich, Ulbrich, 2012, this means that it can be transferred one-to-one to the DFP update H_{k+1}^{DFP}.

The last characteristic, that H_{k+1} should be "near" H_k, has a far more powerful meaning than the other two. Many authors use only this to define the BFGS formula which of course is perfectly legitimate. As already mentioned, this property leads to the fact that the formula can be considered as unique and it has something to do with the rate of convergence. The two go hand in hand.
One wants the BFGS method to be similar to Newton's method in terms of convergence. This means that it should converge superlinearly. This can be proven by the Dennis-Moré condition. The connection between this condition and the characteristic is shown in Ulbrich, Ulbrich, 2012 by

Lemma 1.3.2 (Ulbrich, Ulbrich, 2012, Lemma 13.2). *x^* fulfils the sufficient condition of second order. Algorithm Algorithm 3 with $\alpha_k = 1$ for all $k \in \mathbb{N}$ generates a sequence $\{x_k\}_k$ convergent to x^* and also holds*

$$\lim_{k\to\infty} \|H_{k+1} - H_k\| = 0,$$

then H_k satisfies the Dennis-Moré condition and $\{x_k\}_k$ converges q-superlinear to x^*.

Therefore one looks for quasi-Newton updates for which H_{k+1} is close to H_k in each iteration, so that the distance between them converges towards zero.
We would now like to consider the third property (H_{k+1} is "near" H_k) from the point of view of the uniqueness of the formula. One obtains this by considering the formula for B_{k+1}^{BFGS} as the solution to an optimization problem. More information can be found in Geiger, Kanzow, 1999 subsection 11.1 and Nocedal, Wright, 2006, subsection 6.1. The derivation with the optimization problem is again closely related to the DFP update, but as said before, more about this can be found in the mentioned sources. The following two statements provide us with the uniqueness of the BFGS formula.

Lemma 1.3.3 (Geiger, Kanzow, 1999, Lemma 11.7). *Let be $s \in \mathbb{R}^n$, $y \in \mathbb{R}^n$ with $y \neq 0$ and a symmetric matrix $B \in \mathbb{R}^{n\times n}$ given. Furthermore let $W \in \mathbb{R}^{n\times n}$ be symmetric and positive definite. Then the unique solution of the inverse weighted problem*

$$\begin{aligned} \min_{B_+} \quad & \|W(B_+ - B)W\|_F^2 \\ s.t. \quad & B_+ = B_+^T, \qquad B_+ y = s \end{aligned} \tag{1.3.3}$$

is given by

$$B_+^W = B + \frac{(s-By)(W^{-2}y)^{\mathrm{T}} + W^{-2}y(s-By)^{\mathrm{T}}}{(W^{-2}y)^{\mathrm{T}}y} - \frac{y^{\mathrm{T}}(s-By)W^{-2}y(W^{-2}y)^{\mathrm{T}}}{((W^{-2}y)^{\mathrm{T}}y)^2}.$$

In order to be able to convert this to the BFGS formula one is looking for the so-called weighting matrix W:

Theorem 1.3.4 (Geiger, Kanzow, 1999, Theorem 11.8). *Let $B \in \mathbb{R}^{n\times n}$ be symmetric and positive definite and $s, y \in \mathbb{R}^n$ with $s^{\mathrm{T}}y > 0$. Let $Q \in \mathbb{R}^{n\times n}$ be a symmetric and positive definite matrix with $Qs = y$, and let $W = Q^{\frac{1}{2}}$ be a square root of Q. Then the unique solution of the inverse weighted problem (Eq. (1.3.3)) with the weighted W is given by*

$$B_+^{BFGS} = B + \frac{(s-By)s^{\mathrm{T}} + s(s-By)^{\mathrm{T}}}{y^{\mathrm{T}}s} - \frac{(s-By)^{\mathrm{T}}yss^{\mathrm{T}}}{(y^{\mathrm{T}}s)^2}. \tag{1.3.4}$$

One can choose $Q = W^2 = \tilde{G}_k$ and $\tilde{G}_k$ is the average Hessian, i.e.

$$\tilde{G}_k = \int_0^1 \nabla^2 f(x_k + \tau s_k)d\tau = \int_0^1 \nabla^2 f(x_k + \tau\alpha_k d_k)d\tau,$$

which is positive definite for a strong convex function. The property

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k) = \int_0^1 \nabla^2 f(x_k + \tau s_k)s_k d\tau = \tilde{G}_k s_k$$

follows from Taylor's formula Sun, Yuan, 2006. With this choice of weighting matrix W, the norm

$$\|A\|_{W^2} = \|WAW\|_{\mathrm{F}}$$

is non-dimensional, which is a desirable property, since we do not wish the solution of (Eq. (1.3.3)) to depend on the units of the problem Nocedal, Wright, 2006. The existence of Q follows from Lemma 11.5. in Geiger, Kanzow, 1999 and since it is symmetric and positiv definite, the existence of a spd matrix W follows from it, see Theorem B.6. in Geiger, Kanzow, 1999. The specified minimum characteristic with respect to the weighted norms mentioned in the theorem automatically ensures the invariance of the BFGS method under affin-linear variable transformations. This important characteristic is also present in the Newton method Ulbrich, Ulbrich, 2012.
The initial approximation B_0^{BFGS} must still be discussed. Unfortunately, there is no perfect strategy for this yet. One possibility is to use information about the problem and approximate the Hessian inverse

by finite differences at x_0. One could also use a multiple of the identity matrix βI, where β is a scaling factor for the variables. But to determine this factor is problematic. If β is too large, so that the first step $d_0 = -\beta g_0$ is too long, many function evaluations may be required to find a suitable value for the stepsize α_0. A quite effective heuristic is to scale the starting matrix after the first step, but before the first BFGS update is performed. The provisional value $B_0^{BFGS} = I$ is changed by setting

$$B_0^{BFGS} = \frac{y_1^T s_1}{y_1^T y_1} I$$

before applying the update to obtain B_1^{BFGS}. This formula attempts to make the size of B_0^{BFGS} similar to that of $\nabla^2 f(x_0)^{-1}$ Nocedal, Wright, 2006.

1.4 The BFGS-Method

~~After all the preparations we now come to the actual method~~. We present a globalized BFGS method ~~where we calculate a stepsize α_k in each iteration and multiply the direction d_k by it. If a step size is not calculated, i.e. setting $\alpha_k = 1$ in every iteration, it is called local BFGS method. Global convergence cannot be shown for it. But on the other hand, superlinear convergence can be proved. The convergence analysis of the local BFGS method is very complex and would go beyond the scope of this thesis. Statements about the convergence rate and the derivation of these results can be found in Geiger, Kanzow, 1999, 11.3 Lokale Konvergenz des BFGS-Verfahrens.~~

For the globalized method, it must be ensured that the curvature condition Eq. (1.2.6) is fulfilled. ~~It is therefore important that a stepsize strategy is chosen to ensure that~~ by choosing the step size accordingly. The globalization of the BFGS method is similar to the globalization of Newtons method. ~~But~~ In contrast to it, not the Armijo rule (a stepsize strategy which determines a stepsize α_k satisfying Eq. (1.1.2)) is chosen here, but the Wolfe-Powell stepsize strategy, which ensures $s_k^T y_k > 0$ for all $k \in \mathbb{N}$. This type of line search is called inexact line search, approximate line search or acceptable line search. The stepsize $\alpha_k > 0$ is chosen such that the objective function has an acceptable descent amount, i.e., such that the descent $f(x_k) - f(x_k + \alpha_k d_k) > 0$ is acceptable. In contrast to this, there is the so called exact line search or optimal line search where a $\alpha_k > 0$ is chosen such that the objective function in the direction d_k is minimized, i.e., $f(x_k + \alpha_k d_k) = \min_{\alpha > 0} f(x_k + \alpha d_k)$ Sun, Yuan, 2006, p. 71.

Since, in practical computation, an exact optimal stepsize generally cannot be found, and it is also expensive to find almost an exact stepsize, therefore the inexact line search with less computation load is highly popular Sun, Yuan, 2006, p. 72. Also the convergence rate does not depend on the exact line search. Therefore, as long as there is an acceptable stepsize rule which ensures that the objective function has sufficient descent, the exact line search can be avoided and the computing efforts will be decreased greatly Sun, Yuan, 2006, p. 102.

We call the following algorithm "Inverse Global BFGS Method" because the updating formula for the approximation of the inverse of the Hessian ($B_k^{BFGS} \mapsto B_{k+1}^{BFGS}$) is used, since we are spared the solving of a system of equations and we only have to work with matrix-vector-multiplications. The algorithm could be formulated with the approximation of the actual Hessian H_k^{BFGS}, but that would increase the effort again to $O(n^3)$, which is not desirable Nocedal, Wright, 2006, p. 141. In practice, it must be decided whether solving a system of equations or matrix-vector-multiplication is more

advantageous for the underlying problem. We assume in this thesis that the latter is the better choice. In the following we will express the results with the approximation of the inverse, B_k^{BFGS}, even if in the original literature the direct update formula Eq. (1.3.1) is used. This serves as a simplification and does not lead to any complications, because the proofs and thus the general theory do not depend on which variant is used.

Algorithm 4 Inverse Global BFGS Method

1: Given starting point $x_0 \in \mathbb{R}^n$, convergence tolerance $\epsilon > 0$, an initial symmetric and positive definite matrix $B_0^{BFGS} \in \mathbb{R}^{n\times n}$, $k = 0$.
2: **while** $\|\nabla f(x_k)\| > \epsilon$ **do**
3: Compute search direction $d_k = -B_k^{BFGS}\nabla f(x_k)$.
4: Find a stepsize α_k that satisfies the (strong) Wolfe conditions.
5: Set $x_{k+1} = x_k + \alpha_k d_k,\ s_k = x_{k+1} - x_k,\ y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.
6: Compute B_{k+1}^{BFGS} by means of Eq. (1.3.2).
7: Set $k = k + 1$.
8: **end while**
9: **return** x_k

It can be shown that Algorithm 4 is well defined:

Theorem 1.4.1 (Geiger, Kanzow, 1999, Theorem 11.37). *Be $f : \mathbb{R}^n \to \mathbb{R}$ continuously differentiable and bounded below. Then for the globalized BFGS method Algorithm 4:*

(i) $s_k^T y_k > 0$ *for all* $k \in \mathbb{N}$.

(ii) *The matrices* B_{k+1}^{BFGS} *are symmetric and positive definite for all* $k \in \mathbb{N}$.

(iii) *The method is well defined.*

The derivation to this theorem in Geiger, Kanzow, 1999 shows that finding a stepsize α_k, which satisfies the Wolfe conditions (Eq. (1.1.2) and Eq. (1.1.3)) or strong Wolfe conditions (Eq. (1.1.2) and Eq. (1.1.4)), is crucial Geiger, Kanzow, 1999, p. 166. Eq. (1.2.7) shows that the stepsize ensures the curvature condition $s_k^T y_k > 0$. This in turn ensures that the positive definiteness of the matrix B_k^{BFGS} is passed to B_{k+1}^{BFGS}, see Theorem 1.3.1.
Let us now turn to the convergence analysis of Algorithm 4. It is desirable that each limit point of a sequence $\{x_k\}_k$ generated by Algorithm 4 is a stationary point of f and that we get locally superlinear convergence. Unfortunately, neither of these statements is in general true Geiger, Kanzow, 1999, p. 167.
Let us first deal with the global convergence. The difficulty and importance of the convergence problem of whether the BFGS method with the Wolfe line search converges globally for general functions has been addressed in many situations. But recent studies provide a negative answer to it for nonconvex functions (see e.g. Dai, 2002 or Mascarenhas, 2004) Dai, 2012, p. 3.
Consequently, we are not able to establish truly global convergence results for general nonlinear objective functions. We cannot prove that the iterates $\{x_k\}_k$ generated by Algorithm 4 approach a stationary point x^* of the problem from any starting point x_0 and any (suitable) initial Hessian inverse approximation B_0^{BFGS} Nocedal, Wright, 2006, p. 153.
We present a statement about global convergence from Nocedal, Wright, 2006, which is based on results of Powell, 1976 and is also very common. For that we require that the objective function is

convex. To be more precise, the following assumptions must be made for a reasonable convergence statement:

Assumption 1.4.2 (Nocedal, Wright, 2006, Assumption 6.1.). (i) *The objective function f is twice continuously differentiable.*

(ii) *The level set $\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is convex, and there exist positive constants m and M such that*

$$m\|z\|^2 \leq z^{\mathrm{T}} \nabla^2 f(x) z \leq M\|z\|^2$$

for all $z \in \mathbb{R}^n$ and $x \in \mathcal{L}$.

The second part of Assumption 1.4.2 implies that $\nabla^2 f(x)$ is positive definite on $\mathcal{L}$ and that f has an unique minimizer x^* in $\mathcal{L}$ Nocedal, Wright, 2006, p. 153. The following theorem is the best possible statement for global convergence under Assumption 1.4.2:

Theorem 1.4.3 (Nocedal, Wright, 2006, Theorem 6.5.). *Let H_0 be any symmetric positive definite initial matrix, and let x_0 be a starting point for which Assumption 1.4.2 is satisfied. Then the sequence $\{x_k\}_k$ generated by Algorithm 4 (with $\epsilon = 0$) converges to the minimizer x^* of f.*

The proof can be found in Nocedal, Wright, 2006, p. 154. Nevertheless, it should be noted that Theorem 1.1.6, the Zoutendijk conditionm, is crucial to prove global convergence.
We see that the BFGS Method with the Wolfe-Powell stepsize strategy applied to a smooth convex function f from an arbitrary starting point $x_0 \in \mathbb{R}^n$ and from any initial approximation $B_0^{BFGS} \in \mathbb{R}^{n \times n}$, that is symmetric and positive definite, is globally convergent. This is a very strong convergence result for the BFGS method, and it is currently not known whether this also applies to the DFP method Nocedal, Wright, 2006, p. 156.
It can also be shown that Algorithm 4 not only converges globally by using the Wolfe-Powell stepsize strategy but also by using a great number of inexact, efficient stepsize strategies used in practice for uniformly convex objective functions, which can be seen as an indication of the numerical stability of this method Werner, 1978/79, p. 327.
~~Now let's talk about the rate of convergence of Algorithm 4~~. As mentioned at the beginning, the local BFGS method achieves superlinear convergence to a stationary point second order Geiger, Kanzow, 1999, Satz 11.33. But in the local method, the step size is always equal one, i.e. $\alpha_k = 1, \ \forall k$. The crux of the matter is that this step size must satisfy the Wolfe conditions to be accepted and thus we get superlinear convergence for Algorithm 4.
In practical implementations Algorithm 4 this unit stepsize is usually used as the first trial stepsize. Under suitable assumptions on f, this stepsize will be accepted by the line search as the iterates tend to the solution and will enable superlinear convergence Dai, 2012, p. 6. In Dennis, Moré, 1974 this idea is pursued and a detailed derivation is given. But we want to present a very well known result from Nocedal, Wright, 2006, which applies to general nonlinear objective functions and is also based on the results of Dennis, Moré, 1974. ~~The reason for this will be shown later when we discuss the BFGS method for manifolds.~~

Assumption 1.4.4 (Nocedal, Wright, 2006, Assumption 6.2.). *The Hessian matrix is Lipschitz continuous at x^*, that is,*

$$\|\nabla^2 f(x) - \nabla^2 f(x^*)\| \leq L\|x - x^*\|,$$

for all x near x^*, where L is a positive constant.

~~With this assumption we are in a position to prove superlinear convergence. The analysis of the theorem's proof makes use of the Theorem 1.1.7 characterization of superlinear convergence.~~

Theorem 1.4.5 (Nocedal, Wright, 2006, Theorem 6.6.). *Suppose that f is twice continuously differentiable and that the iterates $\{x_k\}_k$ generated by Algorithm 4 converge to a minimizer x^* at which Assumption 1.4.4 holds. Suppose also that*

$$\sum_{k=0}^{\infty} \|x - x^*\| < \infty \tag{1.4.1}$$

holds. Then $\{x_k\}_k$ converges to x^* at a superlinear rate.

offen?

In practice it is not seldom observed that the sequence of matrices $\{H_k^{BFGS}\}_k$ generated by Algorithm 4 using the direct formula Eq. (1.3.1) converges ~~against~~ $\nabla^2 f(x^*)$ ~~in a solution~~ x^* ~~of the optimization problem~~. In this case, the superlinear convergence follows directly from Eq. (1.1.9). ~~However, one has to make sure again that the stepsize $\alpha_k = 1$ is always chosen, as long as it is sufficient for the Wolfe conditions.~~

In fact, the convergence of $\{H_k^{BFGS}\}_k$ against the exact Hesse matrix $\nabla^2 f(x^*)$ ~~can be proved for certain~~ classes of functions, not only for the BFGS method, but for a ~~whole~~ series of further quasi-Newton methods. The problem is, for generic functions ~~the~~ convergence of the sequence $\{H_k^{BFGS}\}_k$ against $\nabla^2 f(x^*)$ cannot be proven. ~~Even if the~~ sequence $\{H_k^{BFGS}\}_k$ converges, it does not necessarily converge against the Hessian $\nabla^2 f(x^*)$. Fortunately, this is not necessary to prove superlinear convergence Geiger, Kanzow, 1999, p. 167-168.

1.5 A Cautious BFGS-Method

In Dai, 2012 a four-dimensional example where the cost function is smooth (polynomial) and nonconvex was presented such that the globalized BFGS method does not converge. From this it can be concluded that Algorithm 4 unfortunately does not converge in general. In this subsection we present a simple modification of the globalized BFGS method, which allows us to establish a global convergence theorem for nonconvex problems.

We present the method from Li, Fukushima, 2001, which modifies the method by a so-called cautious update. It can be shown that this method with a line search, which satisfies the Wolfe conditions, converges globally if the function to be minimized has Lipschitz continuous gradients. Moreover, under appropriate conditions, it can be shown that the cautious update eventually reduces to the ordinary update, i.e. B_k^{BFGS} is updated using Eq. (1.3.2). ~~In this thesis we express the method with the~~

approximation of the inverse. Again, it has no influence on the convergence results and is used for simplicity.
A good property of Eq. (1.3.2) is, that B_{k+1}^{BFGS} inherits the positive definiteness of B_k^{BFGS} as long as the curvature condition Eq. (1.2.6) holds, see Theorem 1.3.1. The curvature condition, $s_k^T y_k$, is guaranteed to hold if the stepsize α_k is determined by a stepsize strategy which satisfies the (strong) Wolfe conditions (Eq. (1.1.2) and Eq. (1.1.3) or Eq. (1.1.2) and Eq. (1.1.4)).
Another well-known step size strategy is the Armijo rule. It provides a stepsize α_k which only satisfies Eq. (1.1.2). This α_k does not ensure the curvature condition Eq. (1.2.6) and hence B_{k+1}^{BFGS} is not necessarily positive definite even if B_k^{BFGS} is positive definite. To ensure the positive definiteness, Eq. (1.2.6) is sometimes used to decide whether or not B_k^{BFGS} is updated, i.e. B_{k+1}^{BFGS} is determined by

$$B_{k+1}^{BFGS} = \begin{cases} \text{using Eq. (1.3.2)} & y_k^T s_k > 0 \\ B_k^{BFGS} & \text{otherwise.} \end{cases}$$

Dong-Hui Li and Masao Fukushima presented in Li, Fukushima, 2001 a cautious update rule similar to the above and establish a global convergence theorem for nonconvex optimization problems. The motivation for this modification comes from the following lemma:

Lemma 1.5.1 (Li, Fukushima, 2001, Lemma 2.1). *If the BFGS method with a line search that satisfies the Wolfe conditions is applied to a continuously differentiable function f that is bounded below, and if there exists a constant $M > 0$ such that the inequality*

$$\frac{\|y_k\|^2}{y_k^T s_k} \leq M \tag{1.5.1}$$

holds for all k, then

$$\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

If f is twice continuously differentiable and convex, then Eq. (1.5.1) always holds whenever the sequence of iterates $\{x_k\}_k$ is bounded. Therefore, global convergence of the BFGS method follows immediately from Lemma 1.5.1.
If f is nonconvex, it seems difficult to guarantee Eq. (1.5.1). For that, in Li, Fukushima, 2001 a cautious update was introduced:

$$B_{k+1}^{CBFGS} = \begin{cases} \text{using Eq. (1.3.2)} & \frac{y_k^T s_k}{\|s_k\|^2} \geq \mu \|\nabla f(x_k)\|^\lambda \\ B_k^{CBFGS} & \text{otherwise.} \end{cases} \tag{1.5.2}$$

where μ and λ are positive constants. Using this update rule, we get the following algorithm:

Algorithm 5 Cautious BFGS Algorithm

1: Given starting point $x_0 \in \mathbb{R}^n$, convergence tolerance $\epsilon > 0$, an initial symmetric and positive definite matrix $B_0^{CBFGS} \in \mathbb{R}^{n\times n}$, choose constants $\lambda > 0$ and $\mu > 0$, $k = 0$.
2: **while** $\|\operatorname{grad} f(x_k)\| > \epsilon$ **do**
3: Compute search direction $d_k = -B_k^{CBFGS}\nabla f(x_k)$.
4: Find a stepsize α_k that satisfies the (strong) Wolfe conditions.
5: Set $x_{k+1} = x_k + \alpha_k d_k,\ s_k = x_{k+1} - x_k,\ y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.
6: Compute B_{k+1}^{CBFGS} by means of Eq. (1.5.2).
7: Set $k = k + 1$.
8: **end while**
9: **return** x_k

With Eq. (1.5.2) we see immediately that the matrices $\{B_k^{CBFGS}\}_k$ generated by the Algorithm 5 are all positive definite, see Theorem 1.3.1. This implies that $\{f(x_k)\}_k$ is a decreasing sequence for $\alpha_k > 0$ in every step.
We now come to the convergence analysis of this method. In Li, Fukushima, 2001 it is shown that it converges globally with weaker assumptions than those of the general globalized BFGS method. An acceptable rate of convergence is also shown. The weaker assumptions are:

Assumption 1.5.2 (Li, Fukushima, 2001, Assumption A)**.** *The level set*

$$\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$$

is bounded, the function f is continuously differentiable on $\mathcal{L}$, and there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall\, x, y \in \mathcal{L}.$$

We see that there is no longer a requirement that the sublevel set $\mathcal{L}$ must be convex. Instead, the assumption about Lipschitz continuous gradients was added. Since $\{f(x_k)\}_k$ is a decreasing sequence, it follows that the sequence $\{x_k\}_k$ generated by Algorithm 5 is contained in $\mathcal{L}$. We have the following statement on global convergence:

Theorem 1.5.3 (Li, Fukushima, 2001, Theorem 3.3.)**.** *Let Assumption 1.5.2 hold and $\{x_k\}_k$ be generated by Algorithm 5. Then*

$$\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0$$

holds.

Theorem 1.5.3 shows that there exists a subsequence of $\{x_k\}_k$ converging to a stationary point x^* of Eq. (1.1.1). If f is convex, then x^* is a global minimum of f. Since the sequence $\{f(x_k)\}_k$ converges, it is clear that every accumulation point of $\{x_k\}_k$ is a global optimal solution of Eq. (1.1.1).

cluster?

Corollary 1.5.4 (Li, Fukushima, 2001, Corollary 3.4.). *Let Assumption 1.5.2 hold and $\{x_k\}_k$ be generated by Algorithm 5. If f is convex, then the whole sequence $\{\nabla f(x_k)\}_k$ converges to zero. Consequently, every accumulation point of $\{x_k\}_k$ is a global optimal solution.*

informell $\nabla f \to 0$
$\|\nabla f\|$ conv. to zero.

~~If f is not convex, then Corollary 1.5.4 is not guaranteed. If some additional assumptions are made, then the whole sequence $\{x_k\}_k$ converges to a local optimal solution~~:

This can be generalized to nonconvex functions as follows.

Theorem 1.5.5 (Li, Fukushima, 2001, Theorem 3.5.). *Let f be twice continuously differentiable. Suppose that $\lim_{k\to\infty} s_k = 0$. If there exists an accumulation point x^* of $\{x_k\}_k$ at which $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite, then the whole sequence $\{x_k\}_k$ converges to x^*. If in addition, $\nabla^2 f(\cdot)$ is Hölder continuous and $c_1 \in (0, 0.5)$ holds, then the convergence rate is superlinear.*

The assumptions imply that x^* is a strict local optimal solution. In the proof is shown that Algorithm 5 is moving to Algorithm 4, that means when k is sufficiently large, the condition $y_k^T s_k / \|s_k\|^2 \geq \mu \|\nabla f(x_k)\|^\lambda$ is always satisfied, which implies that the algorithm reduces to the "ordinary" BFGS method. The superlinear convergence then follows from the results of the convergence analysis of Algorithm 4.

In Li, Fukushima, 2001 it is also noted that the parameters need not be constant while the method is running. By choosing $\lambda = 0.01$ if $\|\nabla f(x_k)\| \geq 1$ and $\lambda = 3$ if $\|\nabla f(x_k)\| < 1$ it is tried to make the cautious update Eq. (1.5.2) closer to the original BFGS update Eq. (1.3.2). Another option is that λ can also be within an interval $[a, b]$ with $a > 0$. More generally, the value $\mu\|\nabla f(x_k)\|^\lambda$ can be replaced by a general forcing function $\theta(\|\nabla f(x_k)\|)$, which is strictly monotone with $\theta(0) = 0$. For all these variants of parameter selection or adjustment of the cautious trigger, the above-mentioned convergence results hold.

A (kein englisch)

increasing?

Numerical results in Li, Fukushima, 2001 show that Algorithm 5 is compatible with Algorithm 4. It is observed that the trigger in the cautious update was usually satisfied what implies that Algorithm 4 is generally "cautious" and hence the method seldom fails in practice. It was also observed that the choice of the parameter λ affects the performance of the method. If λ is chosen appropriately, then $y_k^T s_k / \|s_k\|^2 \geq \mu \|\nabla f(x_k)\|^\lambda$ is almost always satisfied and Algorithm 5 essentially reduces to Algorithm 4. When λ was chosen poorly in the tests, which means that $y_k^T s_k / \|s_k\|^2 \geq \mu \|\nabla f(x_k)\|^\lambda$ was often violated, then the performance of Algorithm 5 was worse than the performance of Algorithm 4, even failing to converge.

not wording?

A (kein englisch)

verstehe ich nicht

1.6 Limited-Memory BFGS-Method

One of the disadvantages of the Quasi-Newton methods is that a $n \times n$ matrix (namely B_{k+1}^{BFGS}) must be stored in each iteration. Even when using the symmetry of this matrix, a memory requirement of $n(n+1)/2$ matrix entries remains. For large-scale optimization problems is not feasible Geiger,

Kanzow, 1999, p. 197.
Limited-memory quasi-Newton methods, also called variable-storage quasi-Newton methods, are useful for solving large problems whose Hessian matrices cannot be computed at a reasonable cost or are not sparse. The methods save only a few n-dimensional vectors, instead of storing and computing fully dense $n \times n$ approximations of the Hessian. The main idea is to use the curvature information (the information about the curvature of the model, which is obtained by approximating the Hessian) from only the most recent iterations to construct the Hessian approximation. Curvature information from earlier iterations, which is less likely to be relevant to the actual behavior of the Hessian at the current iteration, is discarded in the interest of saving storage. Their rate of convergence is often acceptable (albeit linear), despite these modest storage requirements Nocedal, Wright, 2006.
Due to the outstanding importance of the BFGS-method in the class of quasi-Newton methods Geiger, Kanzow, 1999, p. 197, it is also predominantly used as a limited-memory variant, called L-BFGS. But there are also limited-memory versions of other quasi-Newton methods such as the Symmetric Rank-One (SR1) method Nocedal, Wright, 2006, p. 177.
In subsection 1.3 three different formulae for the approximation of the Hessian inverse were introduced, see Eq. (1.3.2). We start from the last one

$$B_{k+1}^{BFGS} = \left(I - \frac{s_k y_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}\right) B_k^{BFGS} \left(I - \frac{y_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}\right) + \frac{s_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}.$$

For given vectors $s_k, y_k \in \mathbb{R}^n$ with $s_k^{\mathrm{T}} y_k > 0$ one sets

$$\rho_k = \frac{1}{s_k^{\mathrm{T}} y_k}, \quad V_k = I - \rho_k y_k s_k^{\mathrm{T}}, \tag{1.6.1}$$

obtaining

$$B_{k+1}^{BFGS} = V_k^{\mathrm{T}} B_k^{BFGS} V_k + \rho_k s_k s_k^{\mathrm{T}}. \tag{1.6.2}$$

The matrix B_{k+1}^{BFGS} is obtained by updating B_k^{BFGS} using the pair $\{s_k, y_k\}$ Sun, Yuan, 2006, p. 293. Since the inverse Hessian approximation B_k^{BFGS} will generally be dense, the cost of storing and manipulating it is prohibitive when the number of variables is large. To circumvent this problem, one stores a modified version of B_k^{BFGS} implicitly, by storing a certain number (say, m) of the vector pairs $\{s_i, y_i\}$ Nocedal, Wright, 2006, p. 177. After the new iterate is x_{k+1} computed, the oldest vector pair in the set $\{s_i, y_i\}_{i=k-m}^{k-1}$ (namely $\{s_{k-m}, y_{k-m}\}$) is discarded and the new pair $\{s_k, y_k\}$ obtained from the current step is added. This works accorcing to the strategy "first in, first out", which means that the vectors that were imported first are also discarded first. In this way, the set of vector pairs includes curvature information from the m most recent iterations. Practical experience has shown that modest values of m (between 3 and 20) often produce satisfactory results. The strategy of keeping the m most recent pairs $\{s_i, y_i\}_{i=k-m}^{k-1}$ works well in practice. Indeed no other strategy has yet proved to be consistently better Nocedal, Wright, 2006, p. 179. Normally, for large-scale problems, one takes $m \ll n$. In practice, the choice of m depends on the dimension of the problem and the storage of the employed computer

Sun, Yuan, 2006, p. 295.
The update process in detail: at iteration k, the current iterate is x_k and the set of vector pairs is given by $\{s_i, y_i\}_{i=k-m}^{k-1}$. At first some initial Hessian approximation $B_k^{(0)}$ is choosen for the k-th iteration (in contrast to the standard BFGS iteration, this initial approximation is allowed to vary from iteration to iteration). The formula Eq. (1.6.2) is applied m times repeatedly, i.e.

$$B_k^{(j+1)} = V_{k-m+j}^{\mathrm{T}} B_k^{(j)} V_{k-m+j} + \rho_{k-m+j} s_{k-m+j} s_{k-m+j}^{\mathrm{T}}, \ \ j = 0, 1, \cdots, m-1. \tag{1.6.3}$$

The L-BFGS approximation, called B_k^{L-BFGS}, reads the following:

$$\begin{aligned} B_k^{L-BFGS} = B_k^{(m)} =& V_{k-1}^{\mathrm{T}} B_k^{(m-1)} V_{k-1} + \rho_{k-1} s_{k-1} s_{k-1}^{\mathrm{T}} = \\ =& \cdots = \\ =& (V_{k-1}^{\mathrm{T}} \cdots V_{k-m}^{\mathrm{T}}) B_k^{(0)} (V_{k-m} V_{k-m+1} \cdots V_{k-1}) + \\ &+ \rho_{k-m} (V_{k-1}^{\mathrm{T}} \cdots V_{k-m+1}^{\mathrm{T}}) s_{k-m} s_{k-m}^{\mathrm{T}} (V_{k-m+1} \cdots V_{k-1}) \\ &+ \rho_{k-m+1} (V_{k-1}^{\mathrm{T}} \cdots V_{k-m+2}^{\mathrm{T}}) s_{k-m+1} s_{k-m+1}^{\mathrm{T}} (V_{k-m+2} \cdots V_{k-1}) \\ &+ \cdots \\ &+ \rho_{k-1} s_{k-1} s_{k-1}^{\mathrm{T}}. \end{aligned}$$

That means B_k^{L-BFGS} can be calculated completely from $B_k^{(0)}$ and the vector pairs $\{s_i, y_i\}_{i=k-m}^{k-1}$. B_k^{L-BFGS} must be considered as an approximation of B_k^{BFGS}. Nevertheless this matrix fulfils the Quasi-Newton equation Eq. (1.2.2) Geiger, Kanzow, 1999.
In fact, there is no need to compute and save B_k^{L-BFGS} explicitly. Instead, one only saves the pairs $\{s_i, y_i\}_{i=k-m}^{k-1}$ and computes $B_k^{L-BFGS} g_k = B_k^{L-BFGS} \nabla f(x_k)$ Sun, Yuan, 2006. The product can be obtained by performing a sequence of inner products and vector summations involving g_k and the pairs $\{s_i, y_i\}_{i=k-m}^{k-1}$ Nocedal, Wright, 2006. So, we have

$$\begin{aligned} B_k^{L-BFGS} g_k = &(V_{k-1}^{\mathrm{T}} \cdots V_{k-m}^{\mathrm{T}}) B_k^{(0)} (V_{k-m} V_{k-m+1} \cdots V_{k-1}) g_k + \\ &+ \rho_{k-m} (V_{k-1}^{\mathrm{T}} \cdots V_{k-m+1}^{\mathrm{T}}) s_{k-m} s_{k-m}^{\mathrm{T}} (V_{k-m+1} \cdots V_{k-1}) g_k \\ &+ \rho_{k-m+1} (V_{k-1}^{\mathrm{T}} \cdots V_{k-m+2}^{\mathrm{T}}) s_{k-m+1} s_{k-m+1}^{\mathrm{T}} (V_{k-m+2} \cdots V_{k-1}) g_k \\ &+ \cdots \\ &+ \rho_{k-1} s_{k-1} s_{k-1}^{\mathrm{T}} g_k. \end{aligned}$$

Since $V_i g_k = (I - \rho_i y_i s_i^{\mathrm{T}}) g_k$ for $i = k-1, \cdots, k-m$, one can derive a recursive method to compute the product efficiently Sun, Yuan, 2006, p. 293. See the L-BFGS two-loop recursion for $B_k^{L-BFGS} g_k$, Algorithm 6.

Algorithm 6 L-BFGS two-loop recursion for $B_k^{L-BFGS} g_k$

$q = g_k$
for $i = k-1, k-2, \cdots, k-m$ **do**
    $\rho_i = \frac{1}{s_i^{\mathrm{T}} y_i}$
    $\alpha_i = \rho_i s_i^{\mathrm{T}} q$
    $q = q - \alpha_i y_i$
end for
$r = B_k^{(0)} q$
for $i = k-m, k-m+1, \cdots, k-1$ **do**
    $\beta = \rho_i y_i^{\mathrm{T}} r$
    $r = r + s_i(\alpha_i - \beta)$
end for
stop with result $B_k^{L-BFGS} g_k = r$

Without considering the multiplication $B_k^{(0)} q$, the L-BFGS two-loop recursion requires $4mn$ multiplications. If $B_k^{(0)}$ is diagonal, then n additional multiplications are needed. Apart from being inexpensive, this recursion has the advantage that the multiplication by the initial matrix $B_k^{(0)}$ is isolated from the rest of the computations, allowing this matrix to be chosen freely and to vary between iterations. One may even use an implicit choice of $B_k^{(0)}$ by defining some initial approximation $H_k^{(0)}$ to the Hessian (not its inverse) and obtaining r by solving the system $H_k^{(0)} r = q$ Nocedal, Wright, 2006, p. 178.
$B_k^{(0)}$ can be an arbitrarily, symmetrical and positive definite matrix. In general $B_k^{(0)}$ will be a multiple of the identity matrix, so that it can be stored very easily Geiger, Kanzow, 1999, p. 198. A method for choosing $B_k^{(0)}$ that has proven effective in practice is to set $B_k^{(0)} = \gamma_k I$, where

$$\gamma_k = \frac{s_{k-1}^{\mathrm{T}} y_{k-1}}{y_{k-1}^{\mathrm{T}} y_{k-1}}. \tag{1.6.4}$$

γ_k is the scaling factor that attempts to estimate the size of the true Hessian matrix along the most recent search direction. This choice helps to ensure that the search direction d_k is well scaled, and as a result the stepsize $\alpha_k = 1$ is accepted in most iterations. It is important that the line search is based on the (strong) Wolfe conditions, so that the BFGS updating is stable Nocedal, Wright, 2006, p. 178-179.
With the previous theoery, the following algorithm can be created for the L-BFGS-method:

Unlike the conventional quasi-Newton methods, this one follows a different algorithmic sequence. Here, first the approximation of the Hessian inverse B_k^{L-BFGS} is calculated for the current iteration x_k and then the next iteration x_{k+1} is calculated. For example with the (globalized) BFGS-method, the new iteration x_{k+1} is calculated first and then the approximation of the Hessian inverse for the new iteration B_{k+1}^{BFGS} is calculated. This is the core idea of this method. Instead of passing the completely calculated matrix, only the vector pairs $\{s_i, y_i\}_{i=k-m}^{k-1}$ are passed in each iteration and at the beginning the approximation B_k^{L-BFGS} is created from these. The matrices B_k^{L-BFGS} are not explicitly stored, but only the vector pairs needed for the calculation and the start matrix $B_k^{(0)}$. For small values of m and larger dimensions n, the memory requirement for the L-BFGS-method is thus considerably lower than

Algorithm 7 L-BFGS-Method

$x_0 \in \mathbb{R}^n$, $B_0^{L-BFGS} \in \mathbb{R}^{n\times n}$ spd, $0 \leq \epsilon < 1$, $\sigma \in (0, \frac{1}{2})$, $\rho \in (\sigma, 1)$, $m \in \mathbb{N}$, set $k = 0$
while $\|\nabla f(x_k)\| > \epsilon$ **do**
    Choose $B_k^{(0)}$ (e.g. $B_k^{(0)} = \gamma_k I$ from Eq. (3.6.2)).
    Compute $d_k = -B_k^{L-BFGS}\nabla f(x_k)$ with $B_k^{L-BFGS}\nabla f(x_k)$ from Algorithm 6.
    Find a stepsize $\alpha_k = \alpha(\sigma, \rho)$ that satisfies the (strong) Wolfe conditions.
    Set $x_{k+1} = x_k + \alpha_k d_k$.
    if $k > m$ **then**
        Discard the vector pairs $\{s_{k-m}, y_{k-m}\}$ from storage.
    end if
    Compute and save $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.
    Set $k = k + 1$.
end while
return x_k

for the (globalized) BFGS-method itself, namely $O(mn)$ instead of $O(n^2)$, which is due to the fact that the stepsize d_k can be obtained with $O(mn)$ operations Geiger, Kanzow, 1999, p. 200-201.
In practical applications of the L-BFGS-method, the strong Powell-Wolfe stepsize strategy is used. This is because the L-BFGS-method seems to depend more on the choice of a "good" stepsize $\alpha_k > 0$ than, for example, the (globalized) BFGS-method, and because the "optimal" stepsize can be better approximated by means of the strong Powell-Wolfe rule Geiger, Kanzow, 1999, p. 212-213.
During its first $m - 1$ iterations, the L-BFGS-method (Algorithm 7) is equivalent to the inverse global BFGS-method (Algorithm 4) if the initial matrix is the same in both methods ($B_0^{L-BFGS} = B_0^{BFGS}$), and if L-BFGS chooses $B_k^{(0)} = B_0^{L-BFGS}$ at each iteration Nocedal, Wright, 2006, p. 179.
Before discussing the convergence properties, it must first be ensured that the L-BFGS-method (Algorithm 7) is well defined:

Theorem 1.6.1 (Geiger, Kanzow, 1999, Note 12.3). *If $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and bounded below, then $s_k^T y_k > 0$ holds for the sequences $\{s_k\}_k$ and $\{y_k\}_k$ generated by the L-BFGS-method (Algorithm 7). Furthermore, the matrices of the sequence $\{B_k^{L-BFGS}\}_k$ are symmetric and positive definite and the L-BFGS-method (Algorithm 7) is well defined.*

The following statement can be made about global convergence:

Theorem 1.6.2 (Sun, Yuan, 2006, Theorem 5.7.4). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable and uniformly convex function. Then the iterative sequence $\{x_k\}_k$ generated by the L-BFGS-method (Algorithm 7) converges to the unique minimizer x^* of f.*

Thus sequences $\{x_k\}_k$ generated by the L-BFGS-method, like by the inverse global BFGS-method, converge globally for twice continuously differentiable, uniformly convex functions. The following can be said about their rate of convergence:

Theorem 1.6.3 (Sun, Yuan, 2006, Theorem 5.7.7). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable*

and uniformly convex function. Assume that the iterative sequence $\{x_k\}_k$ generated by the L-BFGS-method (Algorithm 7) converges to the unique minimizer x^* of f. Then the rate of convergence is at least R-linear.

This theorem indicates that the L-BFGS-method often converges slowly, which leads to a relatively large number of function evaluations. Also, it is inefficient on highly ill-conditioned optimization problems. Though there are some weaknesses, L-BFGS-method is a main choice for large-scale problems in which the true Hessian is not sparse, because, in this case, it may outperform other rival algorithms Sun, Yuan, 2006.

The memoryless BFGS-method should also be mentioned. Here, $B_k^{BFGS} = I$ is inserted into the formula Eq. (1.6.2). It reads

$$B_{k+1}^{BFGS} = V_k^{\mathrm{T}} V_k + \rho_k s_k s_k^{\mathrm{T}}.$$

This formula satisfies the quasi-Newton equation Eq. (1.2.2), is positive definite and is called the memoryless BFGS formula. Obviously, if $m = 1$ and $B_k^{(0)} = I$ for all $k \in \mathbb{N}$, the limited-memory BFGS-method is just the memoryless BFGS-method. The idea is to use only the information from the previous iteration Sun, Yuan, 2006.

2 Riemannian Manifolds

- Riemannian Manifold
- Tangent Space with the defined opeartions
- Self-adjoint Operators on the tagent space
- Matrix Representation of Operators
- Gradient
- Hessian
- Retraction
- Exponential
- Logarithm
- Vector Transport
- Isometric Vector Transport
- Parallel Transport/Translation
- Vector transport by differentiated retraction
- Flat and sharp operations
- Tangent Vector Field
- Normal Neighborhood
- Levi-Civita connection

Definition 2.0.1 (Absil, Mahony, Sepulchre, 2008, Definition 4.1.1)**.** *A retraction on a manifold $\mathcal{M}$ is a smooth mapping R from the tangent bundle $\mathcal{TM}$ onto $\mathcal{M}$ with the following properties. Let R_x denote the restriction of R to $\mathcal{T}_x\mathcal{M}$.*

(i) $R_x(0_x) = x$, *where* 0_x *denotes the zero element of* $\mathcal{T}_x\mathcal{M}$.

(ii) *With the canonical identification* $\mathcal{T}_{0_x}\mathcal{T}_x\mathcal{M} \simeq \mathcal{T}_x\mathcal{M}$, R_x *satisfies*

$$\mathrm{D}R_x(0_x) = \mathrm{id}_{\mathcal{T}_x\mathcal{M}},$$

where $\mathrm{id}_{\mathcal{T}_x\mathcal{M}}$ *denotes the identity mapping on* $\mathcal{T}_x\mathcal{M}$.

Definition 2.0.2 (Absil, Mahony, Sepulchre, 2008, Definition 8.1.1)**.** *A vector transport on a manifold $\mathcal{M}$ is a smooth mapping*

$$T : \mathcal{TM} \oplus \mathcal{TM} \to \mathcal{TM}$$

$$(\eta_x, \xi_x) \mapsto T_{\eta_x}(\xi_x)$$

satisfying the following properties for all $x \in \mathcal{M}$*:*

(i) (Associated retraction) There exists a retraction R*, called the retraction associated with* T*, such that the following diagram commutes*

$$\begin{array}{ccc} (\eta_x, \xi_x) & \xrightarrow{T} & T_{\eta_x}(\xi_x) \\ \downarrow & & \downarrow \pi \\ \eta_x & \xrightarrow[R]{} & \pi(T_{\eta_x}(\xi_x)) \end{array}$$

where $\pi(T_{\eta_x}(\xi_x))$ *denotes the foot of the tangent vector* $T_{\eta_x}(\xi_x)$*.*

(ii) (Consistency) $T_{0_x}(\xi_x) = \xi_x$ *for all* $\xi_x \in \mathcal{T}_x\mathcal{M}$*;*

(iii) (Linearity) $T_{\eta_x}(a\xi_x + b\zeta_x) = aT_{\eta_x}(\xi_x) + bT_{\eta_x}(\zeta_x)$*.*

Definition 2.0.3 (Huang, 2013, p. 10)**.** *A vector transport* $T : \mathcal{TM} \oplus \mathcal{TM} \to \mathcal{TM}$ *with associated retraction* R *is called isometric if it satisfies for all* $x \in \mathcal{M}$

$$g_{R(\eta_x)}(T_{\eta_x}(\xi_x), T_{\zeta_x}(\xi_x)) = g_x(\eta_x, \zeta_x)$$

for all $(\eta_x, \xi_x) \in \mathcal{T}_x\mathcal{M} \oplus \mathcal{T}_x\mathcal{M}$*.*

Definition 2.0.4 (Absil, Mahony, Sepulchre, 2008, p. 172)**.** *A vector transport* $T : \mathcal{TM} \oplus \mathcal{TM} \to \mathcal{TM}$ *by differentiated retraction is a vector transport given by*

$$T_{\eta_x}(\xi_x) = \mathrm{D}R_x(\eta_x)[\xi_x]$$

i.e.,

$$T_{\eta_x}(\xi_x) = \frac{\mathrm{d}}{\mathrm{d}t} R_x(\eta_x + t\xi_x)|_{t=0}$$

where $R_x : \mathcal{T}_x\mathcal{M} \to \mathcal{M}$ *is a retraction.*

Definition 2.0.5 (Qi, 2011, Definition 2.2.1)**.** *Let $(\mathcal{M}, g)$ be a Riemannian manifold and let $X = X^i \partial_i$ be a vector field on $\mathcal{M}$, where $\{\partial_i\}$ is a local frame for the tangent bundle $\mathcal{TM}$. The flat of X is defined by $X^\flat = g_{ij} X^i dx^j = X_j dx^j$ where $\{dx^i\}$ is the dual coframe and the metric g is defined locally, using Einstein notation, as $g = g_{ij} dx^i \otimes dx^j$. Equivalently, we have $X^\flat(Y) = g(X, Y)$ for all vectors X and Y.*

Definition 2.0.6 (Absil, Mahony, Sepulchre, 2008, p. 192)**.** *Let X be a topological space. A neighborhood of a point $x \in X$ is a subset of X that includes an open set containing x.*

3 The BFGS-Method For Riemannian Manifolds

3.1 Preliminaries

- Riemannian Newton-Method
- (strong) retraction-convexity
- Table of differences to the Euclidean setting

Generalization of the Wolfe conditions to Riemannian manifolds:

$$f(R_{x_k}(\alpha_k \eta_k)) \leq f(x_k) + c_1 \alpha_k g_{x_k}(\operatorname{grad} f(x_k), \eta_k) \tag{3.1.1}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} f(R_{x_k}(t\eta_k))|_{t=\alpha_k} \geq c_2 \frac{\mathrm{d}}{\mathrm{d}t} f(R_{x_k}(t\eta_k))|_{t=0} \tag{3.1.2}$$

Other generalizations of the Wolfe conditions are possible. For example, if the vector transport T is an isometry, then (2.3) can be replaced by:

$$g_{x_k}({\mathrm{T}^{R_{x_k}}_{x_k,\alpha_k\eta_k}}^{-1}(\operatorname{grad} f(R_{x_k}(\alpha_k \eta_k))), \eta_k) \geq c_2 g_{x_k}(\operatorname{grad} f(x_k), \eta_k) \tag{3.1.3}$$

For parallel transport and the exponential map as the retraction, conditions Eq. (3.1.2) and Eq. (3.1.3) are identical Qi, 2011, p. 13.

Generalization of the strong Wolfe conditions to Riemannian manifolds:

$$f(R_{x_k}(\alpha_k \eta_k)) \leq f(x_k) + c_1 \alpha_k g_{x_k}(\operatorname{grad} f(x_k), \eta_k) \tag{3.1.4}$$

$$|g_{R_{x_k}(\alpha_k\eta_k)}(\operatorname{grad} f(R_{x_k}(\alpha_k \eta_k)), \mathrm{D}R_{x_k}(\alpha_k \eta_k)[\eta_k])| \leq c_2 |g_{x_k}(\operatorname{grad} f(x_k), \eta_k)| \tag{3.1.5}$$

where $0 < c_1 < c_2 < 1$, by Sato, Iwai, 2015.

Proposition 3.1.1 (Sato, Iwai, 2015, Proposition 2.1.). *Let $\mathcal{M}$ be a Riemannian manifold with a retraction R. If a smooth objective function f on $\mathcal{M}$ is bounded below on $\{R_{x_k}(\alpha\eta_k)|\alpha > 0\}$ for $x_k \in \mathcal{M}$ and for a descent direction $\eta_k \in \mathcal{T}_{x_k}\mathcal{M}$, and if constants c_1 and c_2 satisfy $0 < c_1 < c_2 < 1$, then there exists a stepsize α_k which satisfies the strong Wolfe conditions Eq. (3.1.4) and Eq. (3.1.5).*

Definition 3.1.2 (Huang, 2013, Definition 4.3.1). *For a function $f\colon \mathcal{M} \to \mathbb{R}\colon x \mapsto f(x)$ on a Riemannian manifold $\mathcal{M}$ with retraction R define $\tilde{m}_{x,\eta}(t) = f(R_x(t\eta))$ for $x \in \mathcal{M}$ and $\eta \in \mathcal{T}_x\mathcal{M}$. The function f is retractionconvex with respect to the retraction R in a set $\mathcal{S}$ if for all $x \in \mathcal{S}$, all $\eta \in \mathcal{T}_x\mathcal{M}$ and $\|\eta\| = 1$, $\tilde{m}_{x,\eta}(t)$ is convex for all t which satisfies $R_x(t\eta) \in \mathcal{S}$. Moreover, f is strongly retraction-convex in S if $\tilde{m}_{x,\eta}(t)$ is strongly convex for all $x \in \mathcal{S}$ and all $\|\eta\| = 1$ such that $R_x(\eta) \in \mathcal{S}$.*

Definition 3.1.3 (Cruz Neto, Melo, Sousa, 2017, p. 5). *A function $f\colon \mathcal{M} \to \mathbb{R}$ on a Riemannian manifold $\mathcal{M}$ is convex if its restriction to every geodesic in $\mathcal{M}$ is a convex function along the geodesic, i.e., if for every geodesic segment $\gamma\colon [a,b] \to \mathcal{M}$ and every $t \in [0,1]$,*

$$f(\gamma((1-t)a + tb)) \leq (1-t)f(\gamma(a)) + tf(\gamma(b)).$$

A convex function f is strictly convex if this inequality is strict whenever $t \in (0,1)$. A convex function is always continuous. If f is smooth, it is known that f is (strictly) convex provided its Hessian is positive (definite) semidefinite, or equivalently if $(f \circ \gamma)'' \geq 0$ (> 0) for every geodesic $\gamma\colon I \subset \mathbb{R} \to \mathcal{M}$.

Definition 3.1.4. *Let H be a Hilbert space with inner product $\langle\cdot,\cdot\rangle$ and $\mathcal{L}(\mathrm{H})$ the set of linear and continuous operators on H. An operator $A \in \mathcal{L}(\mathrm{H})$ is called positive definite if*

$$\langle Ax, x\rangle \geq 0$$

holds for all $x \in \mathrm{H}$.

Definition 3.1.5. *Let H be a Hilbert space with inner product $\langle\cdot,\cdot\rangle$ and $\mathcal{L}(\mathrm{H})$ the set of linear and continuous operators on H. An operator $A \in \mathcal{L}(\mathrm{H})$ is called self-adjoint if $A = A^*$ holds, i.e.*

$$\langle Ax, y\rangle = \langle x, Ay\rangle$$

holds for all $x, y \in \mathrm{H}$.

Definition 3.1.6. *An iterative update scheme of an algorithm on a Riemannian manifold $\mathcal{M}$ is defined as: Starting with $x_0 \in \mathcal{M}$ (an initial guess) the algorithm computes*

$$x_{k+1} = R_{x_k}(\alpha_k\eta_k), \quad k = 1, 2, \cdots,$$

where $\eta_k \in \mathcal{T}_{x_k}\mathcal{M}$ is a tangent vector and $\alpha_k > 0$ is a stepsize, which are determined in each iteration and $R_{x_k} : \mathcal{T}_{x_k}\mathcal{M} \to \mathcal{M}$ a retraction, depending on the iterate $x_k \in \mathcal{M}$.

Definition 3.1.7 (Zhang, Sra, 2016, Definition 2). *A differentiable function $f : \mathcal{M} \to \mathbb{R}$ is said to be geodesically μ-strongly convex if for any $x, y \in \mathcal{M}$,*

$$f(y) \geq f(x) + g_x(\nabla f(x), \log_x y) + \frac{\mu}{2} d(x, y)^2$$

or, equivalently, for any geodesic γ such that $\gamma(0) = x$, $\gamma(1) = y$ and $t \in [0, 1]$,

$$f(\gamma(t)) \leq (1 - t)f(x) + tf(y) - \frac{\mu}{2} t(1 - t)d(x, y)^2.$$

Theorem 3.1.8 (Deng, 2011, Theorem 1.1). *Let* H *and* K *be Hilbert spaces over the same field. Let $A \in \mathcal{L}(\mathrm{H})$ and $G \in \mathcal{L}(\mathrm{K})$ both be invertible, and $Y, Z \in \mathcal{L}(\mathrm{K}, \mathrm{H})$. Then $A + YGZ^*$ is invertible iff $G^{-1} + Z^*A^{-1}Y$ is invertible. In which case,*

$$(A + YGZ^*)^{-1} = A^{-1} - A^{-1}Y(G^{-1} + Z^*A^{-1}Y)^{-1}Z^*A^{-1}. \tag{3.1.6}$$

Theorem 3.1.9 (Qi, 2011, Theorem 2.4.1). *Consider any iteration of form $x_{k+1} = \exp_{x_k} \alpha_k \eta_k$, where η_k is a descent direction and α_k satisfies the Wolfe conditions Eq. (3.1.1) and Eq. (3.1.2). Suppose that f is bounded below on $\mathcal{M}$ and that f is continuously differentiable in an open set $\mathcal{N}$ containing the level set $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$, where x_0 is the starting point of the iteration. Assume also that the gradient* grad f *is Lipschitz continuous on $\mathcal{N}$, then*

$$\sum_{k \geq 0} \cos(\theta_k)^2 \|\mathrm{grad}\, f(x_k)\|^2 < \infty,$$

where

$$\cos(\theta_k) = -\frac{g_{x_k}(\mathrm{grad}\, f(x_k), \eta_k)}{\|\mathrm{grad}\, f(x_k)\| \|\eta_k\|}.$$

Theorem 3.1.10 (Qi, 2011, Theorem 2.3.1). *Let $\mathcal{M}$ be a manifold endowed with a C^2 vector transport $\mathrm{T}^{\mathrm{retr}}_{\cdot \leftarrow \cdot}$ and an associated retraction* retr. *Let F be a C^2 tangent vector field on $\mathcal{M}$. Also let $\mathcal{M}$ be endowed with an affine connection ∇. Let $\mathbb{D}F(x)$ denote the linear transformation of $\mathcal{T}_x\mathcal{M}$ defined by $\mathbb{D}F(x)[\xi_x] = \nabla_{\xi_x} F$ for all tangent vectors ξ_x to $\mathcal{M}$ at x. Let $\{\mathcal{B}_k\}_k$ be a sequence of bounded nonsingular linear transformations of*

$\mathcal{T}_{x_k}\mathcal{M}$, *where* $k = 0, 1, \cdots$, $x_{k+1} = \mathrm{retr}_{x_k}\eta_k$, *and* $\eta_k = -\mathcal{B}_k^{-1}[F(x_k)]$. *Assume that* $\mathbb{D}F(x^*)$ *is nonsingular,* $x_k \neq x^*$, $\forall k$, *and* $\lim_{k\to\infty} x_k = x^*$. *Then* $\{x_k\}_k$ *converges superlinearly to* x^* *and* $F(x^*) = 0$ *if and only if*

$$\lim_{k\to\infty} \frac{\|\mathcal{B}_k[\eta_k] - \mathrm{T}^{\mathrm{retr}}_{\xi_k \leftarrow x^*} \circ \mathbb{D}F(x^*) \circ (\mathrm{T}^{\mathrm{retr}}_{\xi_k \leftarrow x^*})^{-1}[\eta_k]\|}{\|\eta_k\|} = 0,$$

where $\xi_k \in \mathcal{T}_{x^*}\mathcal{M}$ *is defined by* $\xi_k = \mathrm{retr}^{-1}_{x^*} x_k$, *i.e.* $\mathrm{retr}_{x^*}\xi_k = x_k$.

3.2 Quasi-Newton Methods For Riemannian Manifolds

This subsection creates a foundation for quasi-Newton methods on Riemannian manifolds. We try to name the most important points of the general theory, so that the derivation of the BFGS method on this structure is reasonable and makes sense. Quasi-Newton methods on Riemannian manifolds are often obtained by generalizing their Euclidean counterparts. This will also happen in some aspects in this paper but the purpose is that we want to start ... and build the theory
In subsection 1.2 the following properties have proved to be important for the approximating matrices H_k and B_k in the Euclidean case: positive definiteness, symmetry and satisfying the quasi-Newton equation. We will now investigate the Riemannian analogue. Let us begin with the quasi-Newton equation, the core of the theory. The equation reads for the Euclidean case:

$$H_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k) \quad \text{or} \quad H_{k+1}s_k = y_k.$$

This cannot be transferred one-to-one to manifolds. For the generalization to the Riemannian metric one idea would be to use $\mathrm{retr}^{-1}_{\cdot}\cdot$ as a minus operator, since it returns a tangent vector which defines a curve connecting the two points x_k and x_{k+1}, and to compare the vectors, one could map $\mathrm{grad}\, f(x_k)$ to the tangent space of x_{k+1} using a vector transport. And since we now have an equation between tangent vectors, the matrix H_{k+1} becomes a linear operator $\mathcal{H}_{k+1}$ on the tangent space $\mathcal{T}_{x_{k+1}}\mathcal{M}$. This leads to a naive quasi-Newton equation:

$$\mathcal{H}_{k+1}[\mathrm{T}^{\mathrm{retr}}_{x_k,\eta_k}(\eta_k)] = \mathrm{grad}\, f(x_{k+1}) - \mathrm{T}^{\mathrm{retr}}_{x_k,\eta_k}(\mathrm{grad}\, f(x_k)) \tag{3.2.1}$$

where $\eta_k \in \mathcal{T}_{x_k}\mathcal{M}$ is the update vector at the iterate x_k, i.e. $\mathrm{retr}_{x_k}\eta_k = x_{k+1}$ Absil, Mahony, Sepulchre, 2008, p. 179.
But here the first problems arise, namely the determination of the vector transport $\mathrm{T}_{\cdot,\cdot}(\cdot)$ and the retraction $\mathrm{retr}_{\cdot}\cdot$. In Chapter 2 the two concepts were presented and kept as general as possible. So there is no unique vector transport $\mathrm{T}_{\cdot,\cdot}(\cdot)$ and no unique retraction $\mathrm{retr}_{\cdot}\cdot$ in general. The quasi-Newton equation and the resulting quasi-Newton method depend crucially on the choice of these two maps. The above equation is not wrong a priori but to get a well-defined method and clear results regarding its convergence, it is useful to take a closer look at these functions.
We now consider the most natural generalization of the Euclidean quasi-Newton equation Eq. (1.2.2). It

lends itself to parallel transport $\mathrm{P}_{\cdot\leftarrow\cdot}(\cdot)$ and the exponential map $\exp_{\cdot}\cdot$ or its inverse, the logarithmic map $\log_{\cdot}\cdot$. In subsection 1.2 Taylor's theorem was used, among other things, to deduce the secant equation. Similarly, there is a version of Taylor's Theorem for a vector field on a manifold Huang, 2013. The following Lemma is a first-order Taylor formula for tangent vector fields.

Lemma 3.2.1 (Absil, Mahony, Sepulchre, 2008, Lemma 7.4.7)**.** *Let $x \in \mathcal{M}$, let $\mathcal{V}$ be a normal neighborhood of x, and let ζ be a C^1 tangent vector field on $\mathcal{M}$. Then, for all $y \in \mathcal{V}$,*

$$\mathrm{P}_{x\leftarrow y}(\zeta_y) = \zeta_x + \nabla_\xi\zeta + \int_0^1 (\mathrm{P}_{x\leftarrow\gamma(\tau)}(\nabla_{\gamma'(\tau)}\zeta) - \nabla_\xi\zeta)\mathrm{d}\tau, \tag{3.2.2}$$

where γ is the unique minimizing geodesic satisfying $\gamma(0) = x$ and $\gamma(1) = y$, and $\xi = \log_x y = \gamma'(0)$.

Applying Taylor's Theorem on the gradient of f at x_{k+1}, one obtains

$$\mathrm{P}_{x_k\leftarrow x_{k+1}}(\operatorname{grad} f(x_{k+1})) = \operatorname{grad} f(x_k) + \nabla_\xi \operatorname{grad} f(x_k) + \int_0^1 (\mathrm{P}_{x_k\leftarrow\gamma(\tau)}(\nabla_{\gamma'(\tau)} \operatorname{grad} f(x_k)) - \nabla_\xi \operatorname{grad} f(x_k))\mathrm{d}\tau,$$

where γ_k is the unique minimizing geodesic satisfying $\gamma_k(0) = x_k$ and $\gamma_k(1) = x_{k+1}$, and $\xi = \log_{x_k} x_{k+1} = \gamma_k'(0)$.
Ignoring the integral remainder term and rearranging yields

$$\mathrm{P}_{x_k\leftarrow x_{k+1}}(\operatorname{grad} f(x_{k+1})) - \operatorname{grad} f(x_k) \approx \nabla_\xi \operatorname{grad} f(x_k) = \operatorname{Hess} f(x_k)[\log_{x_k} x_{k+1}]$$

This formula is very similar to the Euclidean quasi-Newton equation. It is defined on $\mathcal{T}_{x_k}\mathcal{M}$, the desired approximation $\mathcal{H}_{k+1}$ of the Hessian $\operatorname{Hess} f(x_{k+1})$ must be an operator on $\mathcal{T}_{x_{k+1}}\mathcal{M}$. Applying parallel transport, yields

$$\operatorname{grad} f(x_{k+1}) - \mathrm{P}_{x_{k+1}\leftarrow x_k}(\operatorname{grad} f(x_k)) = \mathcal{H}_{k+1}[\mathrm{P}_{x_{k+1}\leftarrow x_k}(\log_{x_k} x_{k+1})] \tag{3.2.3}$$

This is one Riemannian version of the quasi-Newton equation Eq. (1.2.2). There are several possible generalizations of the Euclidean secant condition to a Riemannian manifold Huang, 2013, p. 17. In the following, the term Riemannian quasi-Newton equation is used to refer to Eq. (3.2.3). We introduce $s_k = \mathrm{P}_{x_{k+1}\leftarrow x_k}(\log_{x_k} x_{k+1})$ and $y_k = \operatorname{grad} f(x_{k+1}) - \mathrm{P}_{x_{k+1}\leftarrow x_k}(\operatorname{grad} f(x_k))$ to shorten Eq. (3.2.3) - as in the Euclidean case - further to

$$y_k = \mathcal{H}_{k+1}[s_k] \quad \text{or equivalently} \quad \mathcal{B}_{k+1}[y_k] = s_k,$$

where $\mathcal{B}_{k+1} = \mathcal{H}_{k+1}^{-1}$.
From the definition of s_k we can conclude that we use exponential map $\exp_{\cdot}\cdot$ as retraction $\operatorname{retr}_{\cdot}\cdot$ in this variant, since $\log_{\cdot}\cdot = \exp_{\cdot}\cdot^{-1}$. This means the iterative-scheme has the form

$$x_{k+1} = \exp_{x_k} \alpha_k \eta_k,$$

where $\alpha_k > 0$ is a stepsize to be determined and $\eta_k \in \mathcal{T}_{x_k}\mathcal{M}$ is an update direction to be determined. The equation Eq. (3.2.3) explicitly uses the exponential map and parallel transport, due to its origins in Taylor's Theorem, but there's no need for that. As already mentioned, the Riemannian quasi-Newton method depends on the choice of these two maps. Alternate forms of this equation can be derived by using a different retraction or a different vector transport Huang, 2013. If these change, the work will draw attention to this and discuss the resulting effects on the method. For now we work with the equation Eq. (3.2.3), i.e. we get the next iteration by using the exponential map $\exp_{\cdot}\cdot$ as the retraction $\operatorname{retr}_{\cdot}\cdot$ and use the parallel transport $\mathrm{P}_{\cdot\leftarrow\cdot}(\cdot)$ as vector transport $\mathrm{T}_{\cdot,\cdot}(\cdot)$ to compare tangent vectors.
Now we come to the calculation of the update direction η_k. From the equation Eq. (3.2.3) it can be concluded that the operator $\mathcal{H}_k$ approximates the Hessian operator $\operatorname{Hess} f(x_k)$ of f at x_k. We consider the pullback $\hat{f}_{x_k} = f \circ \exp_{x_k} : \mathcal{T}_{x_k}\mathcal{M} \to \mathbb{R}$ at the current iterate x_k, because we want to build a quadratic model of f around x_k. Let m_k be the order-2 Taylor expansion of $\hat{f}_{x_k}$ around the origin 0_{x_k} of $\mathcal{T}_{x_k}\mathcal{M}$, i.e.

$$\begin{aligned}
\hat{f}_{x_k}(\eta) \approx m_k(\eta) &= \hat{f}_{x_k}(0_{x_k}) + \mathrm{D}\hat{f}_{x_k}(0_{x_k})[\eta] + \frac{1}{2}\mathrm{D}^2\hat{f}_{x_k}(0_{x_k})[\eta,\eta] \\
&= f(x_k) + g_{x_k}(\eta, \operatorname{grad}\hat{f}_{x_k}(0_{x_k})) + \frac{1}{2}g_{x_k}(\eta, \operatorname{Hess}\hat{f}_{x_k}(0_{x_k})[\eta]) \\
&= f(x_k) + g_{x_k}(\eta, \operatorname{grad} f(x_k)) + \frac{1}{2}g_{x_k}(\eta, \operatorname{Hess} f(x_k)[\eta])
\end{aligned}$$

since $\mathrm{D}\operatorname{retr}_{x_k} 0_{x_k} = \mathrm{id}_{\mathcal{T}_{x_k}\mathcal{M}}$ (see Definition 2.0.1), it follows that $\mathrm{D}\hat{f}_{x_k}(0_{x_k}) = \mathrm{D}f(x_k)$, hence $\operatorname{grad}\hat{f}_{x_k}(0_{x_k}) = \operatorname{grad} f(x_k)$. Since the exponential map $\exp_{\cdot}\cdot$ is a second-order retraction, $\operatorname{Hess}\hat{f}_{x_k}(0_{x_k}) = \operatorname{Hess} f(x_k)$ holds and m_k is in fact order 2 Absil, Mahony, Sepulchre, 2008, p. 139
Since the Hessian of f at the iterate x_k should be approximated by the operator $\mathcal{H}_k$, we get the following model

$$m_k(\eta) = f(x_k) + g_{x_k}(\eta, \operatorname{grad} f(x_k)) + \frac{1}{2}g_{x_k}(\eta, \mathcal{H}_k[\eta]).$$

If we take the derivative $\frac{\mathrm{d}}{\mathrm{d}\eta}$ of the model, set it to zero and solve the equation, we get

$$\eta_k = -\mathcal{H}_k^{-1}[\operatorname{grad} f(x_k)] = -\mathcal{B}_k[\operatorname{grad} f(x_k)] \in \mathcal{T}_{x_k}\mathcal{M}.$$

It remains to be clarified whether this is a descent direction and thus can be used as an update vector. Before discussing this, we consider a property which is required for the approximation in the Euclidean case. In this case we demand that H_k or B_k are symmetric for all k. Since the Hessian matrix $\nabla^2 f(x_k)$ is for twice continuously differentiable functions always symmetric and since H_k or B_k are approximating it, it makes sense that this should be required. The Riemannian Hessian has a similar characteristic:

Proposition 3.2.2 (Absil, Mahony, Sepulchre, 2008, Proposition 5.5.3)**.** *The Riemannian Hessian is symmetric (in the sense of the Riemannian metric). That is,*

$$g(\operatorname{Hess} f[\eta], \xi) = g(\eta, \operatorname{Hess} f[\xi])$$

for all $\eta, \xi \in \mathfrak{X}(\mathrm{M})$.

We see immediately that symmetry can be generalized almost one to one for tangent spaces. But the term "symmetrical" could be misleading in some places. Since the Hessian can be seen as an operator on a tangent space, we call it self-adjoint instead. It follows: On a Riemannian manifold the Hessian $\operatorname{Hess} f(\cdot)$ of a function f is always a self-adjoint operator. The natural consequence is that it is required that the approximations $\mathcal{H}_k$ and $\mathcal{B}_k$ should also be self-adjoint Huang, 2013, p. 20.
Finally, it remains to be clarified whether the η_k, as determined above, is really a descent direction. In the Euclidean case, a descent direction is defined by the fact that the inner product of the vector indicating this direction and the gradient is negative. Consequently it is required that for quasi-Newton methods the approximations H_k and B_k are positive definite in each iteration to ensure a continuous descent. This idea can also be adopted here. In the Riemannian case a descent direction of a function $f\colon \mathcal{M} \to \mathbb{R}$ at a point $x \in \mathcal{M}$ denotes a tangent vector $\eta \in \mathcal{T}_x\mathcal{M}$ with $g_x(\operatorname{grad} f(x), \eta) < 0$. This property ensures that the objective function f indeed decreases along the search direction Ring, Wirth, 2012, p. 5. In our case this means

$$g_{x_k}(\operatorname{grad} f(x_k), -\alpha_k \mathcal{B}_k[\operatorname{grad} f(x_k)]) < 0.$$

Using the linearity of g_{x_k} and since $\alpha_k > 0$, this implies that $g_{x_k}(\operatorname{grad} f(x_k), \mathcal{B}_k[\operatorname{grad} f(x_k)]) = g_{x_k}(\mathcal{B}_k[\operatorname{grad} f(x_k)], \operatorname{grad} f(x_k)) > 0$ must hold in every iteration. So it makes sense that we require that the $\mathcal{B}_k$ (and thus of course also $\mathcal{H}_k$) is a positive definite operator in every iteration, so that the η_k is a descent direction. The goal is to find updates hence all operators in the sequence $\{\mathcal{B}_k\}_k$ are positive definite and self-adjoint to create a continuous descent.
The positive definiteness of the approximating operators $\mathcal{H}_k$ and $\mathcal{B}_k$ has a consequence that is similar to one we have already seen in the Euclidean case: the curvature condition Eq. (1.2.6). Much as in the Euclidean case, it is essential that

$$g_{x_{k+1}}(s_k, y_k) > 0 \tag{3.2.4}$$

holds, otherwise the secant condition $\mathcal{H}_{k+1}[s_k] = y_k$ cannot hold with $\mathcal{H}_{k+1}$ positive definite, whereas

positive definiteness of the operators is the key to guarantee that the search directions η_k are descent directions Huang, 2013, p. 54. In the Euclidean setting, the inequality Eq. (1.2.6) holds for any two points x_k and x_{k+1}, if the objective f is strongly convex. This can also be adopted almost one-to-one. But we need a different characterization of geodesically μ-strongly convex functions on Riemannian manifolds.

Theorem 3.2.3. *A differentiable function $f\colon \mathcal{M} \to \mathbb{R}$ is geodesically μ-strongly convex if and only if*

$$g_x(\mathrm{P}_{y\leftarrow x}(\operatorname{grad} f(y)) - \operatorname{grad} f(x), \log_x y) \geq \mu d(x,y)^2$$

holds for any $x, y \in \mathcal{M}$.

Proof. From Definition 3.1.7 we have

$$g_x(\operatorname{grad} f(x), \log_x y) \leq f(y) - f(x) - \frac{\mu}{2} d(x,y)^2 \qquad \text{A}$$
$$g_y(\operatorname{grad} f(y), \log_y x) \leq f(x) - f(y) - \frac{\mu}{2} d(x,y)^2. \qquad \text{B}$$

Now we draw -1 twice from the inner product $g_y(\cdot,\cdot)$ of the left side of inequality B, i.e.

$$g_y(\operatorname{grad} f(y), \log_y x) = (-1)\cdot g_y(\operatorname{grad} f(y), -\log_y x) = g_y(-\operatorname{grad} f(y), -\log_y x).$$

Then we use, that $-\log_y x = \mathrm{P}_{y\leftarrow x}(\log_x y)$ and apply the parallel transport $\mathrm{P}_{x\leftarrow y}(\cdot)$ on both arguments

$$g_y(-\operatorname{grad} f(y), -\log_y x) = g_x(-\mathrm{P}_{x\leftarrow y}(\operatorname{grad} f(y)), \log_x y).$$

That means

$$g_x(-\mathrm{P}_{x\leftarrow y}(\operatorname{grad} f(y)), \log_x y) \leq f(x) - f(y) - \frac{\mu}{2} d(x,y)^2. \qquad \text{B}$$

holds. Now adding the inequalities A and B leads to

$$g_x(\operatorname{grad} f(x) - \mathrm{P}_{x\leftarrow y}(\operatorname{grad} f(y)), \log_x y) \leq -\mu d(x,y)^2$$

Multiplying both sides with -1 gives us the inequality. □

If y is set equal to x_{k+1} and x equal to x_k, only the parallel transport $\mathrm{P}_{x_{k+1}\leftarrow x_k}(\cdot)$ has to be applied to both arguments and we see that the Riemannian curvature condition Eq. (3.2.4) holds for geodesically μ-strongly convex functions for all points $x_k, x_{k+1} \in \mathcal{M}$. If the function is not geodesically μ-strongly convex, it cannot be guaranteed that the condition always holds. This is where determining the stepsize comes into play. Assuming that the stepsize α_k meets the Wolfe conditions Eq. (3.1.3) or strong Wolfe conditions Eq. (3.1.5), we get

$$\begin{aligned} g_{x_{k+1}}(s_k, y_k) &= g_{x_{k+1}}(\mathrm{P}_{x_{k+1}\leftarrow x_k}(\alpha_k\eta_k), \operatorname{grad} f(x_{k+1}) - \mathrm{P}_{x_{k+1}\leftarrow x_k}(\operatorname{grad} f(x_k))) \\ &= g_{x_{k+1}}(\mathrm{P}_{x_{k+1}\leftarrow x_k}(\alpha_k\eta_k), \operatorname{grad} f(x_{k+1})) - g_{x_k}(\alpha_k\eta_k, \operatorname{grad} f(x_k)) \\ &= \alpha_k g_{x_{k+1}}(\mathrm{P}_{x_{k+1}\leftarrow x_k}(\eta_k), \operatorname{grad} f(x_{k+1})) - \alpha_k g_{x_k}(\eta_k, \operatorname{grad} f(x_k)) \\ &= \alpha_k g_{x_k}(\mathrm{P}_{x_k\leftarrow x_{k+1}}(\operatorname{grad} f(x_{k+1})), \eta_k) - \alpha_k g_{x_k}(\eta_k, \operatorname{grad} f(x_k)) \\ &\geq \alpha_k c_2 g_{x_k}(\eta_k, \operatorname{grad} f(x_k)) - \alpha_k g_{x_k}(\eta_k, \operatorname{grad} f(x_k)) \\ &= (c_2 - 1)\alpha_k g_{x_k}(\eta_k, \operatorname{grad} f(x_k)) \end{aligned} \tag{3.2.5}$$

Since $c_2 < 1$ and η_k is a descent direction, the right side is positive and the curvature condition holds. Using the strong Wolfe conditions Eq. (3.1.5), the inequality still remains correct.
At the end of this subsection a very general quasi-Newton method will be presented. We will now dispense with an exact definition of retraction and vector transport in order to preserve the generality. Depending on the choice of a certain combination of these two and the update formula for the approximation, this leads to different algorithms and thus to different results with respect to convergence.

Algorithm 8 General Riemannian Quasi-Newton Method

1: Riemannian manifold $\mathcal{M}$ with Riemannian metric g, vector transport T on $\mathcal{M}$ with associated retraction retr, smooth real-valued function f on $\mathcal{M}$, initial iterate $x_0 \in \mathcal{M}$, initial Hessian approximation $\mathcal{H}_0$.
2: **while** not converged **do**
3: Obtain $\eta_k \in \mathcal{T}_{x_k}\mathcal{M}$ by solving $\mathcal{H}_k[\eta_k] = -\operatorname{grad} f(x_k)$.
4: Determine the stepsize $\alpha_k > 0$ by line search.
5: Set $x_{k+1} = \operatorname{retr}_{x_k}(\alpha_k\eta_k)$.
6: Define $s_k = \mathrm{T}_{x_k,\alpha_k\eta_k}(\alpha_k\eta_k)$ and $y_k = \operatorname{grad} f(x_{k+1}) - \mathrm{T}_{x_k,\alpha_k\eta_k}(\operatorname{grad} f(x_k))$.
7: Define the linear operator $\mathcal{H}_{k+1}\colon \mathcal{T}_{x_{k+1}}\mathcal{M} \to \mathcal{T}_{x_{k+1}}\mathcal{M}$ by using s_k, y_k and $\widetilde{\mathcal{H}}_k = \mathrm{T}_{x_k,\alpha_k\eta_k} \circ \mathcal{H}_k \circ (\mathrm{T}_{x_k,\alpha_k\eta_k})^{-1}$, such that
$$\mathcal{H}_{k+1}[s_k] = y_k$$
holds.
8: Set $k = k + 1$.
9: **end while**
10: **return** x_k

3.3 The BFGS Formula For Riemannian Manifolds

Crucial for the representation of the formula is the decision whether we consider the Riemannian quasi-Newton equation Eq. (3.2.3) in $\mathcal{T}_{x_{k+1}}\mathcal{M}$ or in $\mathcal{T}^*_{x_{k+1}}\mathcal{M}$. The connection of both types of formulas can also be derived in the Euclidean. There the quasi-Newton equation has the form $H_{k+1}s_k = y_k$. If you now transpose both sides, $s_k^{\mathrm{T}}H_{k+1} = y_k^{\mathrm{T}}$, the vectors s_k^{T}, y_k^{T} are defined in other vector space, in their cotangent space $\mathbb{R}^{n*} = \mathbb{R}^{1\times n}$. If you derive an update formula from this equation, it will look different, but the matrix H_{k+1} will be the same in every iteration, since it is symmetric.

Something similar can be done in the Riemannian case. The transposition is replaced by the musically isomorphism, which assigns each tangent vector its corresponding cotangent vector. Given $\eta_x \in \mathcal{T}_x\mathcal{M}$, $\eta_x^\flat$ represents the flat of η_x, i.e., $\eta_x^\flat \colon \mathcal{T}_x\mathcal{M} \to \mathbb{R},\ \xi_x \mapsto \eta_x^\flat[\xi_x] = g_x(\eta_x, \xi_x)$. If we apply the isomorphism to both sides of the equation Eq. (3.2.3), we get

$$y_k^\flat[\cdot] = (\mathcal{H}_{k+1}s_k)^\flat[\cdot] \Leftrightarrow y_k^\flat[\cdot] = s_k^\flat[\mathcal{H}^*_{k+1}[\cdot]]$$

where $\mathcal{H}^*_{k+1}$ denotes the adjoint operator of $\mathcal{H}_{k+1}[\cdot]$, i.e., $\mathcal{H}^*_{k+1}$ satisfies $g_{x_{k+1}}(\mathcal{H}_{k+1}[\eta_{x_{k+1}}], \xi_{x_{k+1}}) = g_{x_{k+1}}(\eta_{x_{k+1}}, \mathcal{H}^*_{k+1}[\xi_{x_{k+1}}])$ for all $\eta_{x_{k+1}}, \xi_{x_{k+1}} \in \mathcal{T}_{x_{k+1}}\mathcal{M}$. But since $\mathcal{H}_{k+1}[\cdot]$ is always a self-adjoint operator, $\mathcal{H}_{k+1} = \mathcal{H}^*_{k+1}$ holds. In summary, there are two forms of the Riemannian quasi-Newton equation Eq. (3.2.3), but they are defined in different spaces

$$y_k^\flat[\cdot] = s_k^\flat[\mathcal{H}_{k+1}[\cdot]] \quad \text{or} \quad y_k = \mathcal{H}_{k+1}[s_k].$$

Of course these lead to different presentations of the update formula. We will discuss both variants and start with the one defined on $\mathcal{T}_{x_{k+1}}\mathcal{M}$.

Since convergence depends crucially on the definitions of s_k and y_k, and since this will be discussed later for various cases, we assume for the derivation of a general formula in this chapter only that they are tangent vectors in $\mathcal{T}_{x_{k+1}}\mathcal{M}$ which satisfy the curvature condition in every iteration, i.e. we just use some retraction retr and some vector transport T as in Eq. (3.2.1) and it holds $g_{x_{k+1}}(s_k, y_k) > 0$ for all $k = 0, 1, 2, \cdots$. Most of the preparatory work for the Riemannian BFGS formula, short RBFGS formula, was already done in the Euclidean case. An actual derivation of it has not yet been investigated in the Riemannian case. The approach here is a generalization of the Euclidean case. It is only to show that the derivation follows the same pattern. First the Riemannian version of a DFP update is created and then the variables are swapped. As in the Euclidean case we start with a rank 2 update for the operator $\mathcal{B}_k$, that means we want to make an operator $\mathcal{B}_{k+1}$ on $\mathcal{T}_{x_{k+1}}\mathcal{M}$, which approximates the Hessian inverse of a function $f \colon \mathcal{M} \to \mathbb{R}$ at x_{k+1}, by the addition of two simple operators of rank 1 multiplied by a scalar to $\mathcal{B}_k$.

$$\mathcal{B}_{k+1}[\cdot] = \widetilde{\mathcal{B}}_k[\cdot] + auu^\flat[\cdot] + bvv^\flat[\cdot]$$

where $\widetilde{\mathcal{B}}_k[\cdot] = \mathrm{T}_{x_k,\alpha_k\eta_k} \circ \mathcal{B}_k \circ (\mathrm{T}_{x_k,\alpha_k\eta_k})^{-1}[\cdot] = \mathrm{T}_{x_k,\alpha_k\eta_k}(\mathcal{B}_k[(\mathrm{T}_{x_k,\alpha_k\eta_k})^{-1}(\cdot)])$, $u, v \in \mathcal{T}_{x_{k+1}}\mathcal{M}$ and $a, b \in \mathbb{R}$ are to be determined. Here we see two interesting aspects: First, that the vector transport seems to be needed not only for comparing two vectors in different tangent spaces, but also for transporting operators from one tangent space to the appropriate one Huang, 2013, p. 20. And secondly, that the dyadic product needed to create rank 1 matrices can be transferred by a generalization with the musical isomorphism to create rank 1 operators in the tangent space $\mathcal{T}_{x_{k+1}}\mathcal{M}$ (one sees immediately that $uu^\flat[\cdot]$ and $vv^\flat[\cdot]$ are self-adjoint and positive definite operators from $\mathcal{T}_{x_{k+1}}\mathcal{M}$ to $\mathcal{T}_{x_{k+1}}\mathcal{M}$ with rank 1). From the quasi-Newton equation follows

$$\mathcal{B}_{k+1}[y_k] = \widetilde{\mathcal{B}}_k[y_k] + auu^\flat[y_k] + bvv^\flat[y_k] = s_k$$

Clearly, u and v can not uniquely be determined. One possible choice is

$$u = s_k, \; v = \widetilde{\mathcal{B}}_k[y_k].$$

Hence we obtain

$$a = \frac{1}{u^\flat[y_k]} = \frac{1}{s_k^\flat[y_k]}, \; b = -\frac{1}{v^\flat[y_k]} = \frac{1}{y_k^\flat(\widetilde{\mathcal{B}}_k^*[y_k])}.$$

At this point, a property must now be added to the vector transport T. We require that $\mathcal{B}_{k+1}$ is a self-adjoint operator. Therefore it makes sense to require that $\widetilde{\mathcal{B}}_k$ is also a self-adjoint operator, so that this property is preserved by the update. If we look at the definition of the operator $\widetilde{\mathcal{B}}_k[\cdot] = \mathrm{T}_{x_k,\alpha_k\eta_k} \circ \mathcal{B}_k \circ (\mathrm{T}_{x_k,\alpha_k\eta_k})^{-1}[\cdot]$, we see immediately that we must demand that the vector transport $\mathrm{T}_{x_k,\cdot}$ is isometric for all $k \in \mathbb{N}_0$ Huang, 2013, p. 20. Therefore $(\widetilde{\mathcal{B}}_k[y_k])^\flat[\cdot] = y_k^\flat[\widetilde{\mathcal{B}}_k[\cdot]]$ and $y_k^\flat(\widetilde{\mathcal{B}}_k^*[y_k]) = y_k^\flat(\widetilde{\mathcal{B}}_k[y_k])$ hold. With an isometric vector transport and setting $\mathcal{B}_k = \mathcal{B}_k^{RDFP}$ for all $k \in \mathbb{N}_0$ we get

$$\mathcal{B}_{k+1}^{RDFP}[\cdot] = \widetilde{\mathcal{B}}_k^{RDFP}[\cdot] + s_k \frac{s_k^\flat[\cdot]}{s_k^\flat[y_k]} - \widetilde{\mathcal{B}}_k^{RDFP}[y_k] \frac{y_k^\flat(\widetilde{\mathcal{B}}_k^{RDFP}[\cdot])}{y_k^\flat(\widetilde{\mathcal{B}}_k^{RDFP}[y_k])}$$

Riemannian DFP update formula to approximate $(\mathrm{Hess}\, f(x_{k+1}))^{-1}[\cdot]$. Since the DFP update is known for the approximation of Hessian (see e.g. Huang, 2013, p. 19), it is left to the reader as an exercise to show that this formula is correct. Again, the idea of quasi-Newton methods is very clear. Instead of calculating a complete approximation of the Hessian $\mathrm{Hess}\, f(x_{k+1})$ or its inverse at every iteration, the previous operator is simply updated using the obtained information included in s_k and y_k from the step. As in the Euclidean case, we simply receive RBFGS formula. One exchanges the tangent vectors with each other and now considers an update for the Hessian and not for its inverse, that means on replaces in the formula of $\mathcal{B}_{k+1}^{RDFP}[\cdot]$ the triple $(\mathcal{B}_k^{DFP}, s_k, y_k)$ by $(\mathcal{H}_k^{BFGS}, y_k, s_k)$ to get the update

$$\mathcal{H}_{k+1}^{RBFGS}[\cdot] = \widetilde{\mathcal{H}}_k^{RBFGS}[\cdot] + y_k \frac{y_k^\flat[\cdot]}{s_k^\flat[y_k]} - \widetilde{\mathcal{H}}_k^{RBFGS}[s_k] \frac{s_k^\flat(\widetilde{\mathcal{H}}_k^{RBFGS}[\cdot])}{s_k^\flat(\widetilde{\mathcal{H}}_k^{RBFGS}[s_k])}. \tag{3.3.1}$$

We see that the computation of $\mathcal{H}_{k+1}^{RBFGS}$ requires only first-order information, namely the gradient at x_k and x_{k+l}, a definite advantage over the operator Hess $f(x_{k+1})$ used in Newton's method, which involves second-order information Gabay, 1982, p. 206.

Theorem 3.3.1. *Let* $y_k = \operatorname{grad} f(x_{k+1}) - \mathrm{T}^{\mathrm{retr}}_{x_k,\alpha_k\eta_k}(\operatorname{grad} f(x_k)) \in \mathcal{T}_{x_{k+1}}\mathcal{M}$, $s_k = \mathrm{T}_{x_k,\alpha_k\eta_k}(\alpha_k\eta_k) \in \mathcal{T}_{x_{k+1}}\mathcal{M}$, $s_k \neq 0$ *and assume* T *represents an isometric vector transport. Let* $\mathcal{H}_0^{RBFGS}$ *be any self-adjoint and positive definite operator on* $\mathcal{T}_{x_0}\mathcal{M}$. *Then are all operators of the sequences* $\{\mathcal{H}_k^{RBFGS}\}_k$ *created by the Formula Eq. (3.3.1) self-adjoint and positive definite if and only if*

$$g_{x_{k+1}}(s_k, y_k) > 0. \tag{3.3.2}$$

Proof. See Qi, 2011, Lemma 2.4.1 + Lemma 2.4.2. □

Therefore the RBFGS update produces a series of linear transformations $\mathcal{H}_k^{RBFGS}$ on $\mathcal{T}_{x_k}\mathcal{M}$ that are all self-adjoint and positive definite with respect to the Riemannian metric if an isometric vector transport is used. Note that we have not bounded the condition number of $\mathcal{H}_k^{RBFGS}$ Qi, 2011.

In the Euclidean case we used for the uniqueness of the BFGS formula the insight that it is the unique solution of an optimization problem. This can also be generalized by considering the following

$$\begin{aligned} \min_{\mathcal{B}\colon \mathcal{T}_{x_{k+1}}\mathcal{M} \to \mathcal{T}_{x_{k+1}}\mathcal{M}} \quad & \|\mathcal{B} - \widetilde{\mathcal{B}}_k\|_{\mathcal{W}_\mathcal{H}} \\ \text{s.t.} \quad & \mathcal{B} = \mathcal{B}^*, \qquad \mathcal{B}[y_k] = s_k \end{aligned} \tag{3.3.3}$$

where $\widetilde{\mathcal{B}}_k = \mathrm{T}_{x_{k+1}\leftarrow x_k} \circ \mathcal{B}_k \circ (\mathrm{T}_{x_{k+1}\leftarrow x_k})^{-1}$, $\mathcal{B}_k$ is a self-adjoint and positive definite operator on $\mathcal{T}_{x_k}\mathcal{M}$, $\mathrm{T}_{x_{k+1}\leftarrow x_k}\colon \mathcal{T}_{x_k}\mathcal{M} \to \mathcal{T}_{x_{k+1}}\mathcal{M}$ an isometric vector transport, $s_k, y_k \in \mathcal{T}_{x_{k+1}}\mathcal{M}$ satisfying $g_{x_{k+1}}(s_k, y_k) > 0$, $\mathcal{W}_\mathcal{H}$ a self-adjoint and positive definite operator on $\mathcal{T}_{x_{k+1}}\mathcal{M}$ satisfying $\mathcal{W}_\mathcal{H}[s_k] = y_k$ and $\|\mathcal{A}\|_{\mathcal{W}_\mathcal{H}} = \|\hat{\mathcal{W}}_\mathcal{H}^{\frac{1}{2}} G^{\frac{1}{2}} \hat{\mathcal{A}} G^{-\frac{1}{2}} \hat{\mathcal{W}}_\mathcal{H}^{\frac{1}{2}}\|_{\mathrm{F}}$, G is the matrix expression of the metric and hat denotes matrix expression for the operators $\mathcal{A}$ and $\mathcal{W}_\mathcal{H}$.

see Huang, 2013 p. 204

By using the Sherman–Morrison-Woodbury identity for operators Eq. (3.1.6), on obtains

$$\begin{aligned}\mathcal{B}_{k+1}^{RBFGS}[\cdot] &= \widetilde{\mathcal{B}}_k^{RBFGS}[\cdot] - s_k\frac{y_k^\flat[\widetilde{\mathcal{B}}_k^{RBFGS}[\cdot]]}{y_k^\flat[s_k]} - \widetilde{\mathcal{B}}_k^{RBFGS}[y_k]\frac{s_k^\flat[\cdot]}{s_k^\flat[y_k]} + s_k\frac{y_k^\flat[\widetilde{\mathcal{B}}_k^{RBFGS}[y_k]]s_k^\flat[\cdot]}{(y_k^\flat[s_k])^2} + s_k\frac{s_k^\flat[\cdot]}{s_k^\flat[y_k]} \\ &= \Big(\mathrm{id}[\cdot] - \frac{s_k y_k^\flat[\cdot]}{s_k^\flat[y_k]}\Big)\widetilde{\mathcal{B}}_k^{RBFGS}[\cdot]\Big(\mathrm{id}[\cdot] - \frac{y_k s_k^\flat[\cdot]}{s_k^\flat[y_k]}\Big) + \frac{s_k s_k^\flat[\cdot]}{s_k^\flat[y_k]}\end{aligned} \tag{3.3.4}$$

$$\mathcal{B}_{k+1}^{RBFGS}[\cdot] = \widetilde{\mathcal{B}}_k^{RBFGS}[\cdot] - s_k\frac{y_k^\flat[\widetilde{\mathcal{B}}_k^{RBFGS}[\cdot]]}{y_k^\flat[s_k]} - \widetilde{\mathcal{B}}_k^{RBFGS}[y_k]\frac{s_k^\flat[\cdot]}{s_k^\flat[y_k]} + s_k\frac{y_k^\flat[\widetilde{\mathcal{B}}_k^{RBFGS}[y_k]]s_k^\flat[\cdot]}{(y_k^\flat[s_k])^2} + s_k\frac{s_k^\flat[\cdot]}{s_k^\flat[y_k]}$$

The question remains open what is used as the first approximation of the Hessian $\mathrm{Hess}\, f(x_0)$ or the inverse of the Hessian $(\mathrm{Hess}\, f(x_0))^{-1}$. As in Euclidean, one requires that the initial approximation $\mathcal{H}_0^{RBFGS}$ or $\mathcal{B}_0^{RBFGS}$ is positive definite and self-adjoint. Of course, for reasons of quick availability, the choice often falls on the identity operator $\mathcal{B}_0^{RBFGS} = \mathrm{id}_{\mathcal{T}_{x_0}\mathcal{M}}$ or or the multiplication of the tangent vector by a number $\mathcal{B}_0^{RBFGS}[\eta] = \beta\cdot\eta$. One approach to determine this factor β would be to transfer the formula from the Euclidean, since it is defined only by internal products, i.e. one could choose

$$\beta = \frac{g_{x_1}(y_1, s_1)}{g_{x_1}(y_1, y_1)} \Rightarrow \mathcal{B}_0^{RBFGS} = \frac{g_{x_1}(y_1, s_1)}{g_{x_1}(y_1, y_1)}\cdot \mathrm{id}_{\mathcal{T}_{x_0}\mathcal{M}}$$

and use this for the update to obtain $\mathcal{B}_1^{RBFGS}$ after the search direction has first been calculated using $\mathcal{B}_0^{RBFGS} = \mathrm{id}_{\mathcal{T}_{x_0}\mathcal{M}}$.

3.4 The BFGS Method On Riemannian Manifolds

We now discuss a specific BFGS method for Riemannian manifolds, which can be considered the most natural, since it uses parallel transport as vector transport and the exponential map as retraction. The choice of this combination is intrinsic because it accomplishes the respective tasks, namely the comparison of vectors and smooth differentiation in the Riemannian setup, in the best possible way. We consider here the results of Qi, 2011, which generalize the results of Gabay, 1982, which was also the first work to deal with the BFGS method on Riemannian manifolds.
Since the convergence analysis depends on the choice of retraction and vector transport, we define the algorithm in the corresponding selection of these maps. The choice of parallel transport, P, and exponential map, exp, leads to the following algorithm, a modification of Qi, 2011, Algorithm 2:

We see immediately that the general structure, as it was in Algorithm 4, has not changed. As with the euclidean BFGS algorithm, Algorithm 9 can also be formulated to work with the Hessian approximation, Eq. (3.3.1) rather than with the inverse Hessian approximation B_{k+1}^{RBFGS}. This yields a mathematically

Algorithm 9 Inverse Global RBFGS Method

1: Given starting point $x_0 \in \mathcal{M}$, convergence tolerance $\epsilon > 0$, an initial self-adjoint and positive definite operator $B_0^{RBFGS} : \mathcal{T}_{x_0}\mathcal{M} \to \mathcal{T}_{x_0}\mathcal{M}$, $k = 0$.
2: **while** $\|\operatorname{grad} f(x_k)\|_{x_k} > \epsilon$ **do**
3: Compute search direction $\eta_k = -B_k^{RBFGS}[\operatorname{grad} f(x_k)]$.
4: Find a stepsize α_k that satisfies the (strong) Wolfe conditions.
5: Set $x_{k+1} = \exp_{x_k}(\alpha_k\eta_k)$, $s_k = \mathrm{P}_{x_k,\alpha_k\eta_k}(\alpha_k\eta_k)$, $y_k = \operatorname{grad} f(x_{k+1}) - \mathrm{P}_{x_k,\alpha_k\eta_k}(operatornamegradf(x_k))$.
6: Compute $B_{k+1}^{RBFGS} : \mathcal{T}_{x_{k+1}}\mathcal{M} \to \mathcal{T}_{x_{k+1}}\mathcal{M}$ by means of Eq. (3.3.4).
7: Set $k = k + 1$.
8: **end while**
9: **return** x_k

equivalent algorithm Qi, 2011, p. 13. Again, the question arises which variant is more suitable in practice for the underlying problem. Eq. (3.3.4) makes it possible to cheaply compute an approximation of the inverse of the Hessian. This may make Algorithm 9 advantageous even in the case where we have a cheap exact formula for the Hessian but not for its inverse or when the cost of solving linear systems is unacceptably high.
Algorithm 9 is a globalized algorithm, as the name suggests. That means, a step size $\alpha_k > 0$ is calculated in each iteration. Also in the Riemannian setup one decides to choose a step size that satisfies the Wolfe conditions by using the corresponding retraction and vector transport. However, the most natural way of generalizing the second Wolfe condition to a Riemannian manifold does not guarantee $g_{x_k}(s_k, y_k) > 0$ which is also necessary and sufficient for the positive definite Hessian approximation for Riemannian manifolds.

In fact, any method that uses the Riemannian second Wolfe condition will require at least the action of the differentiated retraction along some particular direction.

It is useful because it makes it possible to cheaply compute an approximation of the inverse of the Hessian.

Now we come to the convergence analysis, which is a generalization of the one from Nocedal, Wright, 2006 for the Riemannian setup. We will first deal with the global convergence of the RBFGS method. We have seen in the Euclidean case that a sufficient condition to achieve global convergence for a convex cost function and local superlinear convergence for a general cost function is preserving the symmetric positive definiteness when updating the Hessian approximation H_k^{BFGS} (or its inverse B_k^{BFGS}) that defines the step Qi, 2011, p. 20-21. In Theorem 3.3.1 we have shown that this is the case for the RBFGS update if and only if the Riemannian curvature condition Eq. (3.2.4) holds in every iteration, which is achieved by the Wolfe conditions (Eq. (3.1.1), Eq. (3.1.2), Eq. (3.1.3)) or strong Wolfe conditions (Eq. (3.1.4), Eq. (3.1.5)). This theorem is used to justify the update step of the algorithm and to show that it preserves the positive-definiteness and self-adjointness of all $\mathcal{H}_k^{RBFGS}$ when the vector transport used is isometric Qi, 2011, p. 23. Of course this is true for the parallel transport P. In order to show global convergence, the following assumptions must be made:

Assumption 3.4.1 (Qi, 2011, Assumptions 2.4.2.). *(i) The objective function f is twice continuously*

differentiable.

(ii) *The level set $\Omega = \{x \in \mathcal{M} : f(x) \leq f(x_0)\}$ is geodesically convex. Let $(\mathcal{M}, g)$ be a Riemannian manifold. A subset C of $\mathcal{M}$ is said to be a geodesically convex set if, given any two points in C, there is a geodesic arc contained within C that joins those two points.*

(iii) *There exists positive constants n and N such that*

$$ng_x(z,z) \leq g_x(G(x)[z], z) \leq Ng_x(z,z) \quad \text{for all } z \in \mathcal{T}_x\mathcal{M} \text{ and } x \in \Omega$$

where $G(x)$ denotes the lifted Hessian $G(x)[\xi] = \operatorname{Hess} \hat{f}_x(\xi) = \operatorname{Hess} f(\operatorname{retr}_x \xi)$.

We see immediately that Assumption 3.4.1 is a generalization for Riemannian manifolds of Assumption 1.4.2. Therefore it is not surprising that the following theorem about global convergence of the RBFGS method follows from a generalization of Theorem 1.4.3.

Theorem 3.4.2 (Qi, 2011, Theorem 2.4.3.). *Let $x_0 \in \mathcal{M}$ be starting point for which Assumption 3.4.1 is satisfied and let $\mathcal{H}_0$ be any linear transformation on $\mathcal{T}_{x_0}\mathcal{M}$ that is self-adjoint and positive definite with respect to the Riemannian metric g. The sequence $\{x_k\}_k$ generated by Algorithm 8 using parallel transport and the exponential map as the retraction converges to the minimizer x^* of f.*

The proof can be found in Qi, 2011, p. 25 and is a generalized version of the proof of Nocedal, Wright, 2006, Theorem 6.5. but nevertheless we want to take a closer look at some important aspects. First, it is important to note that the proof is based on verifying that the search directions and stepsizes satisfy the conditions of Theorem 3.1.9. Since it is exponential map version of the Zoutendijk condition, the line search is restricted to use the exponential map as the retraction to define the next iterate x_{k+1}. We also see that it is assumed that the gradient of the cost function is Lipschitz continuous, which is also expressed by the parallel transport P, and that enforcing the Wolfe conditions enables the strong statement about the angles between the direction vectors and gradients at each step. It should be noted that no restriction is placed on the manner in which the direction vector η_k is generated beyond the assumptions given in the theorem Qi, 2011, p. 23. We can conclude from Theorem 3.1.9, if $\lim_{k\to\infty} \|\operatorname{grad} f(x_k)\| = 0$ holds, that the search directions are never too close to orthogonality with the gradient, i.e. $\cos(\theta_k)^2$ stays away from 0. It follows that the algorithm would achieve global convergence to a set of stationary points. In practice, given the instability of an iteration at stationary points, such an algorithm is often effective at converging to an isolated minimizer when starting close enough, i.e., the global convergence result is used in a local manner Qi, 2011, p. 25. We also note that the convexity of the cost function is only used to guarantee that there is a unique minimizer. One way for this to happen is if f is convex function for the entire domain of interest Qi, 2011, p. 28. Further interesting aspects can be found in the proof of Theorem 3.4.2. Since it is a generalization, one is also interested to avoid establishing a bound on the condition number of the Hessian approximations $\{\mathcal{H}_k^{RBFGS}\}_k$. For that the notion of an average Riemannian Hessian and a function defined in terms of the trace and determinant of a linear operator on a tangent space that is self-adjoint positive definite with respect to the Riemannian metric is introduced, to estimate the size of the largest and smallest eigenvalues of the Hessian approximations. We would like to discuss these three elements. We start with the average Hessian.

$$\tilde{\mathcal{G}}_k = \int_0^1 \mathrm{P}_{x_k,\gamma_{\eta_k}(t)} \circ \mathrm{Hess}\, f(\gamma_{\eta_k}(t)) \circ (\mathrm{P}_{x_k,\gamma_{\eta_k}(t)})^{-1} dt \tag{3.4.1}$$

This clearly shows that the use of parallel transport makes sense for the method and we also assume it for our convergence statement.
After having seen that the method converges globally under certain conditions, we are now interested to know the rate of convergence. In the Euclidean case the procedure converges superlinearly. Therefore we are motivated to get the same result. To show the superlinear convergence of Algorithm 4, Theorem 1.1.7 and especially Eq. (1.1.9) are used. The same methodology will be followed here, but an assumption is still needed:

Assumption 3.4.3 (Qi, 2011, Assumptions 2.4.4.)**.** *Let $x^* \in \mathcal{M}$ be a nondegenerate local minimizer of f, i.e., $\mathrm{grad}\, f(x^*) = 0$ and $\mathrm{Hess}\, f(x^*)$ is positive definite. There is $L > 0$ such that, for all $\xi in \mathcal{T}_{x^*}\mathcal{M}$ and all $\eta in \mathcal{T}_{\mathrm{retr}_{x^*}\xi}\mathcal{M}$ small enough, we have*

$$\|(\mathrm{P}_{\gamma_\eta(t)\leftarrow\gamma_\eta(0)})^{-1} \circ \mathrm{Hess}\, f(y) \circ \mathrm{P}_{\gamma_\eta(t)\leftarrow\gamma_\eta(0)} - \mathrm{P}_{\gamma_\xi(1)\leftarrow\gamma_\xi(0)} \circ \mathrm{Hess}\, f(x^*) \circ (\mathrm{P}_{\gamma_\xi(1)\leftarrow\gamma_\xi(0)})^{-1}\| \leq L \max\{\mathrm{dist}(y, x^*), \mathrm{dist}(x, x^*)\}$$

for $0 \leq t \leq 1$ where $x = \exp_{x^*} \xi$, $y = \exp_x \eta$ and $\gamma_\xi(t)$ and $\gamma_\eta(t)$ are the associated geodesics.

This assumption is ultimately a generalization of 1.3, since this condition is the Riemannian version of Lipschitz continuity of the Hessian $\mathrm{Hess}\, f(x^*)$ of f at x^*.
In Theorem 3.1.10 a requirement on the evolution of the action of in the direction of k relative to the action of the covariant derivative is identified. The requirement is quite general and only requires the transport be twice continuously differentiable.
Let us now turn to the statement on superlinear convergence, which is a generalization of Theorem 1.4.5:

Theorem 3.4.4 (Qi, 2011, Theorem 2.4.5.)**.** *Suppose that f is twice continuously differentiable and that the iterates, x_k, generated by the RBFGS Algorithm using parallel transport and the exponential map converge to a nondegenerate minimizer $x^* \in \mathcal{M}$ at which Assumption 3.4.3 holds. If*

$$\sum_{k=0}^{\infty} \mathrm{dist}(x_k, x^*) < \infty \tag{3.4.2}$$

holds then x_k converges to x^* superlinearly.

The Proof can be found in Qi, 2011, p. 29, there sufficient conditions on the transport and retraction used in the RBFGS method that guarantee the required action of Bk must be identified Qi, 2011, p. 29. Experiments in Qi, 2011 provide substantial evidence that, in practice, both isometric and nonisometric

vector transport achieve superlinear convergence with RBFGS. Theorem 3.1.10 is probably a key part of the explanation for this behavior Qi, 2011, p. 20.

3.5 Cautious BFGS-Method On Riemannian Manifolds

As in the Euclidean, the question arises whether the Riemannian BFGS method converges for general functions. As we have seen in the previous chapter, some kind of Riemannian version of convexity seems necessary at least for global convergence. Therefore, the idea of modifying the method to apply it to a larger class of functions is not out of the question. Again, it is not surprising that ideas which worked well in the Euclidean are transferred to the Riemannian setup.
We now present a method found by Huang, Absil, Gallivan, 2018 which adopts the approach in Li, Fukushima, 2001 for nonconvex problems with a weak line search condition which was introduced in Byrd, Nocedal, 1989.

$$f(\operatorname{retr}_{x_k}\alpha_k\eta_k) - f(x_k) \leq -\chi_1\frac{f'(x_k)^2}{\|\eta_k\|^2} \tag{3.5.1}$$

$$f(\operatorname{retr}_{x_k}\alpha_k\eta_k) - f(x_k) \leq \chi_2 f'(x_k) \tag{3.5.2}$$

The weak line search condition removes completely the need to consider the differentiated retraction.

$$\mathcal{H}_{k+1}^{CRBFGS}[\cdot] = \begin{cases} \widetilde{\mathcal{H}}_k^{CRBFGS}[\cdot] + y_k\frac{y_k^\flat[\cdot]}{s_k^\flat[y_k]} - \widetilde{\mathcal{H}}_k^{CRBFGS}[s_k]\frac{s_k^\flat(\widetilde{\mathcal{H}}_k^{CRBFGS}[\cdot])}{s_k^\flat(\widetilde{\mathcal{H}}_k^{CRBFGS}[s_k])} & \frac{g_{x_{k+1}}(y_k,s_k)}{\|s_k\|^2_{x_{k+1}}} \geq \theta(\|\operatorname{grad} f(x_k)\|) \\ \widetilde{\mathcal{H}}_k^{CRBFGS} & \text{otherwise.} \end{cases} \tag{3.5.3}$$

Algorithm 10 Cautious Riemannian BFGS-Algorithm

1: Riemannian manifold $\mathcal{M}$ with Riemannian metric g; retraction retr; isometric vector transport $\mathrm{T}^{\mathrm{retr}}_{\cdot,\mathcal{S}}(\cdot)$, with retr as the associated retraction; continuously differentiable real-valued function f on $\mathcal{M}$, bounded below; initial iterate $x_0 \in \mathcal{M}$; initial Hessian approximation $\mathcal{H}_0^{CRBFGS}$ that is symmetric positive definite with respect to the metric g; convergence tolerance $\epsilon > 0$; $k = 0$.
2: **while** $\|\operatorname{grad} f(x_k)\| > \epsilon$ **do**
3: Obtain $\eta_k \in \mathcal{T}_{x_k}\mathcal{M}$ by solving $\mathcal{H}_k^{CRBFGS}[\eta_k] = -\operatorname{grad} f(x_k)$.
4: Determine the stepsize $\alpha_k > 0$ that satisfies
5: Set $x_{k+1} = \operatorname{retr}_{x_k}(\alpha_k\eta_k)$.
6: Define $s_k = \mathrm{T}_{x_k,\alpha_k\eta_k}(\alpha_k\eta_k)$ and $y_k = \operatorname{grad} f(x_{k+1}) - \mathrm{T}_{x_k,\alpha_k\eta_k}(\operatorname{grad} f(x_k))$.
7: Define the linear operator $\mathcal{H}_{k+1}^{CRBFGS} : \mathcal{T}_{x_{k+1}}\mathcal{M} \to \mathcal{T}_{x_{k+1}}\mathcal{M}$ by
8: Set $k = k + 1$.
9: **end while**
10: **return** x_k

Assumption 3.5.1 (Huang, Absil, Gallivan, 2018, Assumptions 4.1.+4.2.). *(i) The level set* $\Omega = \{x \in \mathcal{M} : f(x) \leq f(x_0)\}$ *is compact.*

(ii) The function f *is Lipschitz continuously differentiable with respect to the isometric vector transport* T *on* Ω.

Definition 3.5.2 (Huang, Absil, Gallivan, 2018, Definition 4.1.). *Let* $\mathrm{T}_{\cdot\leftarrow(}$)[retr] *be a vector transport associated with a retraction* retr. *A function* f *on* $\mathcal{M}$ *is said to be Lipschitz continuously differentiable with respect to* $\mathrm{T}_{\cdot\leftarrow(}$)[retr] *on* $\mathcal{U} \subset \mathcal{M}$ *if there exists* $L_1 > 0$ *such that*

$$\|\mathrm{T}^{\mathrm{retr}}_{x,\eta}(\operatorname{grad} f(x)) - \operatorname{grad} f(\mathrm{retr}_x\eta)\| L_1\|\eta\|$$

for all $x \in \mathcal{U}$, $\eta \in \mathcal{T}_x\mathcal{M}$ *such that* $\mathrm{retr}_x\eta \in \mathcal{U}$.

Theorem 3.5.3 (Huang, Absil, Gallivan, 2018, Theorem 4.2.). *Let* $\{x_k\}_k$ *be sequences generated by Algorithm 10. If the Assumption 3.5.1 hold, then*

$$\liminf_{k\to\infty} \|\operatorname{grad} f(x_k)\| = 0.$$

Li, Fukushima, 2001, Theorem 3.3.

Assumption 3.5.4 (Huang, Absil, Gallivan, 2018, Assumptions 5.1.). *(i) The objective function* f *is twice continuously differentiable in the level set* Ω.

(ii) The retraction retr *is twice continuously differentiable.*

(iii) The isometric vector transport $\mathrm{T}^{\mathrm{retr}}_{\cdot,\mathcal{S}}(\cdot)$ *with associated retraction* retr *is continuous and satisfies*

the global convergence analysis of the cautious RBFGS does not require a convexity assumption on the cost function.

3.6 Limited-Memory BFGS-Method On Riemannian Manifolds

$$\mathrm{T}^{\mathrm{retr}}_{x,\mathcal{S}\xi}(\xi) = \beta \mathrm{T}^{\mathrm{retr}}_{x,\xi}(\xi), \quad \beta = \frac{\|\xi\|}{\|\mathrm{T}^{\mathrm{retr}}_{x,\xi}(\xi)\|} \tag{3.6.1}$$

$B_k^{(0)}$ can be an arbitrarily, symmetrical and positive definite matrix. In general $B_k^{(0)}$ will be a multiple of the identity matrix, so that it can be stored very easily Geiger, Kanzow, 1999, p. 198. A method for choosing $B_k^{(0)}$ that has proven effective in practice is to set $B_k^{(0)} = \gamma_k I$, where

$$\gamma_k = \frac{s_{k-1}^{\mathrm{T}} y_{k-1}}{y_{k-1}^{\mathrm{T}} y_{k-1}}. \tag{3.6.2}$$

γ_k is the scaling factor that attempts to estimate the size of the true Hessian matrix along the most recent search direction. This choice helps to ensure that the search direction d_k is well scaled, and as a result the stepsize $\alpha_k = 1$ is accepted in most iterations. It is important that the line search is based on the (strict) Wolfe conditions, so that the BFGS updating is stable Nocedal, Wright, 2006, p. 178-179.

The following algorithm is a modified version of Huang, but follows the same structures:

Algorithm 11 Limited-Memory Riemannian BFGS-Algorithm

Riemannian manifold $\mathcal{M}$ with Riemannian metric g, a retraction retr, isometric vector transport $\mathrm{T}^{\mathrm{retr}}_{\cdot,\mathcal{S}}(\cdot)$ that satisfies Eq. (3.6.1), smooth function f on $\mathcal{M}$, initial iterate $x_0 \in \mathcal{M}$, an integer $m > 0$, $k = 0$, $\epsilon > 0$, $0 < c_1 < \frac{1}{2} < c_2 < 1$, $\gamma_0 = 1$, $l = 0$
while $\|\operatorname{grad} f(x_k)\| > \epsilon$ **do**
  $\mathcal{B}_k^{(0)} = \gamma_k \operatorname{id}$. Obtain $\eta_k \in \mathcal{T}_{x_k}\mathcal{M}$ by the following algorithm:
  $q = \operatorname{grad} f(x_k)$
  for $i = k-1, k-2, \cdots, k-m$ **do**
    $\alpha_i = \rho_i s_i^{\mathrm{T}} q$
    $q = q - \alpha_i y_i$
  end for
  $r = \mathcal{B}_k^{(0)}[q]$
  for $i = k-m, k-m+1, \cdots, k-1$ **do**
    $\beta = \rho_i y_i^{\mathrm{T}} r$
    $r = r + s_i(\alpha_i - \beta)$
  end for
  Set $\eta_k = -r$
  Find α_k that satisfies Wolfe conditions Eq. (3.1.1) and Eq. (3.1.2) (or Eq. (3.1.3)).
  Set $x_{k+1} = \operatorname{retr}_{x_k} \alpha_k \eta_k$.
  Set $s_k = \mathrm{T}^{\mathrm{retr}}_{x_k,\mathcal{S},\alpha_k\eta_k}(\alpha_k\eta_k)$, $\beta = \frac{\|\alpha_k\eta_k\|}{\|\mathrm{T}^{\mathrm{retr}}_{x_k,\alpha_k\eta_k}(\alpha_k\eta_k)\|}$, $y_k = \beta^{-1}\operatorname{grad} f(x_{k+1}) - \mathrm{T}^{\mathrm{retr}}_{x_k,\mathcal{S},\alpha_k\eta_k}(\operatorname{grad} f(x_k))$.

  if $k > m$ **then**
    Discard the vector pairs $\{s_{k-m}, y_{k-m}\}$ from storage.
  end if
  Save s_k and y_k.
  Set $k = k+1$.
end while

The following theorem is a generalization of Sun, Yuan, 2006, Theorem 5.7.4.

Theorem 3.6.1. *Let $f\colon \mathcal{M} \to \mathbb{R}$ be a twice continuously differentiable and uniformly convex function. Then the iterative sequence $\{x_k\}_k$ generated by the L-RBFGS-method (Algorithm 11) converges to the unique minimizer x^* of f.*

The following theorem is a generalization of Sun, Yuan, 2006, Theorem 5.7.7.

Theorem 3.6.2. *Let $f : \mathcal{M} \to \mathbb{R}$ be a twice continuously differentiable and uniformly convex function. Assume that the iterative sequence $\{x_k\}_k$ generated by the L-RBFGS-method (??) converges to the unique minimizer x^* of f. Then the rate of convergence is at least R-linear.*

4 Numerics

A practical implementation of RBFGS requires the following ingredients:

(*i*) an efficient numerical representation for points x on $\mathcal{M}$, tangent spaces $\mathcal{T}_x\mathcal{M}$ and the inner products $g_x(\xi_1, \xi_2)$ on $\mathcal{T}_x\mathcal{M}$

(*ii*) an implementation of the chosen retraction $\operatorname{retr}_x \cdot \colon \mathcal{T}_x\mathcal{M} \to \mathcal{M}$

(*iii*) efficient formulas for $f(x)$ and $\operatorname{grad} f(x)$

(*iv*) an implementation of the chosen vector transport $\mathrm{T}_{x,\eta}$ and its inverse $(\mathrm{T}_{x,\eta})^{-1}$

(*v*) a method for solving $\mathcal{H}_k^{RBFGS}[\eta_k] = -\operatorname{grad} f(x_k)$ or alternatively, a method for computing $\eta_k = -\mathcal{H}_k^{RBFGS}[\operatorname{grad} f(x_k)]$.

Qi, 2011

4.1 Realizing the Update-Formula

Hessian Inverse Approximation

Let $\mathcal{M}$ be a Riemannian manifold with $dim(\mathcal{M}) = n$. Let $\{e_1, \cdots, e_n\}$ be a canonical basis of $\mathcal{T}_{x_{k+1}}\mathcal{M}$. Let $mathcalB_{k+1}^{RBFGS} = \{\lfloor_1^{k+1}, \cdots, \lfloor_n^{k+1}\}$. Let $\eta \in \mathcal{T}_{x_{k+1}}\mathcal{M}$.

$$\mathcal{B}_{k+1}^{RBFGS}[\eta] = g_{x_{k+1}}(\lfloor_1^{k+1}, \eta) \cdot e_1 + \cdots + g_{x_{k+1}}(\lfloor_n^{k+1}, \eta) \cdot e_n.$$

$$\lfloor_i^{k+1} = \widetilde{\lfloor}_i^k - \frac{g_{x_{k+1}}(e_i, s_k)}{g_{x_{k+1}}(s_k, y_k)} \cdot \widetilde{\mathcal{B}}_k^{RBFGS}[y_k] - \frac{g_{x_{k+1}}(\widetilde{\mathcal{B}}_k^{RBFGS}[y_k], e_i)}{g_{x_{k+1}}(s_k, y_k)} \cdot s_k + \frac{g_{x_{k+1}}(y_k, \widetilde{\mathcal{B}}_k^{RBFGS}[y_k]) \cdot g_{x_{k+1}}(s_k, e_i)}{(g_{x_{k+1}}(s_k, y_k))^2} \cdot s_k + \frac{g_{x_{k+1}}(s_k, e_i)}{g_{x_{k+1}}(s_k, y_k)} \cdot s_k$$

for $i = 1, \cdots, n$. We define

$$\widetilde{\mathcal{B}}_k^{RBFGS}[y_k] = g_{x_{k+1}}(\widetilde{\lfloor}_1^k, y_k) \cdot e_1 + \cdots + g_{x_{k+1}}(\widetilde{\lfloor}_n^k, y_k) \cdot e_n$$

where $\widetilde{\lfloor}_i^k = \mathrm{P}_{x_{k+1} \leftarrow x_k}(\lfloor_i^k)$.

Hessian Approximation

Let $\mathcal{M}$ be a Riemannian manifold with $dim(\mathcal{M}) = n$. Let $\{e_1, \cdots, e_n\}$ be a canonical basis of $\mathcal{T}_{x_{k+1}}\mathcal{M}$. Let $mathcalH_{k+1}^{RBFGS} = \{\langle_1^{k+1}, \cdots, \langle_n^{k+1}\}$. Let $\eta \in \mathcal{T}_{x_{k+1}}\mathcal{M}$.

$$\mathcal{H}_{k+1}^{RBFGS}[\eta] = g_{x_{k+1}}(\langle_1^{k+1}, \eta) \cdot e_1 + \cdots + g_{x_{k+1}}(\langle_n^{k+1}, \eta) \cdot e_n.$$

$$\langle_i^{k+1} = \widetilde{\langle}_i^k - \frac{g_{x_{k+1}}(\widetilde{\langle}_i^k, s_k)}{g_{x_{k+1}}(s_k, \widetilde{\mathcal{H}}_k^{RBFGS}[s_k])} \cdot \widetilde{\mathcal{H}}_k^{RBFGS}[s_k] + \frac{g_{x_{k+1}}(e_i, y_k)}{g_{x_{k+1}}(y_k, s_k)} \cdot y_k$$

for $i = 1, \cdots, n$. We define

$$\widetilde{\mathcal{H}}_k^{RBFGS}[s_k] = g_{x_{k+1}}(\widetilde{\langle}_1^k, s_k) \cdot e_1 + \cdots + g_{x_{k+1}}(\widetilde{\langle}_n^k, s_k) \cdot e_n$$

where $\widetilde{\langle}_i^k = \mathrm{P}_{x_{k+1} \leftarrow x_k}(\langle_i^k)$.

see Qi, 2011, Chapter 3.3

5 Conclusion

Literature

Bibliography

Absil, P.-A.; R. Mahony; R. Sepulchre (2008). *Optimization Algorithms on Matrix Manifolds*. Princeton University Press.

Broyden, C. G. (1967). "Quasi-Newton Methods and their Application to Function Minimisation". *Mathematics of Computation* 21.99, pp. 368–381.

Byrd, R. H.; J. Nocedal (1989). "A Tool for the Analysis of Quasi-Newton Methods with Application to Unconstrained Minimization". *SIAM Journal on Numerical Analysis* 26.3, pp. 727–739.

Cruz Neto, J.; I. Melo; P. Sousa (2017). "Convexity and Some Geometric Properties". *Journal of Optimization Theory and Applications* 173. DOI: 10.1007/s10957-017-1087-2.

Dai, Y.-H. (2002). "Convergence Properties of the BFGS Algoritm". *SIAM Journal on Optimization* 13, pp. 693–701. DOI: 10.1137/S1052623401383455.

Dai, Y.-H. (2012). "A perfect example for the BFGS method". *Mathematical Programming* 138. DOI: 10.1007/s10107-012-0522-2.

Davidon, W. C. (1959). "VARIABLE METRIC METHOD FOR MINIMIZATION". DOI: 10.2172/4252678.

Deng, C. (2011). "A generalization of the Sherman-Morrison-Woodbury formula". *Appl. Math. Lett.* 24, pp. 1561–1564. DOI: 10.1016/j.aml.2011.03.046.

Dennis, J.; J. J. Moré (1974). "Quasi-Newton Methods, Motivation and Theory". *Siam Review* 19, pp. 46–89.

Fletcher, R. (1970). "A new approach to variable metric algorithms". *The Computer Journal* 13.3, pp. 317–322. DOI: 10.1093/comjnl/13.3.317.

Fletcher, R.; M. J. D. Powell (1963). "A Rapidly Convergent Descent Method for Minimization". DOI: 10.1093/comjnl/6.2.163.

Gabay, D. (1982). "Minimizing a differentiable function over a differential manifold". *Journal of Optimization Theory and Applications* 37. DOI: 10.1007/BF00934767.

Geiger, C.; C. Kanzow (1999). *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben*. Springer. DOI: 10.1007/978-3-642-58582-1.

Goldfarb, D. (1970). "A family of variable-metric methods derived by variational means".

Huang, W. (2013). *Optimization Algorithms on Riemannian Manifolds with Applications*. Florida State University.

Huang, W.; P.-A. Absil; K. A. Gallivan (2018). "A Riemannian BFGS Method without Differentiated Retraction for Nonconvex Optimization Problems". *SIAM Journal on Optimization* 28.1, pp. 470–495.

Li, D.-H.; M. Fukushima (2001). "On the Global Convergence of the BFGS Method for Nonconvex Unconstrained Optimization Problems". *SIAM Journal on Optimization* 11.4, pp. 1054–1064. DOI: 10.1137/s1052623499354242.

Mascarenhas, W. (2004). "The BFGS method with exact line searches fails for non-convex objective functions". *Math. Program.* 99, pp. 49–61. DOI: 10.1007/s10107-003-0421-7.

Nocedal, J.; S. J. Wright (2006). *Numerical Optimization*. Second Edition. Springer. DOI: 10.1007/978-0-387-40065-5.

Powell, M. D. (1976). "Some global convergence properties of a variable metric algorithm for minimization without exact lin".

Qi, C. (2011). "Numerical Optimization Methods on Riemannian Manifolds". Florida State University.

Ring, W.; B. Wirth (2012). "Optimization Methods on Riemannian Manifolds and Their Application to Shape Space". *SIAM Journal on Optimization* 22. DOI: 10.1137/11082885X.

Sato, H.; T. Iwai (2015). "A new, globally convergent Riemannian conjugate gradient method". *Optimization* 64.4, pp. 1011–1031. DOI: 10.1080/02331934.2013.836650.

Shanno, D. F. (1970). "Conditioning of Quasi-Newton Methods for Function Minimization". *Mathematics of Computation* 24.111, pp. 647–656.

Sun, W.; Y.-X. Yuan (2006). *Optimization Theory and Methods: Nonlinear Programming*. Vol. 1. Springer. DOI: 10.1007/b106451.

Ulbrich, M.; S. Ulbrich (2012). *Nichtlineare Optimierung*. Springer. DOI: 10.1007/978-3-0346-0654-7.

Warth, W.; J. Werner (1977). "Effiziente Schrittweitenfunktionen bei unrestringierten Optimierungsaufgaben". *Computing* 19, pp. 59–72. DOI: 10.1007/BF02260741.

Werner, J. (1978/79). "Über die globale Konvergenz von Variable-Metrik-Verfahren mit nicht-exakter Schrittweitenbestimmung." *Numerische Mathematik* 31, pp. 321–334.

Zhang, H.; S. Sra (2016). "First-order Methods for Geodesically Convex Optimization". *29th Annual Conference on Learning Theory*. Ed. by V. Feldman; A. Rakhlin; O. Shamir. Vol. 49. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR, pp. 1617–1638.