On step sizes rules for the continuous stochastic gradient scheme

Convergence results for constant step sizes and an Armijo-type line search

Max Grieshammer · Lukas Pflug · Michael Stingl · Andrian Uihlein

Received: date / Accepted: date

Abstract The recently proposed continuous stochastic gradient descent (CSG) scheme represents an efficient method for the solution of a large class of stochastic optimization problems. During the course of the iterations, CSG uses a tailored convex combination of previously drawn gradient and objective function samples to approximate the full gradient and objective function value at the current position. It was shown that the error in these approximations tends to zero during the optimization process and thus, CSG more and more behaves like a full gradient descent algorithm. In this contribution, we show that convergence results for constant step sizes known from full gradient descent schemes carry over to CSG. Furthermore, we augment CSG by an Armijo-type backtracking line search based on the gradient and objective function approximations and provide convergence results for this variant. The resulting novel methods are applicable for a large class of optimization problems. Their performance is demonstrated in comparison to other stochastic gradient schemes known from literature.

Keywords Stochastic gradient scheme · Convergence analysis · Step size rule · Backtracking line search · Constant step size

Mathematics Subject Classification (2020) 65K05 · 90C06 · 90C15 · 90C30

Max Grieshammer · Michael Stingl · Andrian Uihlein
Chair of Applied Mathematics (Continuous Optimization), Friedrich-Alexander Universität Erlangen-Nürnberg (FAU), Cauerstr. 11, 91058 Erlangen, Germany
E-mail: max.grieshammer@fau.de, michael.stingl@fau.de, andrian.uihlein@fau.de

Lukas Pflug
Competence Unit for Scientific Computing, Friedrich-Alexander Universität Erlangen-Nürnberg (FAU), Martensstr. 5a, 91058 Erlangen, Germany
E-mail: lukas.pflug@fau.de
1 Introduction

In this contribution, we consider optimization problems with an expected-valued cost function, i.e.,
\[
\min_{u \in U} \int_X j(u, x) \mu(dx),
\]
(1.1)
with an admissible set $U$, a parameter set $X$, $j \in C^1(U \times X; \mathbb{R})$ and a (probability) measure $\mu$. Such problems arise naturally in the context of machine learning and noisy simulations [1, 2]. They also appear in other large scale optimization problems, such as topology or shape optimization, where one is interested in a design that is optimal for a continuous spectrum of outer parameters, e.g. [3–5]. One usually does not consider deterministic approaches for the solution of this type of problems, as they are generally too computationally expensive or even intractable. Instead, one uses stochastic optimization schemes like stochastic gradient descent (SG) [6] or stochastic average gradient descent (SAG) [7], to name the most prominent examples.

The continuous stochastic gradient descent (CSG) method, recently proposed in [8], shares properties with both, stochastic, as well as full gradient methods. Contrary to SG, in CSG the gradient samples $\nabla j(u_n, x_n)$ and objective functional samples $j(u_n, x_n)$ drawn in the $n$-th iteration are not discarded after each iteration. Instead, a tailored convex combination of said samples is used, in order to obtain an approximation of the full gradient and the objective function value. In [9], several methods of obtaining the weights of this convex combination were presented and analyzed in an even more general setting than (1.1). The results presented in this contribution can also be generalized in the same fashion. However, for the sake of readability and since our focus lies on step size rules instead, we consider only problems of the form (1.1) and refer to [9, Remark 2.3] for further details.

A key advantage of CSG against most stochastic optimization methods is that the approximation error in both the gradient and the objective function converges to 0 during the optimization process (Lemma 2.6), which has already been shown for diminishing step sizes in [9, Theorem 4.7]. Thus, in the course of the iterations, CSG more and more behaves like a full gradient descent algorithm. Therefore, the question which convergence properties of gradient descent carry over to CSG, arises naturally. Special interest lies in the known disadvantages of SG-type methods, like the lack of efficient stopping criteria (cf. [10]) and an advanced choice of step sizes, or learning rate, as it is often called in the machine learning context.

While stopping criteria based on approximate first order conditions are rather straight-forward to implement thanks to Lemma 2.6, we want to further discuss the step size dependency of SG methods. In most cases, finding an optimal step size requires knowledge of properties of the function which are usually not available. On the other hand, simpler step size rules, like constant step sizes, come with very strict assumptions with respect to (1.1). As an example, linear convergence rates for strongly convex objective functions with a finite sum structure (discrete measure $\mu$ in (1.1)) were proven under the proposed strong growth condition in [11, Section 6]. However, this condition is (as stated by the authors) not satisfied in most applications, as it requires any optimal solution of (1.1) to be stationary for each individual component of the objective function. An overview of different convergence results for adaptive or fixed step size rules can be found in [12–17] and the references therein.
In this contribution, we show that CSG is able to work with step size rules known from classical deterministic optimization methods. We prove that every accumulation point of the sequence generated by CSG carried out with a sufficiently small constant step size is a stationary point of (1.1) and that the sequence in fact converges for non-pathological (see Remark 3.2) cases.

Furthermore, we introduce a backtracking Armijo-type line search based on gradient and objective function approximations and prove that this yields the same convergence results as for the constant step sizes mentioned above. Thus, the resulting optimization method is widely independent of the step size rule.

Before going into detail, we want to state an informal version of the convergence results established later in this work. The full set of assumptions can be found in Section 2.1 and the main convergence results are given in Theorem 3.1, Theorem 3.2 and Theorem 4.1.

Main Result Assume that the objective function in (1.1) has a Lipschitz continuous gradient and is not chosen as a pathological counter example.

Then every accumulation point of the sequence generated by CSG with sufficiently small constant step size or backtracking line search is stationary. Moreover, we obtain convergence in the objective function values.

Structure of the paper

In Section 2, we introduce the quantities appearing in (1.1) and our needed assumptions in detail. Additionally, we provide auxiliary results for later sections. Section 3 contains our convergence results for CSG used with constant step sizes. In Section 4, we introduce CSG with a backtracking scheme and provide convergence results for this method. In Section 5, a heuristic scheme to approximate the needed Lipschitz constant is presented. Lastly, in Section 6, we document numerical experiments that analyze the stability of the new CSG versions. Furthermore, we combine backtracking CSG with the heuristic scheme to approximate the Lipschitz constant to obtain a method that does not require any step size rule at all and test the stability of this algorithm for different problems.

2 Assumptions And Auxiliary Results

In this section, we introduce our general setting and formulate the assumptions we make throughout this contribution. Additionally, the basic CSG algorithm is presented and some preliminary results are stated.

2.1 Setting

Definition 2.1 (Objective function) For $d_o, d_r \in \mathbb{N}$, we introduce the set of admissible optimization variables $\mathcal{U} \subset \mathbb{R}^{d_o}$ and the parameter set $\mathcal{X} \subset \mathbb{R}^{d_r}$. The objective function $J : \mathcal{U} \to \mathbb{R}$ is then given by

$$J(u) := \mathbb{E}[j(u,X)] = \int_{\mathcal{X}} j(u,x) \mu(dx),$$
where we assume $j \in C^1(\mathcal{U} \times \mathcal{X}; \mathbb{R})$ to be measurable and $X \sim \mu$.

Since $\mathcal{U}$ and $\mathcal{X}$ are finite dimensional, we do not have to consider specific norms on these spaces and can instead choose them problem specific. In the following, we will denote them simply by $\| \cdot \|_{\mathcal{U}}$ and $\| \cdot \|_{\mathcal{X}}$.

During the optimization, we need to draw random samples of the random variable $X$. As stated in the following assumption, we need the sequence of samples $(x_n)_{n \in \mathbb{N}} \subset \mathcal{X}$ to be dense in $\text{supp}(\mu)$. This, however, is automatically satisfied with probability 1, as can be seen in Remark 2.1.

**Assumption 2.1 (Sample sequence)** The sequence $(x_n)_{n \in \mathbb{N}}$ is dense in $\text{supp}(\mu)$, where

$$\text{supp}(\mu) := \{ x \in \mathcal{X} : \mu(B_\epsilon(x)) > 0 \text{ for all } \epsilon > 0 \}.$$  

Here, $B_\epsilon(x)$ denotes the open ball with radius $\epsilon > 0$ centered at $x \in \mathcal{X}$ and $(x_n)_{n \in \mathbb{N}}$ is a sequence of independent identically distributed realizations of the random variable $X \sim \mu$.

Notice that $\text{supp}(\mu)$ is a closed subset of $\mathbb{R}^d$.

**Remark 2.1** Let $x \in \text{supp}(\mu)$ and $\epsilon > 0$. Then, given an independent identically distributed sequence $x_1, x_2, \ldots \sim \mu$, we have

$$P(x_n \notin B_\epsilon(x)) = 1 - \mu(B_\epsilon(x)) < 1.$$ 

Hence, by the Borel-Cantelli Lemma [18, Theorem 2.7],

$$P(x_n \notin B_\epsilon(x) \text{ for all } n \in \mathbb{N}) = 0.$$ 

Thus, the sequence $(x_n)_{n \in \mathbb{N}}$ is dense in $\text{supp}(\mu)$ with probability 1.

As for the original CSG method, the sets $\mathcal{U}$ and $\mathcal{X}$ need to satisfy the following regularity conditions, compare [9, Assumption 2.7]:

**Assumption 2.2 (Regularity of $\mathcal{U}$, $\mathcal{X}$ and $\mu$)** The set $\mathcal{U} \subset \mathbb{R}^d$ is compact and convex. The set $\mathcal{X} \subset \mathbb{R}^d$ is open and bounded with $\text{supp}(\mu) \subset \mathcal{X}$. This can always be achieved, as long as $\text{supp}(\mu) \subset \mathbb{R}^d$ is bounded.

Finally, as in the deterministic case, we assume the gradient of the objective function to be Lipschitz continuous, in order to obtain convergence for constant step sizes.

**Assumption 2.3 (Regularity of $\nabla J$)** $J : \mathcal{U} \to \mathbb{R}$ is $L$-smooth, i.e., there exists $L > 0$ such that

$$\|\nabla J(u_1) - \nabla J(u_2)\|_{\mathcal{U}} \leq L\|u_1 - u_2\|_{\mathcal{U}} \text{ for all } u_1, u_2 \in \mathcal{U}.$$
Algorithm 1 General CSG method

1: \textbf{while} Termination condition not met \textbf{do}
2: \hspace{1em} Sample objective function (optional):
3: \hspace{2.5em} \( j_n := j(u_n, x_n) \)
4: \hspace{1em} Sample gradient:
5: \hspace{2.5em} \( g_n := \nabla_j j(u_n, x_n) \)
6: \hspace{1em} Calculate weights \( \alpha_k \)
7: \hspace{2.5em} Calculate search direction:
8: \hspace{2.5em} \( \hat{G}_n := \sum_{k=1}^{n} \alpha_k g_k \)
9: \hspace{1em} Compute objective function value approximation (optional):
10: \hspace{2.5em} \( \hat{J}_n := \sum_{k=1}^{n} \alpha_k j_k \)
11: \hspace{1em} Choose stepsize \( \tau_n \)
12: \hspace{2.5em} Gradient step:
13: \hspace{2.5em} \( u_{n+1} := P_U (u_n - \tau_n \hat{G}_n) \)
14: \hspace{1em} Update index:
15: \hspace{2.5em} \( n \leftarrow n + 1 \)
16: \textbf{end while}

The general CSG method was first proposed in [8] and can be used to solve problems of the form (1.1). Additional information regarding convergence results and the weight calculation mentioned in Step 4 can be found in [9]. We note that the results presented in this contribution can, as proposed in [9], be generalized to a wider class of problems than (1.1). However, in this work, we want to focus solely on step sizes rules and therefore work only in the simpler setting (1.1).

2.2 The CSG Algorithm

For a given starting point \( u_1 \in \mathcal{U} \) and a sample sequence \( x_1, x_2, \ldots \) as stated in Assumption 2.1, the general CSG algorithm was introduced in [8] and is stated in Algorithm 1. In [9, Section 3.1] it was thoroughly analyzed how the computation of the weights (Step 4) can be carried out.

In Step 8, \( P_\mathcal{U} \) denotes the orthogonal projection (in the sense of \( \| \cdot \|_\mathcal{U} \)), i.e.,

\[
P_\mathcal{U}(u) := \arg \min_{\tilde{u} \in \mathcal{U}} \| u - \tilde{u} \|_\mathcal{U}.
\]

We will later show that the CSG method converges to stationary points, which we define next.

**Definition 2.2 (Stationary points)** Let \( J \in C^1(\mathcal{U}) \) be given. We say \( u^* \in \mathcal{U} \) is a stationary point of \( J \), if there exists \( t > 0 \) such that

\[
P_\mathcal{U} \left( u^* - t \nabla J(u^*) \right) = u^*.
\]

Furthermore, we denote by

\[
S(J) := \{ u \in \mathcal{U} : P_\mathcal{U} \left( u - t \nabla J(u) \right) = u \text{ for some } t > 0 \}
\]

the set of all stationary points of \( J \).
2.3 Auxiliary Results

Gradient descent methods for $L$-smooth objective functions have thoroughly been studied in the past (e.g. [19] [20]). The key ingredients for obtaining convergence results with constant step sizes are the descent lemma and the characteristic property of the projection operator, which we state in the following.

Lemma 2.1 (Descent Lemma) If $J : \mathcal{U} \to \mathbb{R}$ is $L$-smooth, then it holds

$$J(u_1) \leq J(u_2) + \nabla J(u_2)^\top (u_1 - u_2) + \frac{L}{2} \|u_1 - u_2\|^2 \quad \forall u_1, u_2 \in \mathcal{U}.$$ 

Lemma 2.2 (Characteristic Property of Projection) If $u_{n+1} = \mathcal{P}_\mathcal{U}(u_n - \tau_n \hat{G}_n)$, then

$$\hat{G}_n(u_n - u_{n+1}) \geq \frac{\|u_n - u_{n+1}\|^2}{\tau_n}.$$ 

Proof Lemma 2.1 corresponds to [21, Lemma 5.7]. Lemma 2.2 is a direct consequence of [21, Theorem 6.41].

Before we move on to results that are specific for the CSG method, we state a general convergence result, which will be helpful in the later proofs.

Lemma 2.3 (Finitely many accumulation points) Let $(u_n)_{n \in \mathbb{N}} \subset \mathbb{R}^d$ be a bounded sequence. Suppose that $(u_n)_{n \in \mathbb{N}}$ has only finitely many accumulation points and it holds $\|u_{n+1} - u_n\| \to 0$. Then $(u_n)_{n \in \mathbb{N}}$ is convergent.

Proof Let $\{\bar{u}_1, \ldots, \bar{u}_K\}$ be the accumulation points of $(u_n)_{n \in \mathbb{N}}$ and define

$$\delta_0 := \min_{i,j \in \{1, \ldots, K\}} \|\bar{u}_i - \bar{u}_j\|,$$

i.e., the minimal distance between two accumulation points of $(u_n)_{n \in \mathbb{N}}$. The accumulation point closest to $u_n$ is defined as:

$$\bar{u}(n) := \arg\min_{u \in \{\bar{u}_1, \ldots, \bar{u}_K\}} \|u_n - u\|.$$ 

Next up, we show that there exists $N \in \mathbb{N}$ such that for all $n \geq N$ it holds $\|u_n - \bar{u}(n)\| < \frac{\delta_0}{4}$. We prove this by contradiction.

Thus, we assume there exist infinitely many $n \in \mathbb{N}$ such that $\|u_n - \bar{u}(n)\| \geq \frac{\delta_0}{4}$.

This subsequence is again bounded and therefore must have an accumulation point. By construction, this accumulation point is no accumulation point of $(u_n)_{n \in \mathbb{N}}$, which is a contradiction.

Now, let $N_1 \in \mathbb{N}$ be large enough such that $\|u_n - \bar{u}(n)\| < \frac{\delta_0}{4}$ for all $n \geq N_1$. By our assumptions, there also exists $N_2 \in \mathbb{N}$ with $\|u_{n+1} - u_n\| < \frac{\delta_0}{4}$ for all $n \geq N_2$. Define $N := \max\{N_1, N_2\}$.

Assume $n \geq N$ and $\bar{u}(n) \neq \bar{u}(n + 1)$ we obtain

$$\text{dist}(B_{\delta_0/4}(\bar{u}(n)), B_{\delta_0/4}(\bar{u}(n + 1))) \geq \frac{\delta_0}{2} > \frac{\delta_0}{4} > \|u_n - u_{n+1}\| \quad \text{for all } n \geq N,$$

where $\text{dist}(A, B) := \inf_{x \in A, y \in B} \|x - y\|$ for $A, B \subset \mathbb{R}^d$. This is a contradiction to $\|u_n - u_{n+1}\| < \frac{\delta_0}{4}$ for all $n \geq N$.

We thus conclude that $u_n \in B_{\frac{\delta_0}{4}}(\bar{u}(n))$ implies $u_{n+1} \in B_{\frac{\delta_0}{4}}(\bar{u}(n))$ as well. Since $\bar{u}(n)$ is the only accumulation point on $B_{\frac{\delta_0}{4}}(\bar{u}(n))$, it follows that $u_n \to \bar{u}(N)$. ∎
2.4 Results for CSG approximations

From now on, let \((u_n)_{n \in \mathbb{N}}\) denote the sequence of iterates generated by Algorithm 1. In this section, we want to show that the CSG approximations \(\hat{J}_n\) and \(\hat{G}_n\) in the course of iterations approach the values of \(J(u_n)\) and \(\nabla J(u_n)\), respectively. While this has already been shown for diminishing step sizes in \([9, \text{Theorem 4.7}]\), here, for the first time, a proof for constant step sizes is provided. This is a key result for the convergence theorems stated in Section 3 and Section 4.

Lemma 2.4 (Density result in \(X\)) Let \((x_n)_{n \in \mathbb{N}}\) be the random sequence appearing in Algorithm 1. For all \(\varepsilon > 0\) there exists \(N \in \mathbb{N}\) such that

\[
\min_{n \in \{1, \ldots, N\}} \|x_n - x\|_X < \varepsilon \quad \text{for all } x \in \text{supp}(\mu).
\]

Proof Utilizing the compactness of \(\text{supp}(\mu) \subset \mathbb{R}^d\), there exists \(T \in \mathbb{N}\) such that \((B_{\varepsilon/2}(m_i))_{i=1, \ldots, T}\) is an open cover of \(\text{supp}(\mu)\) consisting of balls with radius \(\varepsilon/2\) centered at points \(m_i \in \text{supp}(\mu)\). Thus, for each \(x \in \text{supp}(\mu)\) we can find \(i_x \in \{1, \ldots, T\}\) with \(x \in B_{\varepsilon/2}(m_{i_x})\). Hence, by Assumption 2.1, for each \(i = 1, \ldots, T\), there exists \(n_i \in \mathbb{N}\) satisfying

\[
\|x_{n_i} - m_i\|_X < \frac{\varepsilon}{2}.
\]

Defining

\[
N := \max_{i \in \{1, \ldots, T\}} n_i < \infty,
\]

for all \(x \in \text{supp}(\mu)\) we have

\[
\min_{n \in \{1, \ldots, N\}} \|x - x_n\|_X \leq \min_{n \in \{1, \ldots, N\}} (\|x - m_{i_x}\|_X + \|m_{i_x} - x_n\|_X) < \frac{\varepsilon}{2} + \min_{n \in \{1, \ldots, N\}} \|m_{i_x} - x_n\|_X < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.
\]

Lemma 2.5 (Density result in \(U \times X\)) Let \((u_n)_{n \in \mathbb{N}}, (x_n)_{n \in \mathbb{N}}\) be the sequences of optimization variables and sample sequence appearing in Algorithm 1. For all \(\varepsilon > 0\) there exists \(N \in \mathbb{N}\) such that

\[
Z_n(x) := \min_{k \in \{1, \ldots, n\}} (\|u_n - u_k\|_U + \|x - x_k\|_X) < \varepsilon
\]

for all \(n > N\) and all \(x \in \text{supp}(\mu)\).

Proof Since \(U\) is compact, we can find a finite cover \((B_{\varepsilon/4}(m_i))_{i=1, \ldots, T}\) of \(U\) consisting of \(T \in \mathbb{N}\) balls with radius \(\varepsilon/2\) centered at points \(m_i \in U\). Define \(I \subset \{1, \ldots, T\}\) as

\[
I := \{i \in \{1, \ldots, T\} : u_n \in B_{\varepsilon/4}(m_i) \text{ for infinitely many } n \in \mathbb{N}\}.
\]

By our definition of \(I\), for each \(i \in \{1, \ldots, T\}\) \(\setminus I\) there exists \(\tilde{N}_i \in \mathbb{N}\) such that \(u_n \not\in B_{\varepsilon/4}(m_i)\) for all \(n > \tilde{N}_i\). Setting

\[
N_1 := \max_{i \in \{1, \ldots, T\} \setminus I} \tilde{N}_i,
\]
it follows that
\[ u_x \notin B_{\epsilon/4}(m_i) \text{ for each } n > N_1 \text{ and all } i \in \{1, \ldots, T\} \setminus I. \tag{2.1} \]

For \( i \in I \) let \( (u_{n_i}, t \in \mathbb{N}) \) be the subsequence consisting of all elements of \((u_n)_{n \in \mathbb{N}}\) that lie in \( B_{\epsilon/4}(m_i) \). Since \((x_n)_{n \in \mathbb{N}}\) is i.i.d. and dense in \( \text{supp}(\mu) \), so is \((x_{n_i})_{t \in \mathbb{N}}\). By Lemma 2.4 we can find \( K_i \in \mathbb{N} \) such that
\[ \min_{t \in \{1, \ldots, K_i\}} \|x - x_{n_i}\|_x < \frac{\epsilon}{2} \text{ for all } x \in \text{supp}(\mu). \tag{2.2} \]

Define
\[ N_2 := \max_{i \in I} \max_{t \in \{1, \ldots, K_i\}} n_{i,t} \]
as well as \( N := \max(N_1, N_2) \). By (2.1), for all \( n > N \) there exists \( i \in I \) such that
\[ \|u_n - m_i\|_\mu < \frac{\epsilon}{4}. \tag{2.3} \]

Now, for every \( x \in \text{supp}(\mu) \) and all \( n > N \) it holds
\[
Z_n(x) = \min_{k \in \{1, \ldots, n\}} \left( \|u_n - u_k\|_\mu + \|x - x_k\|_x \right) \\
\leq \min_{t \in \{1, \ldots, K_i\}} \left( \|u_n - u_{n_{i,t}}\|_\mu + \|x - x_{n_{i,t}}\|_x \right) \\
\leq \min_{t \in \{1, \ldots, K_i\}} \left( \|u_n - m_i\|_\mu + \|m_i - u_{n_{i,t}}\|_\mu + \|x - x_{n_{i,t}}\|_x \right) \\
< \frac{\epsilon}{4} + \frac{\epsilon}{4} + \frac{\epsilon}{2} = \epsilon,
\]
where we used (2.2), (2.3) and \( u_{n_{i,t}} \in B_{\epsilon/4}(m_i) \) in the last line. \( \square \)

**Lemma 2.6 (Approximation results for \( J \) and \( \nabla J \))**

For all \( \epsilon > 0 \) there exists \( N \in \mathbb{N} \) such that
\[
\|\hat{G}_n - \nabla J(u_n)\| < \epsilon \quad \text{and} \quad \|\hat{J}_n - J(u_n)\| < \epsilon \quad \text{for all } n > N
\]

**Proof** In [19] it was shown that
\[
\|\hat{G}_n - \nabla J(u_n)\| \leq L \left( \frac{1}{n} \sum_{i=1}^{n} Z_n(x_i) + d_W(\mu_n, \mu) \right),
\]
where \( d_W(\mu_n, \mu) \) denotes the Wasserstein distance of the measure \( \mu_n \) and \( \mu \) (see [22]). Since \( d_W(\mu_n, \mu) \to 0 \) (e.g., [23, Theorem 3]), there is \( n_2 \in \mathbb{N} \) satisfying
\[
d_W(\mu_n, \mu) < \frac{\epsilon}{2\pi} \quad \text{for all } n > N_2.
\]

By Lemma 2.5 we can also find \( N_1 \in \mathbb{N} \) such that
\[ Z_n(x) < \frac{\epsilon}{2\pi} \quad \text{for all } x \in \text{supp}(\mu) \text{ and all } n > N_1.
\]

Combining the results above yields
\[ \|\hat{G}_n - \nabla J(u_n)\| < \epsilon \quad \text{for all } n > N := \max(N_1, N_2). \]

By replacing \( L \) with the Lipschitz constant of \( J \), the same steps as above yield the second part of the claim. \( \square \)
3 Convergence Results For Constant Stepsize

In this section, we assume Assumptions 2.1 to 2.3 to be satisfied. Our first result considers the special case in which the objective function $J$ appearing in (1.1) has only finitely many stationary points on $U$. The proof of this result serves as a prototype for the later convergence results, as they share a common idea.

**Theorem 3.1 (Convergence for constant steps)** Assume that $J$ has only finitely many stationary points on $U$.

Then CSG with a positive constant stepsize $\tau_n = \tau < \frac{2}{L}$ converges to a stationary point of $J$.

We want to sketch the proof of Theorem 3.1, before going into details. In the deterministic case, Lemma 2.1 and Lemma 2.2 are used to show that $J(u_{n+1}) \leq J(u_n)$ for all $n \in \mathbb{N}$. It then follows from a telescopic sum argument that $\|u_{n+1} - u_n\| \to 0$, i.e., every accumulation point of $(u_n)_{n \in \mathbb{N}}$ is stationary (compare [20, Theorem 5.1] or [21, Theorem 10.15]).

In the case of CSG, we cannot guarantee monotonicity of the objective function values $(J(u_n))_{n \in \mathbb{N}}$. Instead, we split the sequence into two subsequences. On one of these subsequences, we can guarantee a decrease in function values, while for the other sequence we cannot. However, we prove that the latter sequence can accumulate only at stationary points of $J$. The main idea is then that $(u_n)_{n \in \mathbb{N}}$ can have only one accumulation point, because “switching” between several points conflicts with the decrease in function values that must happen for steps in between.

**Proof (of Theorem 3.1)** By Lemma 2.1 we have

$$J(u_{n+1}) - J(u_n) \leq \nabla J(u_n) \top (u_{n+1} - u_n) + \frac{\tau}{2} \|u_{n+1} - u_n\|^2$$

$$= \hat{G}_n \top (u_{n+1} - u_n) + \frac{L}{2} \|u_{n+1} - u_n\|^2 + \left( \nabla J(u_n) - \hat{G}_n \right) \top (u_{n+1} - u_n).$$

Utilizing Lemma 2.2 and the Cauchy-Schwarz inequality, we now obtain

$$J(u_{n+1}) - J(u_n) \leq \left( \frac{L}{2} - \frac{1}{\tau} \right) \|u_{n+1} - u_n\|^2 + \left\| \nabla J(u_n) - \hat{G}_n \right\| \cdot \|u_{n+1} - u_n\|$$

$$= \left( \frac{L}{2} - \frac{1}{\tau} \right) \|u_{n+1} - u_n\|^2 + \left\| \nabla J(u_n) - \hat{G}_n \right\| \cdot \|u_{n+1} - u_n\|. \quad (3.1)$$

Since $\frac{L}{2} - \frac{1}{\tau} < 0$, our idea is the following:

Steps that satisfy

$$\left\| \nabla J(u_n) - \hat{G}_n \right\| \leq \frac{1}{2} \left( \frac{L}{2} - \frac{1}{\tau} \right) \|u_{n+1} - u_n\|,$$  \quad (3.2)

i.e., steps with small errors in the gradient approximation, will yield decreasing function values.

On the other hand, the remaining steps will satisfy $\|u_{n+1} - u_n\| \to 0$, due to $\| \nabla J(u_n) - \hat{G}_n \| \to 0$ (see Lemma 2.6). With this in mind, we distinguish three cases: In Case 1, (3.2) is satisfied for almost all steps, while in Case 2 it is satisfied for only finitely many steps. In the last case, there are infinitely many steps satisfying and infinitely many steps violating (3.2).
Case 1: There exists $N \in \mathbb{N}$ such that
\[
\|\nabla J(u_n) - \hat{G}_n\| \leq \frac{1}{2} \left( \frac{1}{2} - \frac{L}{L} \right) \|u_{n+1} - u_n\| \quad \text{for all } n \geq N.
\]
In this case, it follows from \[\text{(3.1)}\] that $J(u_{n+1}) \leq J(u_n)$ for all $n \geq N$. Therefore, the sequence $(J(u_n))_{n \in \mathbb{N}}$ is monotonically decreasing for almost every $n \in \mathbb{N}$. Since $J$ is continuous and $\mathcal{U}$ is compact, $J$ is bounded and we therefore have $J(u_n) \to J \in \mathbb{R}$. Thus, it holds
\[
-\infty < \bar{J} - J(u_N) = \sum_{n=N}^{\infty} (J(u_{n+1}) - J(u_n)) \leq \frac{1}{2} \left( \frac{1}{2} - \frac{L}{L} \right) \sum_{n=N}^{\infty} \|u_{n+1} - u_n\|^2.
\]
Since $\frac{1}{2} \left( \frac{1}{2} - \frac{L}{L} \right) < 0$, we must have $\|u_{n+1} - u_n\| \to 0$. Let $(u_{nk})_{k \in \mathbb{N}}$ be a convergent subsequence with $u_{nk} \to \pi \in \mathcal{U}$.

By Lemma 2.6 we have $\hat{G}_{nk} \to \nabla J(\pi)$ and thus
\[
0 = \lim_{k \to \infty} \|u_{nk+1} - u_{nk}\| = \lim_{k \to \infty} \|\mathcal{P}_d(u_{nk} - \tau \nabla \hat{G}_{nk}) - u_{nk}\|
\]
\[
= \|\mathcal{P}_d(\pi - \tau \nabla J(\pi)) - \pi\|,
\]
i.e., every accumulation point of $(u_{nk})_{n \in \mathbb{N}}$ is stationary. Since $J$ has only finitely many stationary points, Lemma 2.3 yields the convergence of $(u_n)_{n \in \mathbb{N}}$ to a stationary point of $J$.

Case 2: There exists $N \in \mathbb{N}$ such that
\[
\|\nabla J(u_n) - \hat{G}_n\| > \frac{1}{2} \left( \frac{1}{2} - \frac{L}{L} \right) \|u_{n+1} - u_n\| \quad \text{for all } n \geq N.
\]

By Lemma 2.6 we have $\|\nabla J(u_n) - \hat{G}_n\| \to 0$. Since $\frac{1}{2} \left( \frac{1}{2} - \frac{L}{L} \right) > 0$, the above inequality directly implies $\|u_{n+1} - u_n\| \to 0$. Analogously to Case 1, we conclude that $(u_n)_{n \in \mathbb{N}}$ converges to a stationary point of $J$.

Case 3: There are infinitely many $n \in \mathbb{N}$ with
\[
\|\nabla J(u_n) - \hat{G}_n\| \leq \frac{1}{2} \left( \frac{1}{2} - \frac{L}{L} \right) \|u_{n+1} - u_n\|
\]
and infinitely many $n \in \mathbb{N}$ with
\[
\|\nabla J(u_n) - \hat{G}_n\| > \frac{1}{2} \left( \frac{1}{2} - \frac{L}{L} \right) \|u_{n+1} - u_n\|.
\]

In this case, we split $(u_n)_{n \in \mathbb{N}}$ disjointly in the two sequences $(u_{a(n)})_{n \in \mathbb{N}}$ and $(u_{b(n)})_{n \in \mathbb{N}}$, such that we have
\[
\|\nabla J(u_{a(n)}) - \hat{G}_{a(n)}\| \leq \frac{1}{2} \left( \frac{1}{2} - \frac{L}{L} \right) \|u_{a(n)+1} - u_{a(n)}\| \quad \text{for all } n \in \mathbb{N}
\]
and
\[
\|\nabla J(u_{b(n)}) - \hat{G}_{b(n)}\| > \frac{1}{2} \left( \frac{1}{2} - \frac{L}{L} \right) \|u_{b(n)+1} - u_{b(n)}\| \quad \text{for all } n \in \mathbb{N}.
\]

We call $(u_{a(n)})_{n \in \mathbb{N}}$ the sequence of descent steps. For $(u_{b(n)})_{n \in \mathbb{N}}$, observe that, as in Case 2, we directly obtain
\[
\|u_{b(n)+1} - u_{b(n)}\| \to 0 \quad \text{(3.3)}
\]
On step sizes rules for the continuous stochastic gradient scheme

\[ \min_{i \in \{1, \ldots, K\}} \| u_{b(n)} - \overline{u}_i \| < \varepsilon \quad \text{for all } n \geq N, \quad (3.4) \]

where \( \overline{u}_1, \ldots, \overline{u}_K \) denote the \( K \in \mathbb{N} \) accumulation points of \( (u_{b(n)})_{n \in \mathbb{N}} \).

Now, we prove by contradiction that \( J(\overline{u}_1) = J(\overline{u}_2) = \ldots = J(\overline{u}_K) \).
Suppose that this is not the case. Then we have at least \( M \geq 2 \) function values of accumulation points and
\[ F := \{ J(u) : u = \overline{u}_1, \ldots, \overline{u}_K \} = \{ f_1, f_2, \ldots, f_M \} \]
for some \( f_1 > f_2 > \ldots > f_M \in \mathbb{R} \). Now, choose \( \varepsilon > 0 \) small enough, such that
\[ 2\varepsilon < \min_{i,j \in \{1, \ldots, K\}} \| \overline{u}_i - \overline{u}_j \| \quad \text{and} \quad c_\varepsilon \varepsilon < f_1 - f_2, \]
where \( c_\varepsilon \) denotes the Lipschitz constant of \( J \).
By \( (3.3) \) and \( (3.4) \), there exists \( N \in \mathbb{N} \) such that for all \( n \geq N \) we have
\[ \| u_{b(n)+1} - u_{b(n)} \| < \frac{\varepsilon}{4} \quad \text{and} \quad \min_{i \in \{1, \ldots, K\}} \| u_{b(n)} - \overline{u}_i \| < \frac{\varepsilon}{4}. \quad (3.5) \]
Therefore, for \( n \geq N \) and \( i \in \{1, \ldots, K\} \), we have
\[ u_{b(n)} \in B_4^\varepsilon(\overline{u}_i) \implies u_{b(n)+1} \in B_4^\varepsilon(\overline{u}_i) \implies u_{b(n)+1} \notin B_4^\varepsilon(\overline{u}_j) \quad \text{for all } j \neq i. \quad (3.6) \]
(3.7)
Especially, for all \( n \geq N \) it holds
\[ u_{b(n)} \in B_4^\varepsilon(\overline{u}_i) \implies J(u_{b(n)+1}) \leq J(\overline{u}_i) + \frac{c_\varepsilon \varepsilon}{2} \quad \text{for all } i = 1, \ldots, K. \quad (3.8) \]

It follows from \( (3.5) \) and \( (3.7) \) that for \( n \geq N + 1 \):
(A) If \( u_{b(n)} \in B_4^\varepsilon(\overline{u}_i) \) and \( u_{b(n+1)} \in B_4^\varepsilon(\overline{u}_j) \) for some \( j \neq i \), then there must be at least one descent step between \( u_{b(n)} \) and \( u_{b(n+1)} \).
(B) If \( u_{b(n)} \in B_4^\varepsilon(\overline{u}_i) \) and \( u_{b(n)-1} \notin B_4^\varepsilon(\overline{u}_i) \), then \( u_{b(n)-1} \) must be a descent step.

Observe that (A) follows directly from \( (3.5) \) and \( (3.7) \), as moving from the vicinity of \( \overline{u}_i \) to a neighborhood of \( \overline{u}_j \) requires that there is an intermediate step \( u_n \) with \( \min_{i \in \{1, \ldots, K\}} \| u_{b(n)} - \overline{u}_i \| \geq \frac{\varepsilon}{4} \). Similarly, (B) is just the second condition in \( (3.7) \) reformulated.

Now, let \( i \in \{1, \ldots, K\} \) be chosen such that \( J(\overline{u}_i) \leq f_2 \) and let \( n_0 \geq N \) be chosen such that \( u_{b(n_0)} \in B_4^\varepsilon(\overline{u}_i) \) and \( u_{b(n_0)+1} \notin B_4^\varepsilon(\overline{u}_i) \). Using \( (3.6) \) and \( (3.8) \), we obtain
\[ J(u_{b(n_0)+1}) \leq J(\overline{u}_i) + \frac{c_\varepsilon \varepsilon}{2} \leq f_2 + \frac{c_\varepsilon \varepsilon}{2} < f_1 - \frac{c_\varepsilon \varepsilon}{2} \]
\[ < J(u) \quad \text{for all } u \in B_4^\varepsilon \left( J^{-1}(\{f_1\}) \cap \{\overline{u}_1, \ldots, \overline{u}_K\} \right). \]
Therefore, descent steps can never reach $B_{\frac{\varepsilon}{2}}(J^{-1}(\{f_1\} \cap \{\varpi_1, \ldots, \varpi_K\}))$ again! It follows from item (B), that $u_n \notin B_{\frac{\varepsilon}{2}}(J^{-1}(\{f_1\} \cap \{\varpi_1, \ldots, \varpi_K\})$ for all $n \geq b(n_0) + 1$, in contradiction to $J^{-1}(\{f_1\} \cap \{\varpi_1, \ldots, \varpi_K\})$ consisting of at least one accumulation point of $(u_n)_{n \in \mathbb{N}}$. Hence, we have

$$J(\varpi_1) = \ldots = J(\varpi_K) =: J.$$  

(3.9)

Next, we show that every accumulation point of $(u_{a(n)})_{n \in \mathbb{N}}$ is stationary. We prove this by contradiction. Assume there exists a non-stationary accumulation point $\overline{\varpi}$ of $(u_{a(n)})_{n \in \mathbb{N}}$. Observe that

$$\min_{i \in \{1, \ldots, K\}} \|\varpi - \varpi_i\| > 0.$$

Case 3.1: $J(\varpi) < J$.

Then, by the same arguments as above, there exists $N \in \mathbb{N}$ and $\varepsilon > 0$ s.t.

$$u_n \notin \bigcup_{i=1}^{K} B_{\frac{\varepsilon}{2}}(\varpi_i) \quad \text{for all } n \geq N.$$

This is a contradiction to $\varpi_1, \ldots, \varpi_K$ being accumulation points of $(u_n)_{n \in \mathbb{N}}$.

Case 3.2: $J(\overline{\varpi}) > J$.

In this case, there exists $N \in \mathbb{N}$ and $\varepsilon > 0$ such that $u_n \notin B_{\frac{\varepsilon}{2}}(\overline{\varpi})$ for all $n \geq N$. This is a contradiction to $\overline{\varpi}$ being an accumulation point of $(u_n)_{n \in \mathbb{N}}$.

Case 3.3: $J(\overline{\varpi}) = J$.

Since $\overline{\varpi}$ is an accumulation point of $(u_{a(n)})_{n \in \mathbb{N}}$, there exists a subsequence $(u_{a(n_k)})_{k \in \mathbb{N}}$ with $u_{a(n_k)} \to \overline{\varpi}$. The sequence $(u_{a(n_k)}-1)_{k \in \mathbb{N}}$ is bounded and therefore has at least one accumulation point $\varpi_{-1}$ and a subsequence $(u_{a(n_k)}-1)_{k \in \mathbb{N}}$ with $u_{a(n_k)}-1 \to \varpi_{-1}$. It follows that

$$P_{U} (\varpi_{-1} - \tau \nabla J(\varpi_{-1})) = \lim_{t \to \infty} P_U \left( u_{a(n_k)} -1 - \tau \hat{G}_{a(n_k)}-1 \right)$$

$$= \lim_{t \to \infty} u_{a(n_k)}$$

$$= \overline{\varpi}.$$

As $\overline{\varpi}$ is not stationary by our assumption, $\varpi_{-1} \neq \overline{\varpi}$ and $\varpi_{-1}$ is no stationary point of $J$. Thus, Lemma 2.1 combined with Lemma 2.2 yields

$$J(\overline{\varpi}) - J(\varpi_{-1}) \leq \left( \frac{\varepsilon}{2} - \frac{1}{2} \right) \|\varpi_{-1} - \overline{\varpi}\|^2 < 0.$$

Therefore, $\varpi_{-1}$ is an accumulation point of $(u_{a(n)})_{n \in \mathbb{N}}$, which satisfies $J(\varpi_{-1}) > J(\overline{\varpi}) = J$. This, however, is impossible, as seen in Case 3.2.

In conclusion, in Case 3, all accumulation points of $(u_n)_{n \in \mathbb{N}}$ are stationary. Thus, on every convergent subsequence we have $\|u_{n_k + 1} - u_{n_k}\| \to 0$. Since $(u_n)_{n \in \mathbb{N}}$ is bounded, this already implies $\|u_{n+1} - u_n\| \to 0$. Now, Lemma 2.3 yields the claimed convergence of $(u_n)_{n \in \mathbb{N}}$ to a stationary point of $J$. 

The idea of the proof above still applies in the case that $J$ is constant on some parts of $U$, i.e., $J$ can have infinitely many stationary points. We obtain the following convergence result:
Theorem 3.2 Let \( \mathcal{S}(J) \) be the set of stationary points of \( J \) on \( U \) as defined in Definition 2.2. Assume that the set

\[
\mathcal{N} := \{ J(u) : u \in \mathcal{S}(J) \} \subset \mathbb{R}
\]

is of Lebesgue-measure zero. Then every accumulation point of the sequence generated by CSG with constant stepsize \( \tau < \frac{2}{L} \) is stationary and we have convergence in function values.

Remark 3.1 Comparing Theorem 3.1 and Theorem 3.2, observe that under the weaker assumptions on the set of stationary points of \( J \), we no longer obtain convergence for the whole sequence of iterates. To illustrate why that is the case, consider the function \( J : \mathbb{R}^d \to \mathbb{R} \) given by

\[
J(u) = \cos(\pi \|u\|_2)
\]

and \( U = \{ u \in \mathbb{R}^d : \|u\|_2 \leq \frac{3}{2} \} \). Then, \( \mathcal{S}(J) = \{ 0 \} \cup \{ u \in U : \|u\| = 1 \} \), i.e., every point on the unit sphere is stationary. Thus, we can not use Lemma 2.3 at the end of the proof to obtain convergence of \( (u_n)_{n \in \mathbb{N}} \). Theoretically, it might happen that the iterates \( (u_n)_{n \in \mathbb{N}} \) cycle around the unit sphere, producing infinitely many accumulation points, all of which have the same objective function value. This, however, did not occur when testing this example numerically.

Remark 3.2 While the assumption in Theorem 3.2 seems unhandy at first, there is actually a rich theory concerning such properties. For example, Sard’s Theorem [24] and generalizations [25] give that the assumption holds if \( J \in C^d \) and \( U \) has smooth boundary. Even though it can be shown that there exist functions, which do not satisfy the assumption (e.g. [26, 27]), such counter-examples need to be precisely constructed and will most likely not appear in any application.

Proof (of Theorem 3.2) Proceeding analog to the proof of Theorem 3.1 we only have to adapt two intermediate results in Case 3:

(R1) The objective function values of all accumulation points of \( (u_{b(n)})_{n \in \mathbb{N}} \) are equal.

(R2) Every accumulation point of \( (u_{a(n)})_{n \in \mathbb{N}} \) is stationary.

Assume first, that (R1) does not hold. Then there exist two stationary points \( \pi_1 \neq \pi_2 \) with \( J(\pi_1) < J(\pi_2) \). Now, (A) and (B) shown in the proof of Theorem 3.1 yield that there must exist an accumulation point \( \pi_3 \) of \( (u_{b(n)})_{n \in \mathbb{N}} \), i.e., a stationary point, with \( J(\pi_1) < J(\pi_3) < J(\pi_2) \). Iterating this procedure, we conclude that the set \( \mathcal{N} \cap [J(\pi_1), J(\pi_2)] \) is dense in \( [J(\pi_1), J(\pi_2)] \).

By continuity of \( u \mapsto P_U(u - \tau \nabla J(u)) - u \) and compactness of \( U \), we see that

\[
\mathcal{N} \cap [J(\pi_1), J(\pi_2)] = [J(\pi_1), J(\pi_2)],
\]

contradicting our assumption that \( \lambda(\mathcal{N}) = 0 \).

For (R2), assume that \( (u_{a(n)})_{n \in \mathbb{N}} \) has a non-stationary accumulation point \( \pi \). Since \( \mathcal{S}(J) \) is compact, it holds

\[
\text{dist}(\{\pi\}, \mathcal{S}(J)) > 0.
\]

Thus, by the same arguments as in Case 3.1, 3.2 and 3.3 within the proof of Theorem 3.1 we observe that such a point \( \pi \) can not exist. \( \square \)
4 Backtracking

For this section, we again assume that Assumptions 2.1 to 2.3 are satisfied.

As \( \|\hat{G}_n - \nabla J(u_n)\| \to 0 \) and \( \|\hat{J}_n - J(u_n)\| \to 0 \) for \( n \to \infty \), we can use these approximations to refine the steplength by a backtracking line search method.

**Definition 4.1** For simplicity, we define

\[ s_n(t) := \mathcal{P}_U \left( u_n - t \hat{G}_n \right). \]

Furthermore, given \( n \) gradient samples \( \nabla_1 j(u_i, x_i) \) and \( n \) cost function samples \( j(u_i, x_i) \), by calculating the weights \( \alpha_i^{(n)}(u) \) w.r.t. a given point \( u \in U \), we define

\[ \tilde{J}_n(u) = \sum_{i=1}^{n} \alpha_i^{(n)}(u) j(u_i, x_i) \quad \text{and} \quad \tilde{G}_n(u) = \sum_{i=1}^{n} \alpha_i^{(n)}(u) \nabla_1 j(u_i, x_i), \]

which are approximations to \( J(u) \) and \( \nabla J(u) \) respectively.

Based on the well known Armijo-Wolfe conditions from continuous optimization \[28, 30\], we introduce the following step size conditions:

**Definition 4.2** For \( 0 < c_1 < c_2 < 1 \), we call \( s_n(\tau_n) \) an Armijo step, if

\[ \tilde{J}_n(s_n(\tau_n)) \leq \tilde{J}_n - c_1 \tilde{G}_n^\top (u_n - s_n(\tau_n)). \] (SW1)

Additionally, we define the following Wolfe-type condition:

\[ \tilde{G}_n(s_n(\tau_n))^\top (s_n(\tau_n) - u_n) \geq c_2 \tilde{G}_n^\top (s_n(\tau_n) - u_n). \] (SW2)

We try to obtain a stepsizethat satisfies (SW1) and (SW2) by a bisection approach, as formulated in Algorithm 2. Since we cannot guarantee to find a suitable stepsizewhile inside \( U \), we perform only a fixed number \( T \in \mathbb{N} \) of backtracking steps.

For our convergence analysis, we assume that in each iteration of CSG with line search, Algorithm 2 is initiated with the same \( \eta_n = \eta > 0 \). From a practical point of view, we might also consider a diminishing sequence \( (\eta_n)_{n \in \mathbb{N}} \) of backtracking initializations (see Section 6). The CSG method with backtracking line search (bCSG) is given in Algorithm 3.

Since all of the terms \( \tilde{J}_n(s_n(\tau_n)), \tilde{J}_n \) and \( \tilde{G}_n \) appearing in (SW1) contain some approximation error when compared to \( J(s_n(\tau_n)), J(u_n) \) and \( \nabla J(u_n) \) respectively, especially the first iterations of Algorithm 3 might profit from a slightly weaker formulation of (SW1). Therefore, in practice, we will replace (SW1) by the non-monotone version

\[ \tilde{J}_n(s_n(\tau_n)) \leq \max_{k \in \{0, \ldots, K\}} \tilde{J}_{n-k} - c_1 \tilde{G}_n^\top (u_n - s_n(\tau_n)), \] (SW1*)

for some \( K \in \{0, \ldots, n\} \) (e.g., we choose \( K = 1 \) throughout Section 6).
Algorithm 2 Backtracking refinement

1: Given $T \in \mathbb{N}$, $0 < \epsilon_1 < \epsilon_2 < 1$ appearing in $SW1$ and $SW2$, $u_n \in U$, and $\eta > 0$.
set $t = 1$, $a = 0$, $b = \infty$, $\eta_A = \infty$.
2: while $t \leq T$ do
3: Calculate step $s = P_U(u_n - \eta \hat{G}_n)$, weights $a_0^{-1}(s)$ and the approximations $\hat{J}(s)$, $\hat{G}(s)$;
4: if $SW1$ is not satisfied then
5: $b = \eta$;
6: else if $s = u_n - \eta \hat{G}_\eta$ and $SW2$ is not satisfied then
7: $a = \eta$; $\eta_A = \eta$;
8: else
9: break;
10: end if
11: if $b < \infty$ then
12: $\eta = \frac{a + b}{2}$;
13: else
14: $\eta = 2a$;
15: end if
16: $t = t + 1$;
17: end while
18: if $t = T + 1$ and $\eta_A < \infty$ then
19: $\tau_n = \eta_A$;
20: else
21: $\tau_n = \eta$;
22: end if

4.1 Convergence Results

For CSG with backtracking line search, we obtain the same convergence results as for constant step sizes:

Theorem 4.1 (Convergence for backtracking line search) Let $S(J)$ be the set of stationary points of $J$ on $U$ as defined in Definition 2.2. Assume that

$$\mathcal{N} := \{ J(u) : u \in S(J) \} \subset \mathbb{R}$$

is of Lebesgue-measure zero and $T$ in Algorithm 3 is chosen large enough, such that $2^{-T} \eta < \frac{1}{2}$. Then every accumulation point of the sequence $(u_n)_{n \in \mathbb{N}}$ generated by Algorithm 3 is stationary and we have convergence in function values.

If $J$ satisfies the stronger assumption of having only finitely many stationary points, $(u_n)_{n \in \mathbb{N}}$ converges to a stationary point of $J$.

Proof Notice first, that there are only two possible outcomes of Algorithm 2: Either $\tau_n$ satisfies $SW1$, or $\tau_n = 2^{-T} \eta < \frac{1}{2}$. Furthermore, as we have seen in the proof of Lemma 2.3, for all $\epsilon > 0$ almost all $u_n$ lie in $\epsilon$-Balls around the accumulation points of $(u_n)_{n \in \mathbb{N}}$, since $(u_n)_{n \in \mathbb{N}}$ is bounded. Therefore, $\hat{J}_n(u_{n+1}) - J(u_{n+1}) \to 0$ and $\hat{G}_n(u_{n+1}) - \nabla J(u_{n+1}) \to 0$ (compare Lemma 2.6). Since we already know,
that the steps with constant stepsize $\tau_n = 2^{-T}\eta < \frac{\varepsilon}{L}$ can be split in descent steps and steps which satisfy $\|u_{n+1} - u_n\| \to 0$, we now take a closer look at the Armijo-steps, i.e., steps with $\tau_n \neq 2^{-T}\eta$.

If $\tau_n \neq 2^{-T}\eta$, by [SW1] and Lemma 2.2 it holds

$$J(u_{n+1}) - J(u_n) \leq -c_1 \frac{\|u_{n+1} - u_n\|^2}{\tau_n^2} + |J(u_n) - J(u_{n+1})|.$$

Therefore, we either have

$$|J(u_n) - J(u_{n+1})| \leq \frac{c_1 \|u_{n+1} - u_n\|^2}{\tau_{\max}^2},$$

in which case it holds $J(u_{n+1}) \leq J(u_n)$, or

$$|J(u_n) - J(u_{n+1})| > \frac{c_1 \|u_{n+1} - u_n\|^2}{\tau_{\max}^2},$$

in which case $\tilde{J}_n(u_{n+1}) - J(u_{n+1}) \to 0$ and $\tilde{J}_n - J(u_n) \to 0$ yield $\|u_{n+1} - u_n\| \to 0$.

Thus, regardless of whether or not $\tau_n = 2^{-T}\eta$, we can split $(u_n)_{n \in \mathbb{N}}$ in a subsequence of descent steps and a subsequence of steps with $\|u_{n+1} - u_n\| \to 0$.

The rest of the proof is now identical to the proof of Theorem 3.1 and Theorem 3.2.

\[\Box\]

5 Estimations for the Lipschitz constant of $\nabla J$

We have already seen, that the Lipschitz constant $L$ of $\nabla J$ is closely connected with efficient bounds on the step sizes. However, in general, we can not expect to have any knowledge of $L$ a priori. Thus, we are interested in an approximation of $L$, that can be calculated during the optimization process.

Investigating the proof of Lemma 2.1 in [21]

$$J(u_1) = J(u_2) + \int_0^1 \langle \nabla J(u_2) + t(u_1 - u_2), u_1 - u_2 \rangle \, dt$$

$$= J(u_2) + \langle \nabla J(u_2), u_1 - u_2 \rangle$$

$$+ \int_0^1 \langle \nabla J(u_2 + t(u_1 - u_2)) - \nabla J(u_2), u_1 - u_2 \rangle \, dt$$

$$\leq J(u_2) + \langle \nabla J(u_2), u_1 - u_2 \rangle$$

$$+ \int_0^1 \|\nabla J(u_2 + t(u_1 - u_2)) - \nabla J(u_2)\| \cdot \|u_1 - u_2\| \, dt$$

$$\leq J(u_2) + \langle \nabla J(u_2), u_1 - u_2 \rangle + \int_0^1 L \|u_1 - u_2\|^2 \, dt$$

$$= J(u_2) + \langle \nabla J(u_2), u_1 - u_2 \rangle + \frac{L}{2} \|u_1 - u_2\|^2,$$
we observe that we do not need the true Lipschitz constant $L$ of $\nabla J$ for the second inequality. Instead, it is sufficient to choose any constant $C = C(u_1, u_2)$ that satisfies

$$\|\nabla J(u_2 + t(u_1 - u_2)) - \nabla J(u_2)\| \leq C\|u_1 - u_2\| \quad \text{for all } t \in [0, 1].$$

To motivate our approach, assume that $J$ is twice continuously differentiable. In this case, a possible approximation to the constant $C_n$ in iteration $n$ is $\|\nabla^2 J(u_n)\|$. Therefore, utilizing the previous gradient approximations, we obtain

$$C_n \approx \frac{\|\nabla^2 J(u_n)\|}{\|u_n - u_{n-1}\|} \approx \frac{\|\hat{G}_n - \hat{G}_{n-1}\|}{\|u_n - u_{n-1}\||},$$

Then, $C_n^{-1}$ yields a good initial step size for our backtracking line search. To circumvent high oscillation of $C_n$, which may arise from the approximation errors of the terms involved, we project $C_n$ onto the interval $[C_{\min}, C_{\max}] \subset \mathbb{R}$, where $0 < C_{\min} < C_{\max} < \frac{T + 1}{T}L$, i.e.,

$$C_n = \min \left\{ C_{\max}, \max \left\{ C_{\min}, \frac{\|\hat{G}_n - \hat{G}_{n-1}\|}{\|u_n - u_{n-1}\|} \right\} \right\}. \quad (5.1)$$

If possible, $C_{\min}$ and $C_{\max}$ should be chosen according to information concerning $L$. However, tight bounds on these quantities are not needed, as long as $T$ is chosen large enough. The resulting SCIBL-CSG (SCaling Independent Backtracking Line search) method is presented in Algorithm 4. Notice that SCIBL-CSG does not require any a priori choice of step sizes and yields the same convergence results as bCSG.

**Algorithm 4 SCIBL-CSG**

1: Given $u_0 \in \mathcal{U}$,
2: while Termination condition not met do
3: \hspace{1em} Sample objective function: $j_n := j(u_n, x_n)$
4: \hspace{1em} Sample gradient: $g_n := \nabla j(u_n, x_n)$
5: \hspace{1em} Calculate weights $\alpha_k$
6: \hspace{1em} Calculate search direction: $\hat{G}_n := \sum_{k=1}^{n} \alpha_k g_k$
7: \hspace{1em} Compute objective function value approximation: $\hat{J}_n := \sum_{k=1}^{n} \alpha_k j_k$
8: \hspace{1em} Calculate $C_n$ by (5.1)
9: \hspace{1em} Calculate stepsize $\tau_n$ by Algorithm 2 with start at $\frac{1}{C_n}$
10: Gradient step: $u_{n+1} := P_\mathcal{U}(u_n - \tau_n \hat{G}_n)$
11: Update index: $n = n + 1$
12: end while

In the following section, we now will illustrate the presented analytical results based on numerical experiments with several academic examples.
6 Numerical Experiments

The calculation of the integration weights $\alpha_k$ (see Algorithm 3) plays a crucial role for the performance of any CSG algorithm. In [9], several methods for the weight computation have been analyzed and compared. Since the focus of this contribution lies entirely on the step sizes, for all numerical results concerning CSG, we choose the so called empirical weights, which are computationally very cheap, but, given a desired precision, typically lead to a larger number of required gradient evaluations.

In the first example, the convergence properties for CSG with constant step sizes are analyzed in a prototype setting. Furthermore, we observe that – as theory predicts – the basic SG method does not yield a convergent sequence of iterates in this example, if carried out with the same step sizes.

The second example aims to demonstrate the stability of CSG with backtracking (bCSG) with respect to the initially chosen step size ($\eta_n$) $\in \mathbb{N}$. For this, we compare the performance of bCSG and the well-known AdaGrad [31] method when carried out for a large spectrum of choices for ($\eta_n$) $\in \mathbb{N}$.

In the final example, we consider a Rosenbrock-type function with multiplicative noise and compare the performance of AdaGrad, bCSG and SCIBL-CSG.

6.1 Convergence for constant step sizes

Define $\mathcal{U} = [-\frac{1}{2}, \frac{1}{2}]$, $\mathcal{X} = (-\frac{1}{2}, \frac{1}{2})$ and consider the problem

$$
\min_{u \in \mathcal{U}} \frac{1}{2} \int_{\mathcal{X}} (u - x)^2 \, dx.
$$

(6.1)

It is easy to see that (6.1) has a unique solution $u^* = 0$. Furthermore, the objective function is $L$-smooth (with Lipschitz constant 1) and even strictly convex. Thus, by Theorem 3.1, the CSG method with a constant step size $\tau < 2$ produces a sequence $u_n$ $\in \mathbb{N}$ that satisfies $u_n \to 0$.

However, even in this highly regular setting, the commonly used basic SG method does not guarantee convergence of the iterates for a constant step size.

To demonstrate this behavior of both CSG and SG, we draw 2000 random starting points $u_0 \in \mathcal{U}$ and compare the iterates produced by CSG and SG with five different constant step sizes ($\tau \in \{0.01, 0.1, 1, 1.9, 1.99\}$). The results are shown in Figure 6.1.

As expected, the iterates produced by the SG method do not converge to the optimal solution, but instead remain in a neighborhood of $u^*$. The radius of said neighborhood depends on the choice of $\tau$ and decreases for smaller $\tau$, see [12, Theorem 4.6].

Convergence for CSG in this example has already been observed numerically in [9, Section 5.1], but has not been proven until now.

6.2 Stability with respect to the step size

As a basic example, we set $\mathcal{U} = [-10, 10]^5$, $\mathcal{X} = (-1, 1)^5$ and consider the Problem

$$
\min_{u \in \mathcal{U}} J(u),
$$

(6.2)
Fig. 6.1 Comparison of the iterates produced by 500 steps of SG (red / first row) and CSG (green / second row) with 2000 random starting points \( u_0 \in U \). Both methods have been tested for the five different constant step sizes \( \tau \in \{0.01, 0.1, 1.1, 1.9, 1.99\} \) (first to fifth column). The shaded areas indicate the quantiles \( P_{0.1}, P_{0.9} \) (light) and \( P_{0.25}, P_{0.75} \) (dark), while the solid line represents the median of the 2000 runs.

where

\[
J(u) = -\int_X \frac{20}{1 + \|u - x\|^2} \, dx.
\]

Problem (6.2) has the unique solution \( u^* = 0 \in U \), which can be found analytically.

As a comparison to our method, we choose the AdaGrad \[31\] algorithm, as it is widely used for problems of type (1.1). Both AdaGrad and CSG start each iteration with a predecribed step size \( \eta_n > 0 \), based on which the calculation of the true step size \( \tau_n \) is performed (see Algorithm 2). We want to test the stability of both methods with respect to the initially chosen step length. For this purpose, we set \( \eta_n = \frac{\tau_0}{n} \), where \( \tau_0 > 0 \), \( n \) is the iteration count and \( d \in [0, 1] \) is fixed.

For each combination of \( \tau_0 \) and \( d \), we choose 1200 random starting points in \( U \) and perform 500 optimization steps with both AdaGrad and backtracking CSG. The median of the absolute error \( \|u_{500} - u^*\| \) after the optimization, depending on \( d \) and \( \tau_0 \), is presented in Figure 6.2.

While there are a few instances where AdaGrad yields a better result than backtracking CSG, we observe that the performance of AdaGrad changes rapidly, especially with respect to the parameter \( d \). The backtracking CSG method on the other hand performs superior in most cases and is much less dependent on the choice of parameters.

6.3 Comparison of SCIBL-CSG, bCSG and AdaGrad

As an academic example to test the performance of Algorithm 3 and Algorithm 4, we consider a Rosenbrock function \[32\] with multiplicative noise. Our optimization problem reads as follows:

\[
\min_{u \in [-3.3]^2} \int_X \left(1 + x\right) \left((1 - u_1)^2 + 100(u_2 - u_1^2)^2\right) \mu(dx).
\]  

(6.3)

Here, we assume the “noise” \( x \) to be standard normal distributed, i.e., \( x \sim \mathcal{N}(0, 1) \). Note that \( \mu \) and \( \mathcal{X} \) do not satisfy Assumption 2.2. However, as the density function
corresponding to $\mu$ decreases exponentially for $|x| \to \infty$, the approximation errors in $G_n$ and $J_n$ still vanish in the course of the iterations.

Problem (6.3) has only one stationary point, $u^* = (1, 1)^T$, which is the global minimum. While $u^*$ can easily be found analytically, optimization problems with Rosenbrock-type functions are notoriously difficult to solve for gradient based methods (see e.g. [33]).

For our test, we choose 5000 random starting points $u_0 \in U$ and compare the absolute error $\|u_n - u^*\|$ in each iteration for

1. AdaGrad with a step size of $\tau_n = \frac{1}{10 \sqrt{n}}$.
2. Backtracking CSG (bCSG) with $\eta_n = \frac{1}{40}$.
3. Scaling independent backtracking CSG (SCIBL-CSG) with $C_{\min} = 10^{-8}$ and $C_{\max} = 10^{8}$.

Both $\tau_n$ in 1. and $\eta_n$ in 2. were chosen as the top-performers from a small test group of candidates. The choices for $C_{\min}$ and $C_{\max}$ are somewhat arbitrary, since we stated in Section 5 that SCIBL-CSG would require almost no a priori analysis of (6.3).

The results are presented in Figure 6.3. Both bCSG and SCIBL-CSG yield comparable results, while AdaGrad achieves hardly any progress in later iterations.

Additionally, we want to know how reasonable the initial guess $C_0$ calculated in SCIBL-CSG is. For this purpose, we analyzed how many backtracking refinements, i.e., steps of Algorithm 2, both SCIBL-CSG and bCSG take in each iteration before accepting a step size. In total, the ordinary bCSG required roughly 1.5 times as many refinement steps when compared to SCIBL-CSG.
Fig. 6.3 Evolution of the absolute error $||u_n - u^*||$ during the optimization process for AdaGrad (red), Backtracking CSG (bCSG, blue) and Scaling independent backtracking CSG (SCIBL-CSG, yellow). The solid lines represent the median of the 5000 runs with different starting points, while the shaded areas correspond to the quantiles $P_{0.25, 0.75}$.

7 Conclusion and Outlook

In this contribution, we provided a convergence proof of the CSG method when carried out with a small enough constant step size for optimization problems of type (1.1). Additionally, it was shown that the CSG algorithm can be augmented by an Armijo-type backtracking line search based on the objective function and gradient approximations calculated in the course of the iterations. The resulting method was proven to converge under mild assumptions and was shown to yield stable results for a large spectrum of hyperparameters. Lastly, we combined a heuristic approach for approximating the Lipschitz constant of the gradient of the objective function with the backtracking CSG algorithm to obtain a method that requires no a priori step size rules and almost no information about the optimization problem.

For all three methods, we stated convergence results that are similar to convergence results for full gradient schemes, i.e., every accumulation of the sequence of iterates is stationary and we have convergence in the objective function values. Furthermore, as is the case for full gradient methods, if the optimization problem has only finitely many stationary points, the presented CSG variants produce a sequence that is guaranteed to converge to one of these stationary points.

Lastly, all results were numerically demonstrated by means of academic test cases and compared to similar optimization methods from literature.

While we were able to show convergence properties of CSG which are similar to results obtained for full gradient approaches, it remains an open question, if we can achieve similar convergence rates as well. Furthermore, the presented methods have so far only been tested for academic examples. Thus, we are interested in applying the CSG method to large scale optimization problems from e.g. topology optimization.

Acknowledgements The research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Project-ID 416229255 - CRC 1411.
References

1. Kim, C., Lee, J., Yoo, J.: Machine learning-combined topology optimization for functionary graded composite structure design. Comput. Methods Appl. Mech. Engrg. 387, Paper No. 114158, 32 (2021). DOI 10.1016/j.cma.2021.114158
2. Evstatiev, E.G., Finn, J.M., Shadwick, B.A., Hengartner, N.: Noise and error analysis and optimization in particle-based kinetic plasma simulations. J. Comput. Phys. 440, Paper No. 110394, 28 (2021). DOI 10.1016/j.jcp.2021.110394
3. Wadbro, E., Berggren, M.: Topology optimization of an acoustic horn. Comput. Methods Appl. Mech. Engrg. 196(1-3), 420–436 (2006). DOI 10.1016/j.cma.2006.05.005
4. Hassan, E., Wadbro, E., Berggren, M.: Topology optimization of metallic antennas. IEEE Trans. Antennas and Propagation 62(5), 2488–2500 (2014). DOI 10.1109/TAP.2014.2309112
5. Semmler, J., Pflug, L., Stingl, M., Leugering, G.: Shape optimization in electromagnetic applications. In: New Trends in Shape Optimization, pp. 251–269. Springer (2015)
6. Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Statistics 22, 400–407 (1951)
7. Schmidt, M., Le Roux, N., Bach, F.: Minimizing finite sums with the stochastic average gradient. Math. Program. 162(1-2, Ser. A), 83–112 (2017)
8. Pflug, L., Bernhardt, N., Grieshammer, M., Stingl, M.: CSG: a new stochastic gradient method for the efficient solution of structural optimization problems with infinitely many states. Struct. Multidiscip. Optim. 61(6), 2595–2611 (2020)
9. Pflug, L., Grieshammer, M., Uihlein, A., Stingl, M.: Csg: A stochastic gradient method for a wide class of optimization problems appearing in a machine learning or data-driven context (2021)
10. Patel, V.: Kalman-based stochastic gradient method with stop condition and insensitivity to conditioning. SIAM J. Optim. 26(4), 2620–2648 (2016)
11. Schmidt, M., Roux, N.L.: Fast convergence of stochastic gradient descent under a strong growth condition (2013)
12. Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. SIAM Rev. 60(2), 223–311 (2018). DOI 10.1137/16M1080173
13. Cevher, V., Vû, B.C.: On the linear convergence of the stochastic gradient method with constant step-size. Optim. Lett. 13(5), 1177–1187 (2019). DOI 10.1007/s11590-018-1331-1
14. Lacoste-Julien, S., Schmidt, M., Bach, F.: A simpler approach to obtaining an o(1/t) convergence rate for the projected stochastic subgradient method (2012)
15. Paquette, C., Scheinberg, K.: A stochastic line search method with expected complexity analysis. SIAM J. Optim. 30(1), 349–376 (2020)
16. Prazeres, M., Oberman, A.M.: Stochastic gradient descent with Polyak’s learning rate. J. Sci. Comput. 89(1), Paper No. 25, 16 (2021). DOI 10.1007/s10915-021-01628-3
17. Tan, C., Ma, S., Dai, Y.H., Qian, Y.: Barzilai-borwein step size for stochastic gradient descent. In: D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, R. Garnett (eds.) Advances in Neural Information Processing Systems 29, pp.
18. Klenke, A.: Probability theory. Universitext. Springer-Verlag London, Ltd., London (2008). A comprehensive course, Translated from the 2006 German original
19. Goldstein, A.A.: Convex programming in Hilbert space. Bull. Amer. Math. Soc. 70, 709–710 (1964). DOI 10.1090/S0002-9904-1964-11178-2
20. Levitin, E., Polyak, B.: Constrained minimization methods. USSR Computational Mathematics and Mathematical Physics 6(5), 1–50 (1966). DOI https://doi.org/10.1016/0041-5553(66)90114-5
21. Beck, A.: First-order methods in optimization, MOS-SIAM Series on Optimization, vol. 25. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA (2017). DOI 10.1137/1.9781611974997.ch1
22. Gibbs, A.L., Su, F.E.: On choosing and bounding probability metrics. International statistical review 70(3), 419–435 (2002)
23. Varadarajan, V.S.: On the convergence of sample probability distributions. Sankhyä 19, 23–26 (1958)
24. Sard, A.: The measure of the critical values of differentiable maps. Bull. Amer. Math. Soc. 48, 883–890 (1942). DOI 10.1090/S0002-9904-1942-07811-6
25. Guillemin, V., Pollack, A.: Differential topology. Prentice-Hall, Inc., Englewood Cliffs, N.J. (1974)
26. Whitney, H.: A function not constant on a connected set of critical points. Duke Mathematical Journal 1(4), 514 – 517 (1935). DOI 10.1215/S0012-7094-35-00138-7
27. Kaufman, R.: A singular map of a cube onto a square. J. Differential Geometry 14(4), 593–594 (1981) (1979)
28. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. Pacific J. Math. 16, 1–3 (1966)
29. Wolfe, P.: Convergence conditions for ascent methods. SIAM Rev. 11, 226–235 (1969). DOI 10.1137/1011036
30. Wolfe, P.: Convergence conditions for ascent methods. II. Some corrections. SIAM Rev. 13, 185–188 (1971). DOI 10.1137/1013035
31. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. 12, 2121–2159 (2011)
32. Rosenbrock, H.H.: An Automatic Method for Finding the Greatest or Least Value of a Function. The Computer Journal 3(3), 175–184 (1960). DOI 10.1093/comjnl/3.3.175
33. Emiola, I., Adem, R.: Comparison of minimization methods for rosenbrock functions (2021)