Grieshammer, Max; Pflug, Lukas; Stingl, Michael; Uihlein, Andrian

The continuous stochastic gradient method: part II–application and numerics

Computational Optimization and Applications

Provided in Cooperation with:
Springer Nature

Suggested Citation: Grieshammer, Max; Pflug, Lukas; Stingl, Michael; Uihlein, Andrian (2023) : The continuous stochastic gradient method: part II–application and numerics, Computational Optimization and Applications, ISSN 1573-2894, Springer US, New York, NY, Vol. 87, Iss. 3, pp. 977-1008,
https://doi.org/10.1007/s10589-023-00540-w

Computational Optimization and Applications (2024) 87:977–1008
https://doi.org/10.1007/s10589-023-00540-w

The continuous stochastic gradient method: part II–application and numerics

Max Grieshammer[1] · Lukas Pflug[1,2] · Michael Stingl[1] · Andrian Uihlein[1]

Received: 7 February 2023 / Accepted: 26 October 2023 / Published online: 24 November 2023

Abstract

In this contribution, we present a numerical analysis of the *continuous stochastic gradient* (CSG) method, including applications from topology optimization and convergence rates. In contrast to standard stochastic gradient optimization schemes, CSG does not discard old gradient samples from previous iterations. Instead, design dependent integration weights are calculated to form a convex combination as an approximation to the true gradient at the current design. As the approximation error vanishes in the course of the iterations, CSG represents a hybrid approach, starting off like a purely stochastic method and behaving like a full gradient scheme in the limit. In this work, the efficiency of CSG is demonstrated for practically relevant applications from topology optimization. These settings are characterized by both, a large number of optimization variables *and* an objective function, whose evaluation requires the numerical computation of multiple integrals concatenated in a nonlinear fashion. Such problems could not be solved by any existing optimization method before. Lastly, with regards to convergence rates, first estimates are provided and confirmed with the help of numerical experiments.

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

✉ Andrian Uihlein
andrian.uihlein@fau.de

Max Grieshammer
max.grieshammer@fau.de

Lukas Pflug
lukas.pflug@fau.de

Michael Stingl
michael.stingl@fau.de

[1] Department of Mathematics, Chair of Applied Mathematics, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany

[2] FAU Competence Center Scientific Computing, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany

Mathematics Subject Classification 65K05 · 90C06 · 90C15 · 90C30

1 Introduction

In this paper, we present a numerical analysis of the *Continuous Stochastic Gradient* (CSG) method, which was first proposed in [1]. Later, in [2], it was shown that the error in the CSG gradient and objective function approximation vanishes during the course of the iterations. This key property of CSG yields strong convergence results known from classic gradient methods, e.g., convergence of the sequence of iterates for constant step sizes, which are beyond the scope of standard stochastic approaches known from literature, like the *Stochastic Gradient* (SG) method [3], or the *Stochastic Average Gradient* (SAG) method [4].

Furthermore, the approximation property of CSG significantly increases the set of possible applications, allowing for more complex structures in the optimization problem than the schemes listed before. While CSG was shown to perform better than various stochastic optimization approaches on academic examples [2], it remains to see if this is also the case for more involved applications. For this purpose, we consider several optimization problems arising in the context of optimal nanoparticle design. These applications focus on optimization with respect to the resulting color of a particulate product, as it represents one of the most prominent fields of research within this setting [5–10].

Moreover, all convergence results stated in [2] provide no insight on the *rate* of convergence. Since this plays a crucial role for the practicability of CSG, it is of great importance to further analyze this quantity. In this contribution, we conjecture estimated convergence rates for the general CSG method and verify them numerically.

1.1 Structure of the paper

Section 2 introduces the application from nanoparticle optics, mentioned above. Two different methods to model the particle, varying greatly in computational effort and design dimension, are presented. After detailing the setting and challenges in the low-dimensional optimization problem, we compare the results of the CSG method to different approaches based on the *fmincon* algorithm provided by MATLAB (Sect. 2.7). Later on, we analyze the high-dimensional problem formulation purely within the CSG framework, since a comparison with generic deterministic optimization schemes is out of scope, due to the associated computational complexity.

Afterwards, Sect. 3 shortly covers techniques to estimate the gradient approximation error during the optimization, before we focus on the convergence rate of CSG in Sect. 4. While the expected rates stated therein are *not* proven, we present detailed numerical examples to solidify our claims. Furthermore, we analyze how the convergence rate depends on the dimension of integration and how to avoid slow convergence, if the objective function admits additional structure.

2 Nanoparticle design optimization

Since the design of a nanoparticle, i.e., its shape, size, material distribution, etc., heavily impacts its optical properties, the task of optimizing a nanoparticle design with respect to a specific optical property arises naturally [11]. In this section, we are interested in using hematite nanoparticles to optimize the color of a paint film [12]. Thus, we start by introducing our main framework for this application.

2.1 Color spaces

First off, we should explain what *optimal color* means in our setting. There are several different methods to describe color mathematically, e.g., assigning each color an RGB representation vector $\mathbf{v} \in \mathbb{R}^3$, where the three components of $\mathbf{v}$ correspond to the red, green and blue value of the color. In our application, we are interested in the color of the paint film as it appears to the human eye. Therefore, the underlying color space should be chosen based on the following property:

> *If the Euclidean distance between the representation vectors of two colors is small, the colors should be almost indistinguishable to the human eye.*

As it turns out, the RGB color space is a very poor choice with respect to this feature. Hence, we instead choose the CIELAB color space [13], which was introduced by the International Commission of Illumination (Commission Internationale de l'Eclairage, CIE), as it was designed with this exact purpose in mind. The CIELAB representation of a color consists of three values $\mathbf{L}$, $\mathbf{a}$ and $\mathbf{b}$. Here, $\mathbf{L}$ corresponds to the lightness of a color and ranges from 0 (black) to 100 (white). The values of $\mathbf{a}$ and $\mathbf{b}$, typically within the range of ± 150, describe the colors position with respect to the opponent color pairs green-red and blue-yellow. A short overview is given in Fig. 1.

Another color space, which naturally arises from our setting, is the CIE 1931 XYZ color space [14]. The values of X, Y and Z can be calculated by integrating the optical properties of a particle over the spectrum of visible light (400–700 nm), which we denote by Λ. Each of these integrations is weighted by the corresponding color matching functions $x, y, z : \Lambda \to \mathbb{R}$.

Thus, in our application, we will first calculate the CIE 1931 XYZ representation of the resulting color and then use the (nonlinear) color space transformation $\Psi : \mathbb{R}^3 \to \mathbb{R}^3$ with $\Psi(\mathrm{X},\mathrm{Y},\mathrm{Z}) = (\mathbf{L}, \mathbf{a}, \mathbf{b})^\top$, to work in the CIELAB color space. For this transformation, we define a reference white point

$$\begin{pmatrix} \mathrm{X}_r \\ \mathrm{Y}_r \\ \mathrm{Z}_r \end{pmatrix} = \begin{pmatrix} 94.72528492 \\ 100 \\ 107.13012997 \end{pmatrix}$$

and denote the relative XYZ values by

$$\tilde{\mathrm{X}} = \frac{\mathrm{X}}{\mathrm{X}_r}, \quad \tilde{\mathrm{Y}} = \frac{\mathrm{Y}}{\mathrm{Y}_r}, \quad \text{and} \quad \tilde{\mathrm{Z}} = \frac{\mathrm{Z}}{\mathrm{Z}_r}.$$

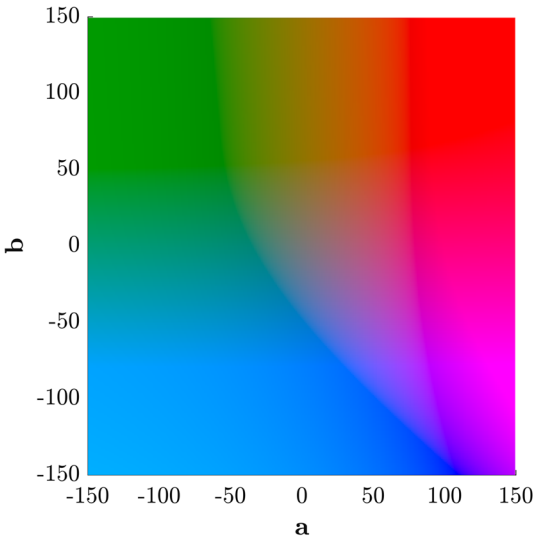

Fig. 1 Resulting color for various different values of **a** and **b**. Positive values of **a** result in red colors, while colors corresponding to negative values of **a** appear green. Similarly, positive **b** values yield yellow colors, while negative **b** values shift the color into the blue spectrum. In this figure, we fixed $\mathbf{L} = 50$

Utilizing the intended CIE parameters $\epsilon = \frac{216}{24389}$ and $\kappa = \frac{24389}{27}$, the LAB color values are then given by

$$\mathbf{L} = 116 f(\tilde{\mathrm{Y}}) - 16, \quad \mathbf{a} = 500\big(f(\tilde{\mathrm{X}}) - f(\tilde{\mathrm{Y}})\big) \quad \text{and} \quad \mathbf{b} = 200\big(f(\tilde{\mathrm{Y}}) - f(\tilde{\mathrm{Z}})\big),$$

where $f : \mathbb{R} \to \mathbb{R}$ is defined as

$$f(t) = \begin{cases} \sqrt[3]{t} & \text{if } t > \epsilon \\ \frac{\kappa t + 16}{116} & \text{otherwise} \end{cases}.$$

2.2 Mie theory and discrete dipole approximation

Given a nanoparticle shape and material, we can use the time-harmonic Maxwell's equations to calculate its optical properties. Specifically, in our setting, we are interested in the absorption (Abs), scattering (Sca) and geometry factor (Geo) [15, Section 2.8]. These properties describe the interactions of a particle with light and are therefore dependent not only on the particle's design, but also its orientation w.r.t. the incoming lightwave as well as the wavelength of said light. The time required and precision achieved in their numerical calculation are, of course, dependent on our model of the nanoparticle and the method used to solve Maxwell's equations. For our setting, we choose two different approaches.

On the one hand, we will use the discrete dipole approximation (DDA) [16–18], in which the particle is discretized into an equidistant grid of dipole cells. Thus, DDA allows the analysis of arbitrary particle shapes and material distributions. The downside lies within the computational complexity of the method, which scales with the total number of dipoles and therefore grows rapidly when increasing the resolution.

While the CSG method is still capable of solving the resulting optimization problem in our experiments, the tremendous computational cost associated to the DDA approach severely impede a detailed analysis of the problem. Especially, there is no computationally feasible, generic optimization scheme to compare our results with. However, we want to note that optimization in the DDA model has already been done in a slightly simpler setting, where the full integral over Λ was replaced by summation over a small number of different wavelengths [19].

On the other hand, Mie theory [20, 21] provides a numerically cheap alternative, at the price of a more restrictive setting. In Mie theory, one only considers radially symmetric particles. In this special setting, it is possible to find analytic solutions based on series expansions to the time-harmonic Maxwell's equations. Therefore, in our first approach, we will only consider core-shell particles, as the utilization of Mie theory allows for a much deeper analysis of the resulting optimization problem and comparison to deterministic optimization approaches, which rely on discretization of the integrals.

2.3 Nanoparticles in paint film—Kubelka–Munk theory

As mentioned above, the XYZ color values of the paint film can be calculated by integration of the corresponding color matching functions x, y, z and the important optical properties of the nanoparticle. The precise method to obtain X, Y and Z is given by the Kubelka–Munk theory [22], augmented by a Saunderson correction [23]. For a paint film, in which nanoparticles with design u are oriented in direction $\nu \in \mathbb{S}^2$, that is illuminated by light with wavelength $\lambda \in \Lambda$, the resulting color can be expressed by the K and S value

$$K(u,\lambda,\nu) = \mathrm{Abs}(u,\lambda,\nu) \quad \text{and} \quad S(u,\lambda,\nu) = \mathrm{Sca}(u,\lambda,\nu)\big(1-\mathrm{Geo}(u,\lambda,\nu)\big)$$

via the reflectance

$$R_\infty(u,\lambda,\nu) = 1 + \frac{8}{3}\frac{K(u,\lambda,\nu)}{S(u,\lambda,\nu)} - \sqrt{\left(\frac{8}{3}\frac{K(u,\lambda,\nu)}{S(u,\lambda,\nu)}\right)^2 + \frac{16}{3}\frac{K(u,\lambda,\nu)}{S(u,\lambda,\nu)}}\,.$$

Now, X, Y and Z can be obtained by

$$\begin{aligned}
\mathrm{X}(u,\nu) &= \int_\Lambda x(\lambda)\frac{(1-\rho_0-\rho_1)R_\infty(u,\lambda,\nu)+\rho_0}{1-\rho_1 R_\infty(u,\lambda,\nu)}\,\mathrm{d}\lambda,\\
\mathrm{Y}(u,\nu) &= \int_\Lambda y(\lambda)\frac{(1-\rho_0-\rho_1)R_\infty(u,\lambda,\nu)+\rho_0}{1-\rho_1 R_\infty(u,\lambda,\nu)}\,\mathrm{d}\lambda,\\
\mathrm{Z}(u,\nu) &= \int_\Lambda z(\lambda)\frac{(1-\rho_0-\rho_1)R_\infty(u,\lambda,\nu)+\rho_0}{1-\rho_1 R_\infty(u,\lambda,\nu)}\,\mathrm{d}\lambda,
\end{aligned}$$

where ρ_0 and ρ_1 are material parameters. In our setting, which we introduce in the next section, we have $\rho_0 = 0.04$ and $\rho_1 = 0.6$. Moreover, x, y and z are the color matching functions, as given in [24].

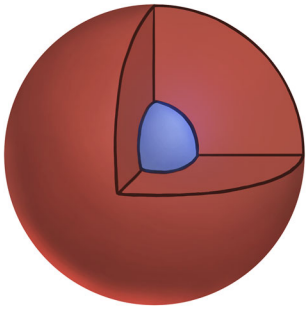

Fig. 2 Radially symmetric core-shell nanoparticle. The inner core (blue) has radius R in the range of 1–75 nm and consists of water. The thickness of the hematite shell (red) is denoted by d and ranges from 1 to 250 nm

2.4 Problem formulation

In our first setting, we consider a radially symmetric core-shell nanoparticle (see Fig. 2), where the inner core consists of water, while the outer shell is made of hematite. Thus, the design u consists of the radius R (1–75 nm) of the core and the thickness d (1–250 nm) of the outer hematite shell, i.e., we have $u = (R, d) \in \mathcal{U} = [1, 75] \times [1, 250]$. Due to the symmetry of the particle, its optical properties do not depend on the orientation $\nu \in \mathbb{S}^2$, which is why we omit it in our further analysis of this setting.

As an additional layer of difficulty, we can, in practice, not expect all nanoparticles present in the paint film to be identical copies of design u. Instead, when trying to produce nanoparticles of a specific design in large quantities, one usually ends up with a mixture of particles of different designs, following a certain probability distribution μ_u, which is dependent on the intended design u.

We model this aspect by assuming that, given a design $u = (R, d)$, the particles present in the paint film follow a truncated normal distribution on the space of reasonable designs $\mathcal{R} \times \mathcal{D} = [10^{-4}, 150] \times [10^{-4}, 500]$ centered around u, i.e.,

$$\tilde{R} \sim \mathcal{N}_{\mathcal{R}}(R, \tfrac{1}{10}R) \quad \text{and} \quad \tilde{d} \sim \mathcal{N}_{\mathcal{D}}(d, \tfrac{1}{10}d).$$

Truncating the normal distribution to the space $\mathcal{R} \times \mathcal{D}$ circumvents nonphysical particles appearing in the design distributions, like designs with negative components. From a numerical point of view, the impact is negligible, as the combined weight of all excluded designs is below typical machine precision, since a design component must deviate from the average by more than 9 standard deviations in order to be rejected. As the paint film no longer consists of identical particles, the K and S values in the Kubelka–Munk model need to be replaced by their averaged counterparts

$$K(u, \lambda) = \iint_{\mathcal{R}\times\mathcal{D}} \mathrm{Abs}(\tilde{R}, \tilde{d}, \lambda)\mathrm{d}\mu_u(\tilde{R}, \tilde{d})$$

and

$$S(u, \lambda) = \iint_{\mathcal{R}\times\mathcal{D}} \mathrm{Sca}(\tilde{R}, \tilde{d}, \lambda)\big(1 - \mathrm{Geo}(\tilde{R}, \tilde{d}, \lambda)\big)\mathrm{d}\mu_u(\tilde{R}, \tilde{d}),$$

before calculating the reflectance $R_\infty(u, \lambda)$ and integrating it over Λ.

The objective in our application is to produce a paint of bright red color. Thus, the complete optimization problem reads

$$\max_{u \in \mathcal{U}} \quad \tfrac{1}{20}\,\mathbf{L}(u) + \tfrac{19}{20}\,\mathbf{a}(u). \tag{1}$$

Due to the compactness of $\mathcal{U}$, $\mathcal{R}$ and $\mathcal{D}$, [2, Assumption 2.2] is obviously satisfied. Furthermore, the mapping from a design u, wavelength λ and orientation v to the optical properties Abs, Sca and Geo is smooth [25, Eqs. 1a, 1b, 1c]. Since every admissible design has a hematite shell of positive thickness, we obtain a lower bound on Abs and Sca. By definition, the geometry factor is always smaller than 1 in absolute value. Consequently, R_∞ depends smoothly on Abs, Sca and Geo. Now, by construction, R_∞ admits values in [0, 1] only. The color matching functions x, y, z are given pointwise and can thus be interpolated with Lipschitz continuous derivative. As a result, X, Y, Z are L-smooth function w.r.t. all arguments. Finally, the function f, appearing in the definition of the color transformation mapping Ψ, is constructed in an L-smooth fashion as well, showing that [2, Assumption 2.3] is satisfied for our setting. By choosing integration weights presented in [2, Section 3], we can also satisfy [2, Assumption 2.4].

2.5 Challenges

The highly condensed fashion, in which (1) is formulated, may obscure a lot of the difficulties that arise when trying to solve it. To get a better understanding of the problem, let us first analyze the abstract structure of the objective function $J(u) = \frac{1}{20}\,\mathbf{L}(u) + \frac{19}{20}\,\mathbf{a}(u)$:

$$\begin{pmatrix} \text{Abs} \\ \text{Sca} \\ \text{Geo} \end{pmatrix} \xrightarrow{\text{integrate } \mathcal{R}\times\mathcal{D}} \begin{pmatrix} K \\ S \end{pmatrix} \xrightarrow{\text{Kubelka-Munk}} R_\infty \xrightarrow{\text{integrate } \Lambda} \begin{pmatrix} \text{X} \\ \text{Y} \\ \text{Z} \end{pmatrix} \xrightarrow{\text{color transf.}\Psi} \begin{pmatrix} \mathbf{L} \\ \mathbf{a} \\ \mathbf{b} \end{pmatrix} \to J(u).$$

Since calculating $J(u)$ and $\nabla J(u)$ requires integrating the optical properties in multiple dimensions and since evaluating said properties for any combination of $\tilde{R}$, $\tilde{d}$ and λ requires solving the time-harmonic Maxwell's equations, standard deterministic approaches, e.g., full gradient methods, run into a prediscretization problem.

On the one hand, the number of integration points needs to be sufficiently large for our setting. In Fig. 3, a slice through the objective function for a fixed value of R and several different amounts of integration points is shown. While we actually do not care too much about the approximation error resulting from a small number of integration points, the artificial local maxima introduced into the objective function by the discretization severely impact the quality of the optimization. In other words, many solutions to the discretized problem are completely unrelated to solutions to (1). We want to note that, even though not all of the stationary points in Fig. 3 correspond to stationary points of (1), the prediscretization still leads to very flat regions in the

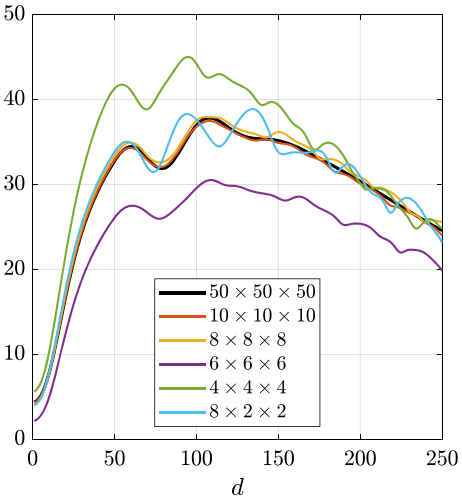

Fig. 3 Objective function values for fixed core radius of 3 nm. Different graphs correspond to different discretizations. The label of a curve shows into how many points the integrals over Λ, $\mathcal{R}$ and $\mathcal{D}$ have been split, respectively. Each of the discretizations introduces artificial stationary points into the objective function

objective functions, which hinder the performance of many solvers. In Fig. 4, this effect is displayed.

On the other hand, the number of integration points is heavily restricted by the computational cost associated to the evaluation of Abs, Sca and Geo. While medium resolutions ($25^3 \sim 15000$ points in total) are still numerically tractable for simple Mie particles, they are outright impossible to achieve in the more general DDA setting, which we want to consider later. For comparison: The optimization in [19] was carried out using a discretization consisting of 20 points in total.

We want to emphasize that standard SG-type schemes, or even the *Stochastic Composition Gradient Descent* (SCGD) method [26], which was used for the comparison for composite objective functions in [2, Section 7.2], are not capable of solving (1). The reason for this lies in the special structure of J, which consists of several integrals nested in nonlinear functions.

2.6 Discretization

For the reasons mentioned above, we will only compare the results obtained by CSG to generic deterministic optimization schemes for various choices of discretization. Since the integration over Λ admits no special structure, we always choose an equidistant partition for this dimension of integration. However, for the integration over $\mathcal{R} \times \mathcal{D}$, we can use our knowledge of μ_u to achieve a better approximation to the true integral. Instead of dividing $\mathcal{R} \times \mathcal{D}$ into an equidistant grid, we utilize the fact that $\tilde{R}$ and $\tilde{d}$ follow truncated one-dimensional normal distributions with parameters independent from each other. Since, for a normal distribution, 99.7% of all weight is concentrated in the 3σ-interval around the mean value, we may only discretize this portion of the full domain in each step.

Moreover, we know the precise density function for both $\tilde{R}$ and $\tilde{d}$. Thus, given a design $u_n = (R_n, d_n)$, we will partition $\left(R_n - \frac{3}{10}R_n, R_n + \frac{3}{10}R_n\right)$ and

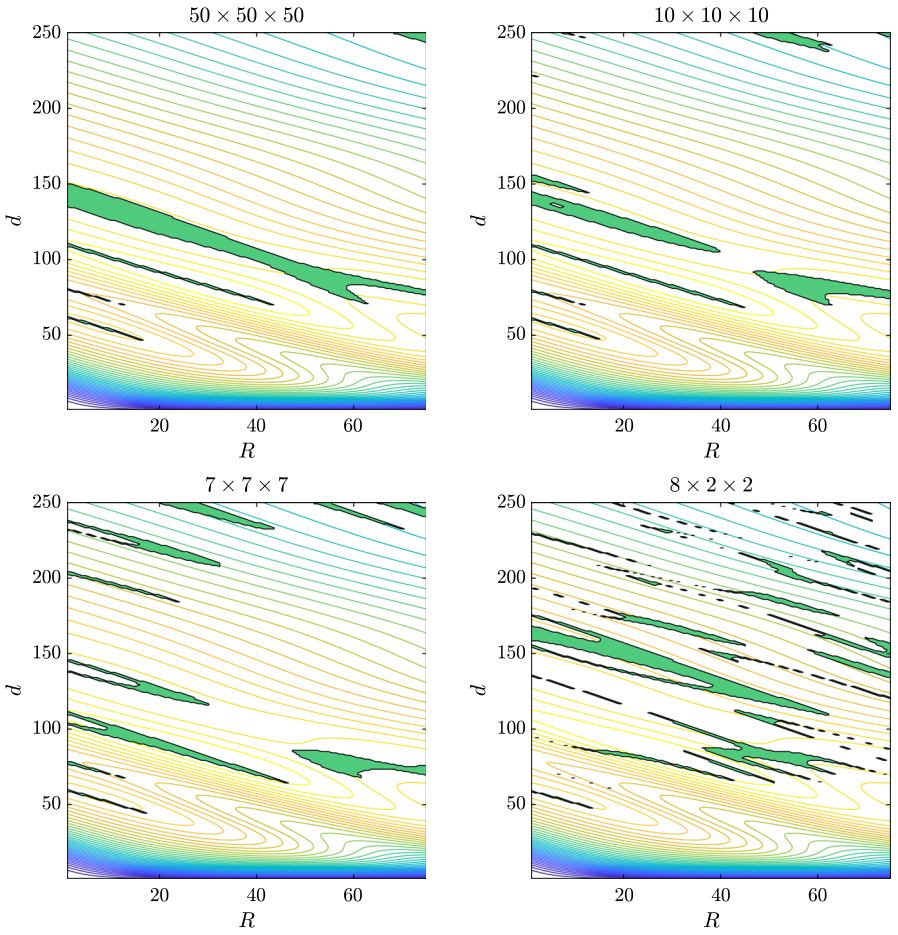

Fig. 4 Flat regions in the discretizted objective functions. The underlying contour plot corresponds to the discretization of $\Lambda \times \mathcal{R} \times \mathcal{D}$ into $50 \times 50 \times 50$ points. For each figure, the green region consists of all points at which the Euclidean norm of the gradient of the discretized objective function is smaller than 0.05. The discretizations of $\Lambda \times \mathcal{R} \times \mathcal{D}$ are given in the titles, respectively

$\left(d_n - \frac{3}{10}d_n, d_n + \frac{3}{10}d_n\right)$ not into equidistant intervals, but instead in intervals of equal weight. This procedure is illustrated in Figs. 5 and 6 and produces very good results even for a small number of sample points.

However, as we have already seen in Fig. 3, even this dedicated discretization scheme introduces additional propbelms into (1). Furthermore, we want to emphasize that choosing a reasonable discretization is a challenge of its own. Not only is there no a priori indication for the general magnitude of the number of points needed, it is also unclear whether or not one should use the same number of points in each direction.

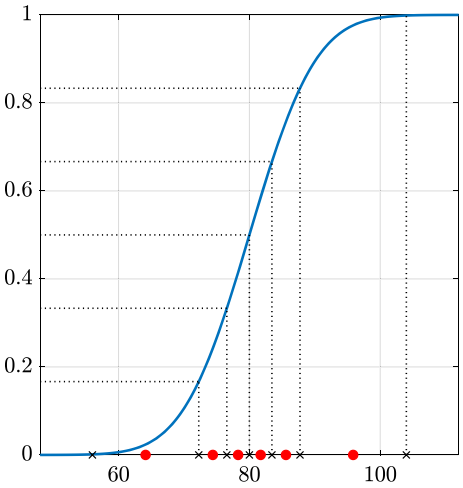

Fig. 5 Cumulative density function for $\tilde{R}$ in the case $R = 80$. The six integration points (red dots) are obtained by dividing (0, 1) in six intervals of equal size and calculating the midpoints of the resulting preimages (black crosses). Note that the preimages are first projected on the 3σ-interval

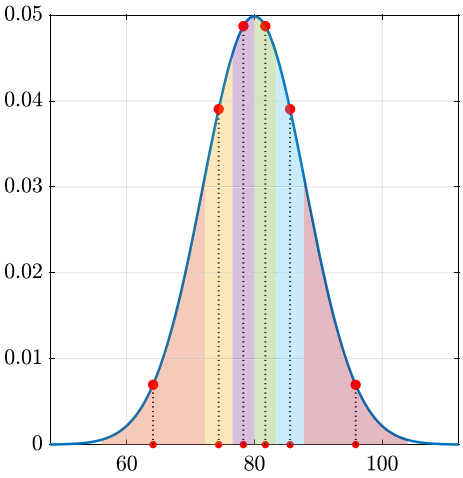

Fig. 6 Density function for $\tilde{R}$ in the case $R = 80$. The red dots represent the six integration points as detailed in Fig. 5. By their special construction, each shaded region under the curve is of equal area

2.7 Numerical results

As mentioned above, the restriction to radially symmetric nanoparticles allows us to apply standard blackbox solvers to (1), in order to have a comparison for the CSG results. In our case, we chose the *fmincon* implementation of an interior point algorithm, integrated in MATLAB, as is it an easy-to-use blackbox algorithm that yields reproducible results.

Specifically, we compared the results of SCIBL-CSG with empirical weights on $\mathcal{R} \times \mathcal{D}$ and exact hybrid weights on Λ (cf. [2, Section 3]) to the fmincon results for three different discretization schemes of $\Lambda \times \mathcal{R} \times \mathcal{D}$. Two of these are equal in each dimension ($10 \times 10 \times 10$ and $7 \times 7 \times 7$), while the last one is asymmetric ($8 \times 2 \times 2$). Once again, we want to stress that finding an appropriate discretization scheme already

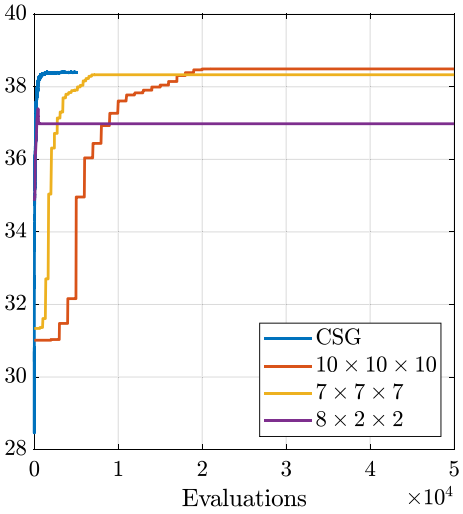

Fig. 7 Median objective function value of all optimization runs in which the final design was closer to the global maximum of (1) than to any other stationary point. The values were obtained using a discretization into $50 \times 50 \times 50$ points

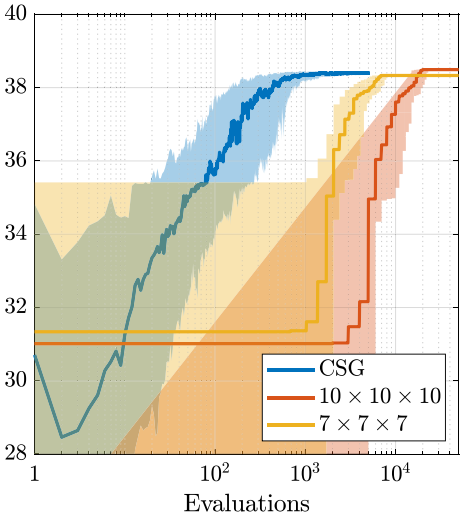

Fig. 8 The medians presented in Fig. 7 (solid lines) and the corresponding quantiles $P_{0.25,0.75}$, indicated by the shaded areas. For better visibility, the number of evaluations is scaled logarithmically and the discretization $8 \times 2 \times 2$ was discarded

requires a thorough analysis of (1). The specific choices listed above represent three of the most promising candidates found during our investigation (Figs. 7, 8).

As we consider this example to be a prototype for more advanced settings from topology optimization, e.g., switching the setting to the DDA model later, we compare the different approaches with respect to the number of inner gradient evaluations, since this is by far the most time-consuming step in these cases. To be precise, an evaluation represents the calculation of Abs, Sca, Geo, ∇ Abs, ∇ Sca and ∇ Geo for a single $(\lambda, \tilde{R}, \tilde{d}) \in \Lambda \times \mathcal{R} \times \mathcal{D}$. These calculations are based on the MATLAB Mie library *MatScat* [27].

Since the produced iterates depend on the initial design, we randomly selected 500 starting points in the whole design domain $\mathcal{U} = [1, 75] \times [1, 250]$. In each optimization

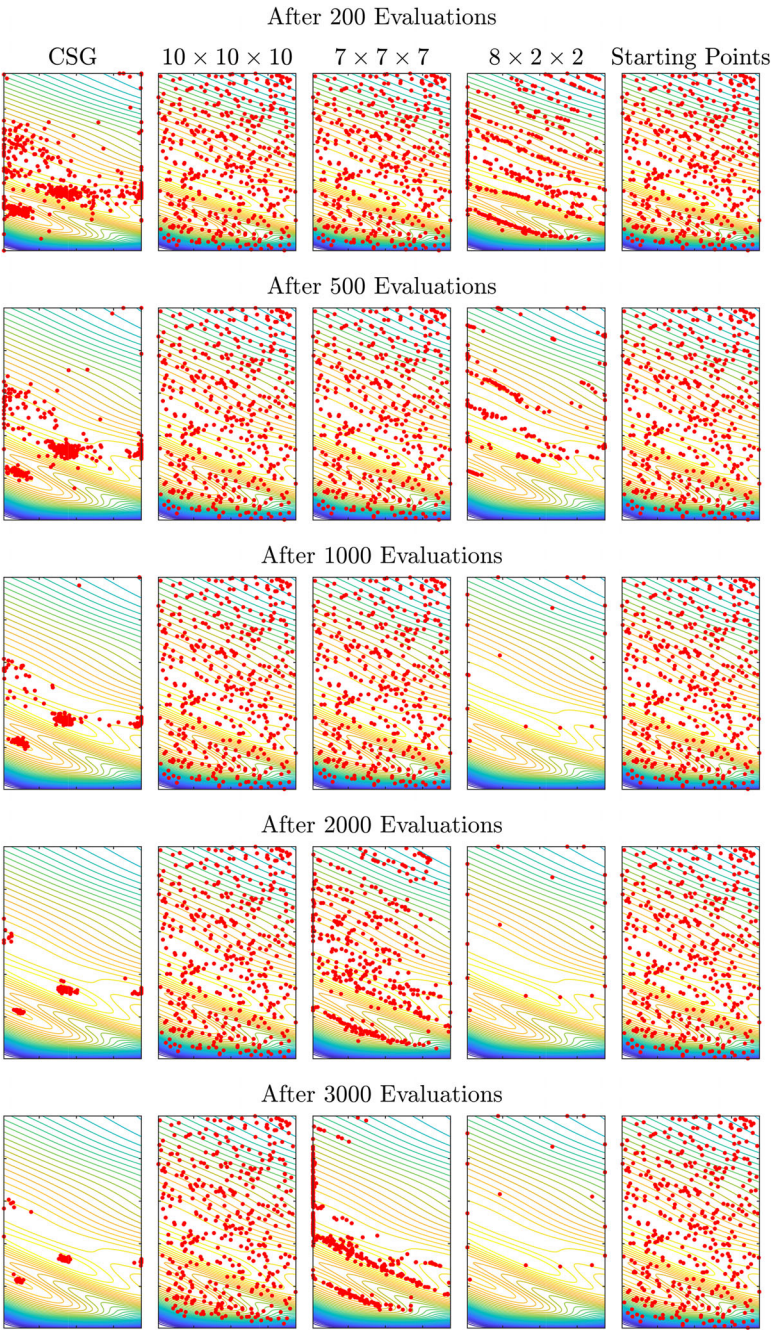

Fig. 9 Iterates of the different optimization approaches for (1) in the whole design domain $\mathcal{U} = [1, 75] \times [1, 250]$. For fmincon, the discretization of $\Lambda \times \mathcal{R} \times \mathcal{D}$ is given in the titles, respectively. To measure the progress, the starting points are also shown. As mentioned above, an evaluation corresponds to the calculation of Abs, Sca, Geo, ∇ Abs, ∇ Sca and ∇ Geo for one combination $(\lambda, \tilde{R}, \tilde{d}) \in \Lambda \times \mathcal{R} \times \mathcal{D}$. Again, the underlying contours are obtained by discretizing $\Lambda \times \mathcal{R} \times \mathcal{R}$ into $50 \times 50 \times 50$ points

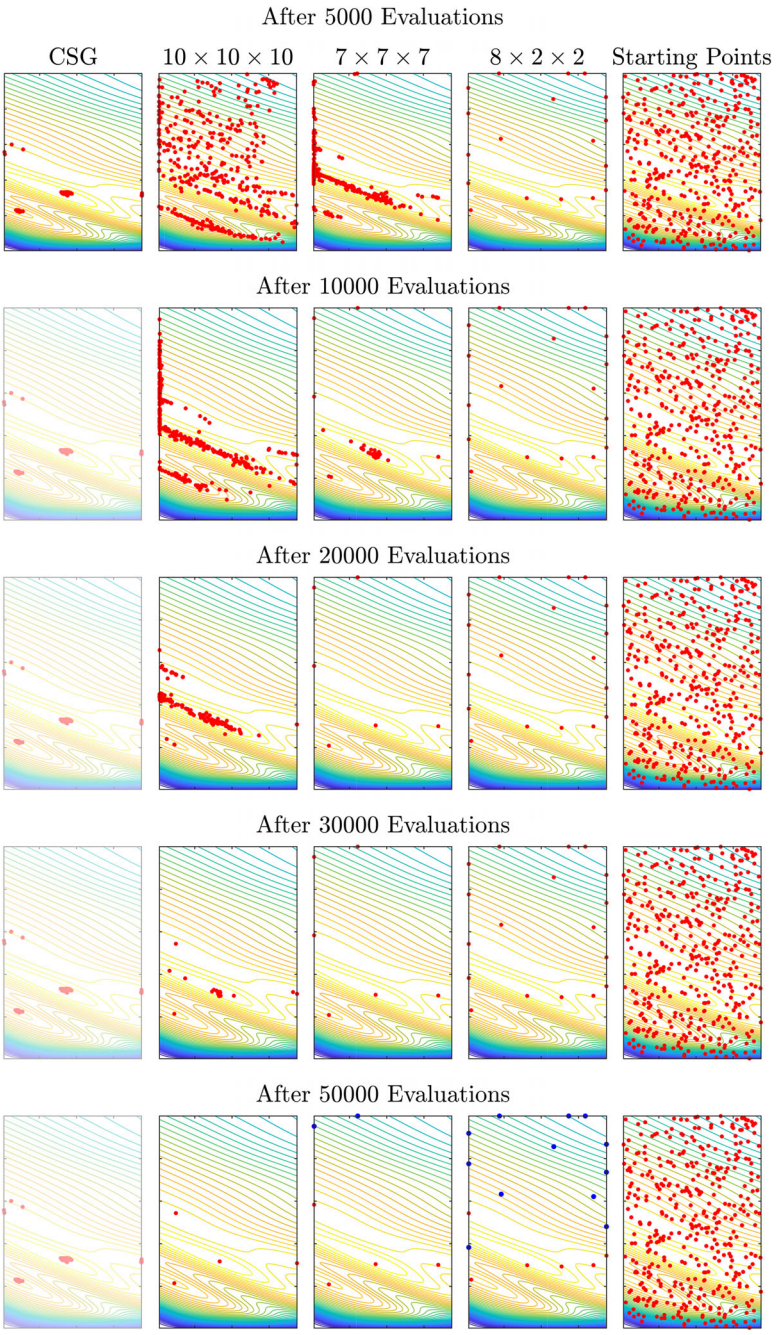

Fig. 10 Continuation of the results for (1) presented in Fig. 9. Since CSG was stopped after 5.000 evaluations, the iterates do not change afterwards, but are still shown as a point of reference. In the last row, final designs obtained by $7 \times 7 \times 7$ and $8 \times 2 \times 2$, which do not correspond to stationary points of (1), are highlighted in blue

run, the total number of evaluations was limited to 50.000 for fmincon and to 5.000 for SCIBL-CSG. To obtain an overview of the general performance of the different approaches, we take snapshots of all iterates after different amounts of evaluations. The results are given in Figs. 9 and 10 and yield a good impression on how fast each method tends to find solutions to (1). Note that, for the sake of readability and better comparison, the final CSG iterates after 5.000 evaluations are shown in all graphs labeled with a higher number of total evaluations.

By comparing Figs. 9 and 10 with Fig. 4, we observe that the artificial flat regions discussed earlier indeed slow down the optimization progress for all choices of pre-discretization. Furthermore, we note that only the highest resolution $10 \times 10 \times 10$ overcomes this approximation error, at the cost of the largest amount of evaluations needed. In contrast, the resolutions $7 \times 7 \times 7$ and $8 \times 2 \times 2$ converge much faster, but some of the final designs are no stationary points of (1). Out of the 500 optimization runs we performed, $7 \times 7 \times 7$ converged to a wrong design, i.e., artificial local minimum, 16 times (3.2%). For $8 \times 2 \times 2$, a wrong design was found in 218 (43.6%) instances, see Fig. 10.

Lastly, we are interested in the performance of each method with respect to $J(u_n)$ over the course of the iterations. Since each local solution to (1) admits a different objective function value, we focus only on the global maximum. For all approaches, we selected all runs whose final designs are closer to the global maximum of (1) than to any other stationary point. The results are shown in Figs. 7 and 8.

2.8 Optimization in the DDA model

As a final example from application, we drop the restriction to core shell particles and consider hematite nanoparticles of arbitrary shape with the DDA model. While the setting is very similar to the setting analyzed above, there are some minor differences.

First, we slightly change the weights appearing in the objective function:

$$\max_{u \in \mathcal{U}} \quad \tfrac{1}{2}\,\mathbf{L}(u) + \tfrac{1}{2}\,\mathbf{a}(u). \tag{2}$$

This change was made purely for aesthetics, as the weights in (1) favour radially symmetric solutions, while (2) admits local solutions with a more interesting design structure. The set $\mathcal{U}$ will be defined later.

Furthermore, we do not assume a particle design distribution anymore, since it is unclear, how such a general shape distribution should look like. However, as the particles are no longer radially symmetric, we now have to consider the orientation of the particle with respect to the incoming light ray instead. Therefore, the K and S values explained in the introduction of this setting need to be averaged over all possible orientations, i.e.,

$$K(u, \lambda) = \frac{1}{\left|\mathbb{S}^2\right|} \iint_{\mathbb{S}^2} \mathrm{Abs}(u, \lambda, \nu) \mathrm{d}\nu$$

and

$$S(u,\lambda) = \frac{1}{|\mathbb{S}^2|}\iint_{\mathbb{S}^2} \mathrm{Sca}(u,\lambda,\nu)\big(1-\mathrm{Geo}(u,\lambda,\nu)\big)\mathrm{d}\nu.$$

Here, $\mathbb{S}^2$ denotes the unit sphere and the particle orientation ν is assumed to be distributed uniformly random over all possible directions.

The design domain is a ball of 300 nm diameter, discretized into $n_0 = 65752$ dipole cells. The design $u \in [\varepsilon, 1]^{n_0} =: \mathcal{U}$ gives the relative amount of hematite to water in each cell, with $\varepsilon = 10^{-4}$. The optical properties of intermediate (grey) material $u^{(i)} \in (0, 1)$ are generated by linear interpolation between the respective properties of water and hematite. Consequently, each admissible design contains a positive amount of hematite, resulting in lower bounds for Abs and Sca. As stated in Sect. 2.4, [2, Assumptions 2.2–2.4] are satisfied, since changing from Mie theory to the DDA model does not interfere with the smoothness of Abs, Sca and Geo w.r.t. (u, λ, ν), see [19, 28].

Generally, one would combine filtering techniques and greyness penalization to obtain a smooth final design without intermediate material (see, e.g., [29]). However, we explicitly refrain from doing so to present a clear analysis of the CSG performance, without interference from secondary layers of smoothing techniques.

As mentioned above, the change to the DDA model significantly increases the computational cost of evaluating Sca, Abs and Geo for a given $(u, \lambda, \nu) \in \mathcal{U} \times \Lambda \times \mathbb{S}^2$. Thus, the deterministic approaches used in the previous setting are no longer computationally feasible.

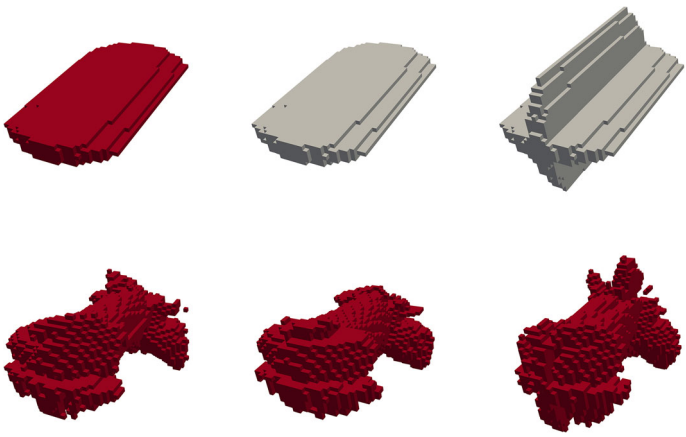

Fig. 11 Representation of the initial designs (top row). Red boxes correspond to cells consisting purely of hematite, while grey boxes indicate an artificial intermediate material, consisting of 50% hematite and 50% water. For later references, we denote the initial designs by *plate (100%)*, *plate (50%)* and *screwdriver (50%)*, respectively. The different final designs, obtained by 5.000 iterations of SCIBL-CSG with outer norm (a) are shown in the bottom row. For better visibility, cells with less than 50% hematite are considered as pure water and left out of the visualization. For each final design, the amount of cells discarded in this fashion is less than 100 (less than 0.15% of all cells)

Furthermore, we want to use this example to analyze the impact of the chosen norm on $\mathcal{U} \times \Lambda \times \mathbb{S}^2$, appearing in the nearest neighbor calculation, which was already mentioned in [2, Section 3.5]. To be precise, calculating the CSG integration weights requires the definition of an outer norm

$$\left\| (u^*, \lambda^*, \nu^*) \right\|_{\text{Out}} = c_u \|u^*\|_{\mathcal{U}} + c_\lambda \|\lambda^*\|_{\Lambda} + c_\nu \|\nu^*\|_{\mathbb{S}^2},$$

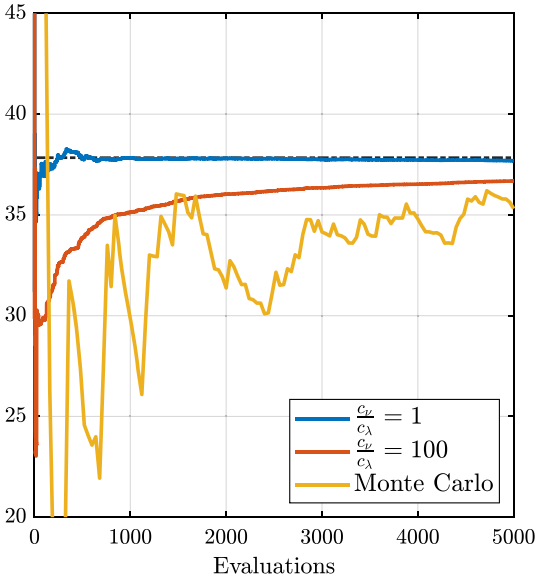

Fig. 12 Objective function approximation for the *screwdriver (50%)* design. The blue and orange curve show the results for CSG with fixed step size $\tau = 0$ and different coefficients of the outer norm $\|\cdot\|_{\text{Out}}$. For Monte Carlo, each inner integral over $\mathbb{S}^2$ was approximated using 40 random directions. The true objective function value $J^* \approx 37.84$ is indicated by the dashed line. The Monte Carlo results are truncated for the sake of readability, as it requires over 8.000 evaluations to reach a good approximation to J^*

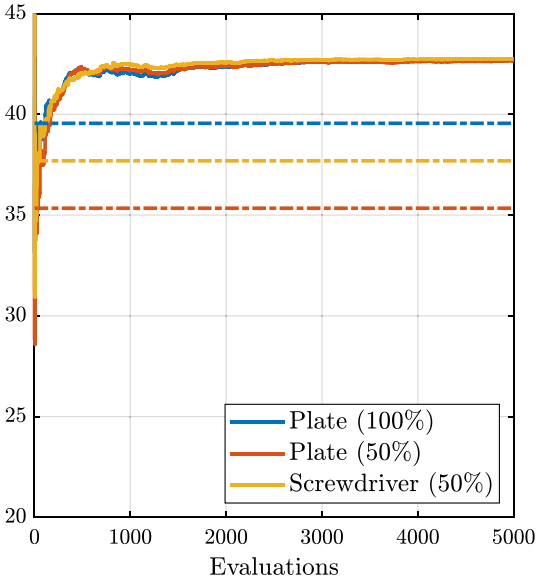

Fig. 13 CSG objective function approximations during the optimization process for all initial designs and choice (a) for $\|\cdot\|_{\text{Out}}$, i.e., $c_u = 1$, $c_\lambda = 100$ and $c_\nu = 100$. The dashed lines indicate the objective function values of each initial design, respectively

where $\|\cdot\|_{\mathcal{U}}$, $\|\cdot\|_{\Lambda}$ and $\|\cdot\|_{\mathbb{S}^2}$ denote norms on the corresponding inner spaces and $c_u, c_\lambda, c_v > 0$. In this application, we choose the Euclidean norm $\|\cdot\|_2$ for each inner space. Additionally, we fix $c_u = 1$, but consider different coefficients c_λ and c_v.

For the optimization, we consider three different initial designs, which are shown in Fig. 11, top row. The objective function value as well as the values of $\mathbf{L}$, $\mathbf{a}$ and $\mathbf{b}$ for these designs were computed using the CSG method with fixed design, i.e., with constant step size $\tau = 0$, and verified by Monte Carlo (see, e.g., [30]) integration. For one of the initial designs, the objective function value approximation of CSG and Monte Carlo integration with respect to the number of evaluations and different choices of $\|\cdot\|_{\text{Out}}$ is shown in Fig. 12.

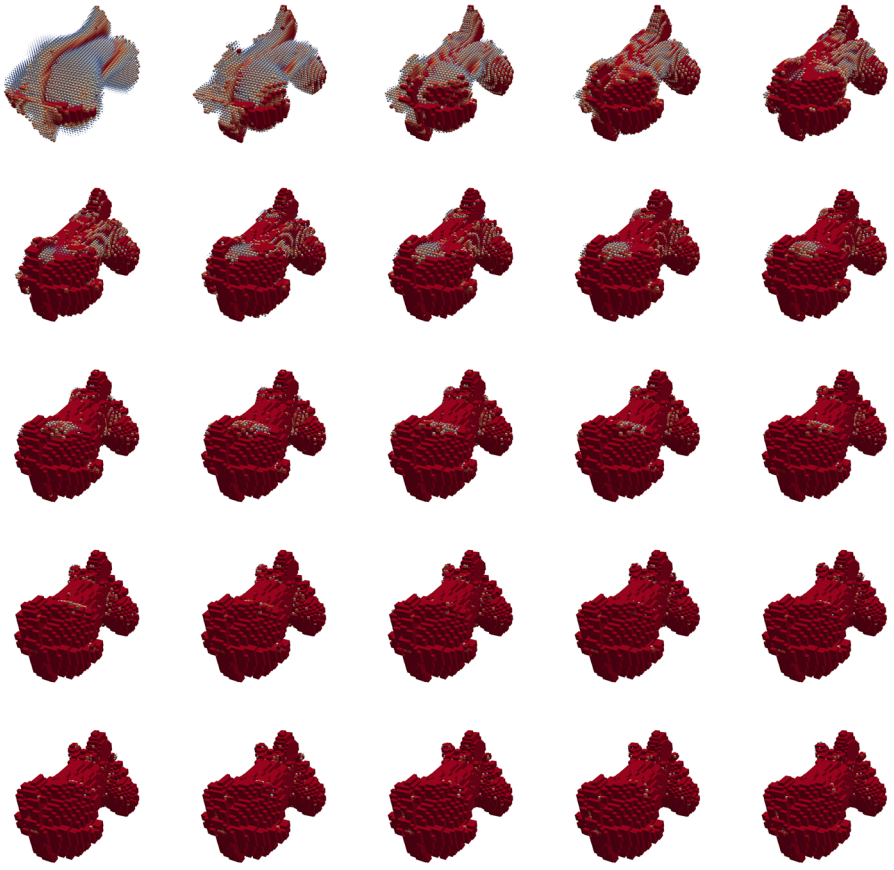

Fig. 14 Top left to bottom right: Design evolution during the optimization process for the *screwdriver (50%)* initial design and outer norm (a). The design snapshots were taken every 200 iterations. Red boxes represent design cells consisting of pure hematite. Intermediate material is indicated via a color gradient, where a cell filled with 50% water and 50% hematite is colored grey. Based on this gradient, depending on the ratio of hematite and water in a cell, the cell color is shifted to red (more hematite) or blue (more water)

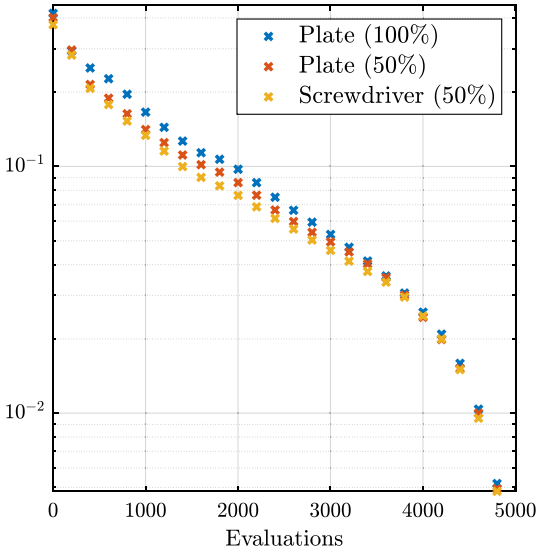

Fig. 15 Euclidean distance (after dividing by $\sqrt{\dim(\mathcal{U})}$ for scaling) between intermediate designs and the respective final design during the SCIBL-CSG optimization process, carried out with outer norm (a)

Each design was optimized with SCIBL-CSG, using inexact hybrid weights for the integration over $\mathbb{S}^2$ and exact hybrid weights for the integration over Λ. For $\|\cdot\|_{\text{Out}}$, we considered four different choices of the parameters:

(a) $c_u = 1$, $c_\lambda = 100$ and $c_\nu = 100$
(b) $c_u = 1$, $c_\lambda = 1$ and $c_\nu = 1$
(c) $c_u = 1$, $c_\lambda = \frac{1}{100}$ and $c_\nu = 1$
(d) $c_u = 1$, $c_\lambda = \frac{1}{100}$ and $c_\nu = \frac{1}{100}$

The results in case (a) for all three initial designs are presented in Fig. 13 and the respective design evolution for the initial design *screwdriver (50%)*, shown in Fig. 11 top row, is depicted in Fig. 14. The corresponding final designs, obtained after 5.000 SCIBL-CSG iterations, are presented in Fig. 11, bottom row. As a second measure for convergence in the design space, the evolution of the norm distance to the respective final designs are shown in Fig. 15 for all three initial designs.

Comparing Figs. 12 and 13, we notice that CSG, using an appropriate outer norm, finds an optimized design almost as fast as it computes the objective function value for a given design. In other words: The full optimization process is only slightly more expensive that the simple evaluation of a single design. Moreover, CSG finds an optimal solution to (2) long before the Monte Carlo approximation to the initial objective function value is converged.

It should, of course, also be noted, that choosing $\|\cdot\|_{\text{Out}}$ should be done with caution, as Fig. 16 shows. While case (a) is, to the best of our knowledge, *not* optimal by any means, cases (b) and (c) clearly show worse results. Choosing $\|\cdot\|_{\text{Out}}$ extremely poorly, i.e., case (d), can even have devastating effects on the performance, see Fig. 17.

This, however, could also imply that the performance might be significantly improved, if problem specific inner and outer norms would be chosen. Especially

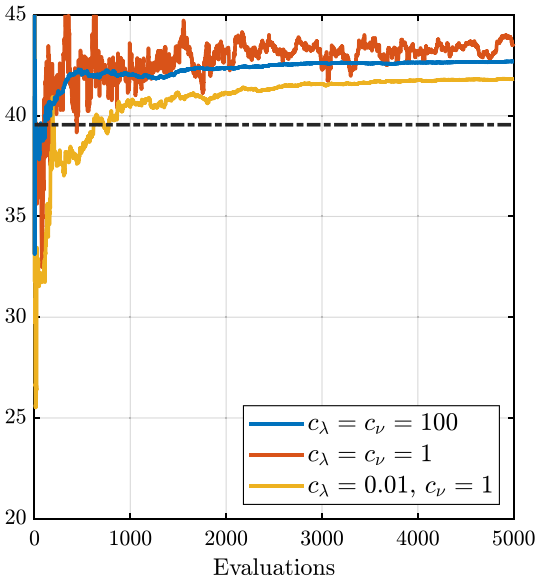

Fig. 16 CSG objective function value approximation during the optimization process for the *plate (100%)* initial design. The dashed line shows the inital objective function value, whereas the different graphs correspond to the choices (a), (b) and (c) for $\|\cdot\|_{\text{Out}}$

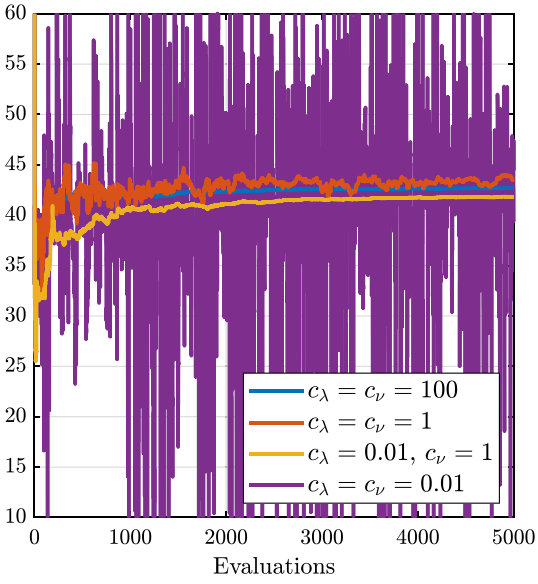

Fig. 17 Results for the *plate (100%)* initial design presented in Fig. 16, augmented by the CSG objective function value approximation in the case that $\|\cdot\|_{\text{Out}}$ was chosen according to (d)

in even more complex settings, techniques to obtain such norms a priori, or even during the optimization process itself, represent one of the most important points for further research.

3 Online error estimation

Before we go into theoretical details, we first collect a few key properties and results concerning CSG, which were shown in [2]. In a first simple setting, we consider optimization problems of the form

$$\begin{aligned} \min \quad & J(u) \\ \text{s.t.} \quad & u \in \mathcal{U} \subset \mathbb{R}^{d_\mathrm{o}} \text{ for some } d_\mathrm{o} \in \mathbb{N}. \end{aligned} \tag{3}$$

Additionally, we assume that $\mathcal{U}$ is compact, and for some $d_\mathrm{r} \in \mathbb{N}$, there exists an open an bounded set $\mathcal{X} \subset \mathbb{R}^{d_\mathrm{r}}$ and a measure μ with $\mathrm{supp}(\mu) \subset \mathcal{X}$, such that J can be written as $J(u) = \int_{\mathcal{X}} j(u, x)\mu(\mathrm{d}x)$. The detailed set of assumptions is given in [2, Section 2]. For now, it is only important that $\nabla_1 j : \mathcal{U} \times \mathcal{X} \to \mathbb{R}^{d_\mathrm{o}}$ is bounded and Lipschitz continuous, i.e., there exist $C, L_j > 0$ with

$$\begin{aligned} \|\nabla_1 j(u, x)\| &\leq C, \\ \|\nabla_1 j(u_1, x_1) - \nabla_1 j(u_2, x_2)\| &\leq L_j\big(\|u_1 - u_2\|_{\mathcal{U}} + \|x_1 - x_2\|_{\mathcal{X}}\big) \end{aligned}$$

for all $(u, x), (u_1, x_1), (u_2, x_2) \in \mathcal{U} \times \mathcal{X}$. Due to the finite dimension of all appearing spaces, we can choose arbitrary norms on $\mathcal{U}$, $\mathcal{X}$ and $\mathbb{R}^{d_\mathrm{o}}$, and simply denote them by $\|\cdot\|_{\mathcal{U}}$, $\|\cdot\|_{\mathcal{X}}$ and $\|\cdot\|$, respectively, unless specific choices are made in numerical experiments.

During the optimization process, CSG computes design dependent integration weights $\big(\alpha_k\big)_{k=1\ldots,n}$ (cf. [2, Section 3]) to build an approximation $\hat{G}_n$ to the true objective function gradient, based on the available samples from previous iterations $\big(\nabla_1 j(u_k, x_k)\big)_{k=1,\ldots,n}$. To be precise, we have

$$\nabla J(u) = \int_{\mathcal{X}} \nabla_1 j(u, x)\mu(\mathrm{d}x) \approx \sum_{k=1}^{n} \alpha_k \nabla_1 j(u_k, x_k) =: \hat{G}_n.$$

It was shown in [2, Lemma 4.6], that

$$\|\nabla J(u_n) - \hat{G}_n\| \to 0 \quad \text{for } n \to \infty \text{ almost surely.}$$

Carefully investigating the methods to obtain the integration weights, we observe that

$$\begin{aligned} \left\|\nabla J(u_n) - \hat{G}_n\right\| &= \left\|\int_{\mathcal{X}} \nabla_1 j(u_n, x)\mu(\mathrm{d}x) - \hat{G}_n\right\| \\ &= \left\|\sum_{i=1}^{n} \int_{M_i} \nabla_1 j(u_n, x)\mu(\mathrm{d}x) - \sum_{i=1}^{n} \nabla_1 j(u_i, x_i)\nu_n(M_i)\right\|, \end{aligned}$$

where ν_n denotes the measure associated to one of the measures listed in [2, Section 3.6], depending on the choice of integration weights, and

$$M_k := \big\{x \in \mathcal{X} \,:\, \|u_n - u_k\|_{\mathcal{U}} + \|x - x_k\|_{\mathcal{X}} \\ < \|u_n - u_j\|_{\mathcal{U}} + \|x - x_j\|_{\mathcal{X}} \text{ for all } j \in \{1, \dots, n\} \setminus \{k\}\big\}.$$

By construction, M_k contains all points $x \in \mathcal{X}$, such that (u_n, x) is closer to (u_k, x_k) than to any other previous point we evaluated $\nabla_1 j$ at. For exact integration weights, we have $\nu_n = \mu$ and thus

$$\begin{aligned}\left\|\nabla J(u_n) - \hat{G}_n\right\| &= \left\|\sum_{i=1}^n \int_{M_i} \nabla_1 j(u_n, x)\mu(\mathrm{d}x) - \sum_{i=1}^n \int_{M_i} \nabla_1 j(u_i, x_i)\mu(\mathrm{d}x)\right\| \\ &\le \sum_{i=1}^n \int_{M_i} \|\nabla_1 j(u_n, x) - \nabla_1 j(u_i, x_i)\|\, \mu(\mathrm{d}x) \\ &\le \sum_{i=1}^n \int_{M_i} L_j \cdot \left(\sup_{x \in M_i} Z_n(x)\right) \mu(\mathrm{d}x) \\ &= L_j \sum_{i=1}^n \mu(M_i) \sup_{x \in M_i} Z_n(x) \\ &\le L_j \sup_{x \in \mathcal{X}} Z_n(x).\end{aligned}$$

Here, Z_n is given by

$$Z_n(x) := \min_{k \in \{1,\dots,n\}} \big(\|u_n - u_k\|_{\mathcal{U}} + \|x - x_k\|_{\mathcal{X}}\big).$$

In other words, the approximation error can be bounded in terms of the Lipschitz constant of $\nabla_1 j$ and the quantity Z_n, which relates to the size of Voronoi cells [31] with positive integration weights.

Both L_j and $\sup_{x \in \mathcal{X}} Z_n(x)$ can be efficiently approximated during the optimization process, e.g. by finite differences of the samples $\big(\nabla_1 j(u_i, x_i)\big)_{i=1,\dots,n}$ and by

$$\sup_{x \in \mathcal{X}} Z_n(x) \approx \max_{k=1,\dots,n} Z_n(x_k),$$

yielding an online error estimation. Such an approximation may, for example, be used in stopping criteria.

4 Convergence rates

Throughout this section, we assume [2, Assumptions 2.1–2.4] to be satisfied. Moreover, for the entire section, let $(u_n)_{n \in \mathbb{N}}$ correspond to the CSG iterates produced for a

fixed random sequence $(x_n)_{n\in\mathbb{N}}$. Then, with probability 1, we have

$$\left\|\hat{G}_n - \nabla J(u_n)\right\| \to 0,$$

see [2, Lemma 4.6]

4.1 Theoretical background

In the convergence analysis presented in [2], we have already seen that the fashion in which the gradient approximation $\hat{G}_n$ is calculated in CSG is crucial for $\|\hat{G}_n - \nabla J(u_n)\| \to 0$ and that this property of CSG in turn is the key to all advantages CSG offers in comparison to classic stochastic optimization methods, like convergence for constant steps, backtracking, more involved optimization problems, etc.

The price we pay for this feature lies within the dependency of $\hat{G}_n$ on the past iterates. For comparison, the search direction $\hat{G}_n^{\mathrm{SG}}$ in a stochastic gradient descent method is given by

$$\hat{G}_n^{\mathrm{SG}} = \nabla_1 j(u_n, x_n).$$

Thus, it is independent of all previous steps and fulfills

$$\mathbb{E}_{\mathcal{X}}\left[\hat{G}_n^{\mathrm{SG}}\right] = \mathbb{E}_{\mathcal{X}}\left[\nabla_1 j(u_n, \cdot)\right] = \nabla J(u_n),$$

i.e., it is an unbiased sample of the full gradient. The combination of these properties allows for a straightforward convergence rate analysis, see, e.g., [32].

In contrast, $\hat{G}_n$ is in general *not* an unbiased approximation to $\nabla J(u_n)$ and moreover *not* independent of $(u_i, x_i)_{i=1,\dots,n-1}$. The main problem in finding the convergence rate of $\|u_{n+1} - u_n\|_{\mathcal{U}} \to 0$ is, that this quantity depends on the approximation error $\|\hat{G}_n - \nabla J(u_n)\|$, which, as we have seen in Sect. 3, depends on Z_n. Since Z_n itself is deeply connected to $\min_k \|u_n - u_k\|_{\mathcal{U}}$, we run into a circular argument.

Therefore, up to now, we are not able to prove convergence rates for the CSG iterates. We can, however, state a prediction to this rate and provide numerical evidence.

Conjecture 4.1 *We conjecture that the CSG method, applied to problem (3), using a constant step size $\tau < \frac{2}{L}$ and empirical integration weights, fulfills*

$$\|u_{n+1} - u_n\|_{\mathcal{U}} = \mathcal{O}\left(\ln(n) \cdot n^{-\frac{1}{\max\{2, d_{\mathrm{r}}\}}}\right)$$

with probability 1.

To motivate this claim, note that, in the proof of [2, Lemma 4.6], it was shown that there exists $C > 0$ such that

$$\left\|\hat{G}_n - \nabla J(u_n)\right\| \le C\left(\int_{\mathcal{X}} Z_n(x)\mu(\mathrm{d}x) + d_W(\mu_n, \mu)\right),$$

where d_W denotes the Wasserstein distance of the two measures μ_n and μ. By [33, Theorem 1], the empirical measure μ_n satisfies

$$\mathbb{E}\big[d_W(\mu_n,\mu)\big] \le C(d_{\mathrm{r}}) \cdot \left(\int_{\mathcal{X}} \|x\|_{\mathcal{X}}^3 \mu(\mathrm{d}x)\right)^{\frac{1}{3}} \cdot \begin{cases} \frac{1}{\sqrt{n}} & \text{if } d_{\mathrm{r}} = 1, \\ \frac{\ln(1+n)}{\sqrt{n}} & \text{if } d_{\mathrm{r}} = 2, \\ n^{-\frac{1}{d_{\mathrm{r}}}} & \text{if } d_{\mathrm{r}} \ge 3. \end{cases}$$

This result is the main motivation for Conjecture 4.1. It can be shown that the rate $n^{-1/d_{\mathrm{r}}}$ for $d_{\mathrm{r}} \ge 3$ is sharp if μ corresponds to a uniform distribution on $\mathcal{X}$. Thus, in this case, it is reasonable to assume a uniform distribution also corresponds to the worst-case rate of $\int_{\mathcal{X}} Z_n(s)\mu(\mathrm{d}x) \to 0$. Assuming that the difference in designs appearing in Z_n is negligible due to the overall convergence of CSG, we obtain the rate

$$\sup_{x\in\mathcal{X}} Z_n(x) = \mathcal{O}\left(\ln(n) \cdot n^{-\frac{1}{\max\{2,d_{\mathrm{r}}\}}}\right).$$

To see this, we fill $\mathcal{X} \subset \mathbb{R}^{d_{\mathrm{r}}}$ with balls (w.r.t. the norm $\|\cdot\|_{\mathcal{X}}$) of radius $\varepsilon > 0$ and denote by $N(\varepsilon) \in \mathbb{N}$ the number of cells. Due to the dimension of $\mathcal{X}$, we have $\mathcal{O}\big(N(\varepsilon)\big) = \varepsilon^{-d_{\mathrm{r}}}$. Now, to achieve $\sup_{x\in\mathcal{X}} Z_n(x) < \varepsilon$, we need each of these cells to contain at least one of the sample points $(x_i)_{i=1,\dots,n}$. It is well-known that the expected number of samples we need to draw for this to happen is given by

$$N(\varepsilon) \sum_{k=1}^{N(\varepsilon)} \frac{1}{k} = \mathcal{O}\left(-\varepsilon^{-d_{\mathrm{r}}} \ln(\varepsilon)\right),$$

where we used

$$\sum_{k=1}^{n} \frac{1}{k} = \mathcal{O}\big(\ln(n)\big) \quad \text{for } n \to \infty.$$

In other words, the convergence rates of $\int_{\mathcal{X}} Z_n(x)\mu(\mathrm{d}x) \to 0$ and $d_W(\mu_n,\mu) \to 0$ are comparable.

Now that we motivated the rates claimed in Conjecture 4.1 for the approximation error $\|\hat{G}_n - \nabla J(u_n)\|$, we use the following proposition to show that the rates of $\|u_{n+1} - u_n\|_{\mathcal{U}} \to 0$ can not be worse.

Proposition 4.2 *Assume that the approximation error $\|\hat{G}_n - \nabla J(u_n)\|$ satisfies*

$$\|\hat{G}_n - \nabla J(u_n)\| = \mathcal{O}\left(\ln(n) \cdot n^{-\frac{1}{\max\{2,d_{\mathrm{r}}\}}}\right).$$

Then, under the assumptions of Conjecture 4.1, it holds

$$\|u_{n+1} - u_n\|_{\mathcal{U}} = \mathcal{O}\left(\ln(n) \cdot n^{-\frac{1}{\max\{2,d_{\mathrm{r}}\}}}\right).$$

Proof Assume for contradiction that this is not the case. Thus, there exists $N \in \mathbb{N}$ such that

$$\left\| \nabla J(u_n) - \hat{G}_n \right\| \leq \tfrac{1}{2} \left(\tfrac{1}{\tau} - \tfrac{L}{2} \right) \|u_{n+1} - u_n\|_{\mathcal{U}} \quad \text{for all } n \geq N. \tag{4}$$

By the descent lemma [34, Lemma 5.7], the characteristic property of the projection operator [34, Theorem 6.41] and the Cauchy-Schwarz inequality, we obtain

$$\begin{aligned} & J(u_{n+1}) - J(u_n) \\ & \quad \leq \nabla J(u_n)^\top (u_{n+1} - u_n) + \tfrac{L}{2} \|u_{n+1} - u_n\|_{\mathcal{U}}^2 \\ & \quad = \hat{G}_n^\top (u_{n+1} - u_n) + \tfrac{L}{2} \|u_{n+1} - u_n\|_{\mathcal{U}}^2 + \left(\nabla J(u_n) - \hat{G}_n \right)^\top (u_{n+1} - u_n) \\ & \quad \leq \left(\tfrac{L}{2} - \tfrac{1}{\tau} \right) \|u_{n+1} - u_n\|_{\mathcal{U}}^2 + \left\| \nabla J(u_n) - \hat{G}_n \right\| \cdot \|u_{n+1} - u_n\|_{\mathcal{U}} \\ & \quad = \left(\left(\tfrac{L}{2} - \tfrac{1}{\tau} \right) \|u_{n+1} - u_n\|_{\mathcal{U}} + \left\| \nabla J(u_n) - \hat{G}_n \right\| \right) \|u_{n+1} - u_n\|_{\mathcal{U}}. \end{aligned}$$

Combining this with (4) gives $J(u_{n+1}) \leq J(u_n)$ for all $n \geq N$, since $\frac{L}{2} < \frac{1}{\tau}$. Thus, the sequence of objective function values $\big(J(u_n)\big)_{n \in \mathbb{N}}$ is monotonically decreasing for all $n \geq N$. By continuity of J and compactness of $\mathcal{U}$, J is bounded and $J(u_n) \to \bar{J}$ for some $\bar{J} \in \mathbb{R}$. Therefore,

$$-\infty < \bar{J} - J(u_N) = \sum_{n=N}^{\infty} \big(J(u_{n+1} - J(u_n) \big) \leq \tfrac{1}{2} \left(\tfrac{L}{2} - \tfrac{1}{\tau} \right) \sum_{n=N}^{\infty} \|u_{n+1} - u_n\|_{\mathcal{U}}^2.$$

Hence, the series

$$\sum_{n=N}^{\infty} \|u_{n+1} - u_n\|_{\mathcal{U}}^2$$

converges, contradicting $\|u_{n+1} - u_n\|_{\mathcal{U}} \neq \mathcal{O}\left(\ln(n) \cdot n^{-\frac{1}{\max\{2, d_{\mathrm{r}}\}}} \right)$. □

4.2 Numerical verification

We want to verify the proclaimed rates numerically. For this purpose, we consider two optimization problems that can easily be scaled to high dimensions. The first problem is given by

$$\min_{u \in \mathcal{U}} \quad \frac{1}{2} \int_{\mathcal{X}} \|u - x\|_2^2 \mathrm{d}x, \tag{5}$$

where $\mathcal{X} = \left[-\frac{1}{2}, \frac{1}{2}\right]^{d_\mathrm{r}}$ and $\mathcal{U} = [-5, 5]^{d_\mathrm{r}}$, i.e., $\mathcal{U}$ and $\mathcal{X}$ have the same dimension. The second problem,

$$\min_{u \in \mathcal{U}} \quad \frac{1}{2} \int_{-0.5}^{0.5} \left\| u - x \cdot \mathbb{1}_{d_\mathrm{o}} \right\|_2^2 \mathrm{d}x, \tag{6}$$

fixes $d_\mathrm{r} = 1$, while $\mathcal{U} = [-5, 5]^{d_\mathrm{o}}$. Here, $\mathbb{1}_{d_\mathrm{o}}$ represents the vector $(1, 1, \ldots, 1)^\top \in \mathbb{R}^{d_\mathrm{o}}$. Note that, in both settings, we have $L_j = 1$. Thus, by Sect. 3, we have

$$\left\| \hat{G}_n - \nabla J(u_n) \right\|_2 \leq \sup_{x \in \mathcal{X}} Z_n(x) \approx \max_{k=1,\ldots,n} Z_n(x_k).$$

The optimal solution to (5) and (6) is given by the zero vector $u^* = 0 \in \mathcal{U}$.

In our analysis, for different values of the dimensions $d_\mathrm{r}, d_\mathrm{o} \in \mathbb{N}$, problems (5) and (6) were initialized with 500 random starting points. The constant step size of CSG was chosen as $\tau = \frac{1}{2}$. We track $\|u_n - u^*\|_2$ and $\max_{k=1,\ldots,n} Z_n(x_k)$ during the optimization process and compare the median of the 500 runs to the rates predicted in Conjecture 4.1. The results can be seen in Figs. 18, 19, 20, and 21. Note that, for the plots of the predicted rates, we omitted the factor $\ln(n)$. Therefore, the corresponding graphs are straight lines, where the slope $-\frac{1}{\max\{2, d_\mathrm{r}\}}$ is equal to the asymptotic slope

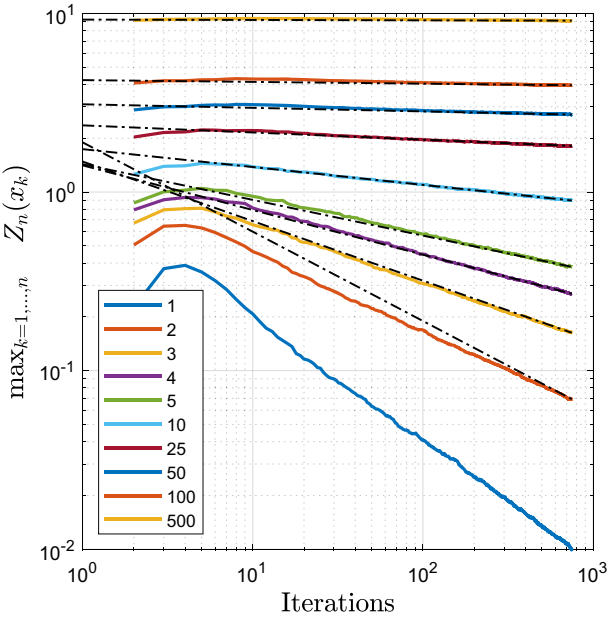

Fig. 18 The bold lines represent the median values of $\max_{k=1,\ldots,n} Z_n(x_k)$ for the equidistant problem (5) with respect to the iteration counter. The different colors indicate the different dimensions $d_\mathrm{r} \in \{1, 2, \ldots, 500\}$. The dotted lines correspond to the respective predicted rates $n^{-\frac{1}{\max\{2, d_\mathrm{r}\}}}$. Since the predictions for $d_\mathrm{r} = 1$ and $d_\mathrm{r} = 2$ are equal, only the case $d_\mathrm{r} = 2$ is shown

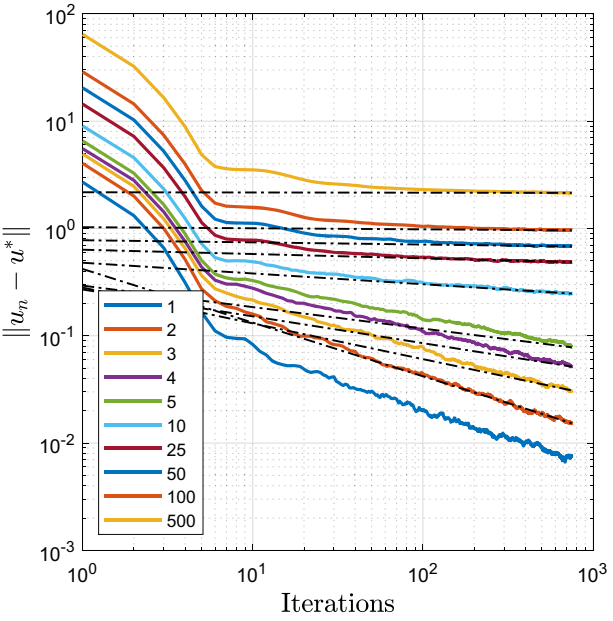

Fig. 19 Median values of $\|u_n - u^*\|$ in the equidimensional setting (5) for different choices of $d_\mathrm{r} \in \{1, 2, \ldots, 500\}$. For each dimension, the predicted worst-case asymptotic line $n^{-\frac{1}{\max\{2,d_\mathrm{r}\}}}$ is indicated by the dotted line. Again, we omit the prediction for $d_\mathrm{r} = 1$, since it has the same slope as in the case for $d_\mathrm{r} = 1$

of the predicted rate, since

$$\ln(n) \cdot n^{-\frac{1}{\max\{2,d_\mathrm{r}\}}} = \mathcal{O}\left(n^{-\frac{1}{\max\{2,d_\mathrm{r}\}}+\varepsilon}\right) \quad \text{for all } \varepsilon > 0.$$

In the equidimensional, i.e., $\dim(\mathcal{X}) = \dim(\mathcal{U})$, setting (5), the experimentally obtained values for Z_n almost perfectly match the claimed rates. For $\|u_n - u^*\|_2$, the observed rates also match the predictions for very small and large dimensions. For $d_\mathrm{r} = 3, 4, 5$, the convergence obtained in the experiments was even slightly faster than predicted. Investigating the results for (6), it is clearly visible that increasing the design dimension d_o, while keeping the parameter dimension d_r fixed, has no influence on the obtained rates of convergence, indicating that CSG is able to efficiently handle large-scale optimization problems.

4.3 Circumventing slow convergence

As we have seen so far, the convergence rate of the CSG method worsens with increasing dimension of integration $d_\mathrm{r} \in \mathbb{N}$. However, it is possible to circumvent this behavior, if the problem admits additional structure. Assume that there exist suitable $\mathcal{X}_1$, $\mathcal{X}_2$, μ_1, μ_2, f_1 and f_2 such that the objective function appearing in (3) can

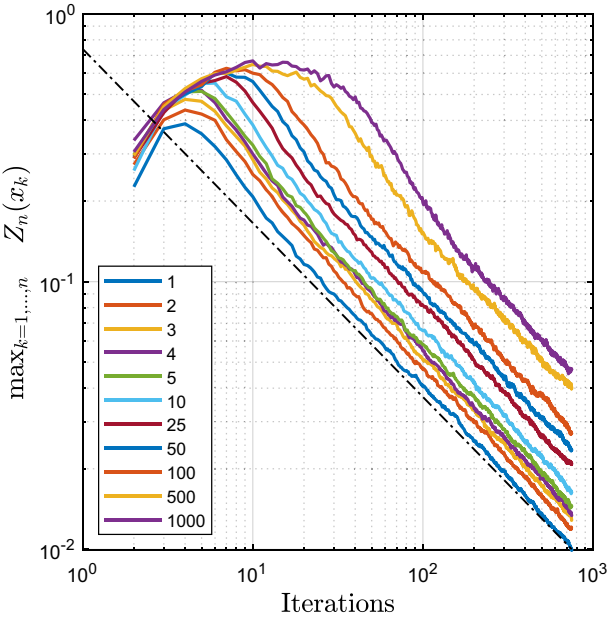

Fig. 20 Results for the median of $\max_{k=1,\ldots,n} Z_n(x_k)$ in setting (6) for different dimensions $d_o \in \{1, 2, \ldots, 1000\}$, indicated by different colors. As we conjectured, the asymptotic slope of all curves is equal, since $d_r = 1$ is fixed. As a point of reference, we added the graph of $n^{-0.65}$, represented by the dotted line

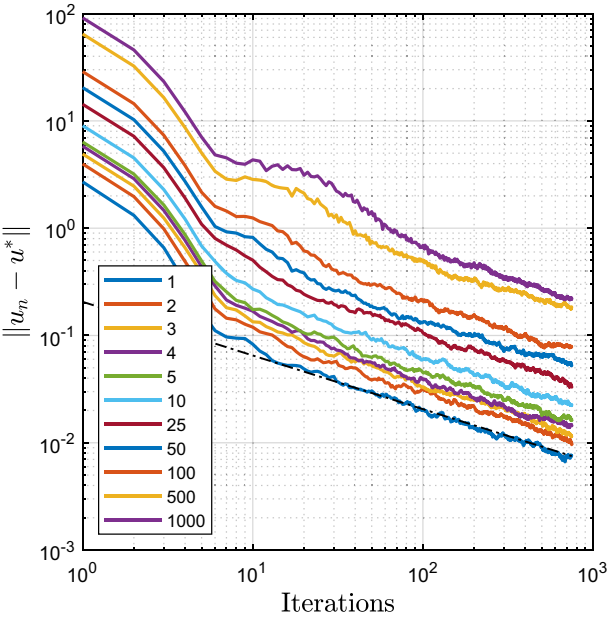

Fig. 21 Median distance to the optimal solution u^* during the course of the iterations for $d_o \in \{1, 2, \ldots, 1000\}$. Again, the asymptotic slope of all curves is equal and we added the line corresponding to $n^{-0.65}$ for comparison

be rewritten as

$$J(u) = \int_{\mathcal{X}} j(u,x)\mu(\mathrm{d}x) = \int_{\mathcal{X}_1} f_1\left(u, x, \int_{\mathcal{X}_2} f_2(u,y)\mu_2(\mathrm{d}y)\right)\mu_1(\mathrm{d}x).$$

Assume further, that $\mathcal{X}_1, \mathcal{X}_2, \mu_1, \mu_2, f_1$ and f_2 satisfy the corresponding equivalents of [2, Assumptions 2.1–2.4].

Now, we can independently calculate integration weights $(\beta_k)_{k=1,\dots,n}$ and $(\alpha_k)_{k=1,\dots,n}$ for the integrals over $\mathcal{X}_1$ and $\mathcal{X}_2$, respectively. The corresponding CSG approximations (indicated by hats) are then given by

$$\begin{aligned} f^{(n)} &:= \int_{\mathcal{X}_2} f_2(u,y)\mu_2(\mathrm{d}y) \approx \sum_{i=1}^{n} \alpha_i f_2(u_i, y_i) =: \hat{f}_n, \\ g^{(n)} &:= \int_{\mathcal{X}_2} \nabla_1 f_2(u,y)\mu_2(\mathrm{d}y) \approx \sum_{i=1}^{n} \alpha_i \nabla_1 f_2(u_i, y_i) =: \hat{g}_n, \\ \nabla J(u_n) &\approx \sum_{i=1}^{n} \beta_i \left(\nabla_1 f_1(u_i, x_i, \hat{f}_i) + \nabla_3 f_1(u_i, x_i, \hat{f}_i)\cdot \hat{g}_i\right) =: \hat{G}_n. \end{aligned}$$

The same steps as performed in the proof of [2, Lemma 4.6] yield the existence of a constant $C_1 > 0$, depending only on the Lipschitz constants of ∇f_1 and ∇f_2, such that

$$\begin{aligned} &\left\| \nabla J(u_n) - \hat{G}_n \right\| \\ &\le C_1\Big(d_W(\mu_1, \nu_n^{\beta}) + \sup_{x\in\mathcal{X}_1} \min_{k=1,\dots,n} \big(\|u_n - u_k\|_{\mathcal{U}} + \|x - x_k\|_{\mathcal{X}_1} + |\hat{f}_n - \hat{f}_k|\big)\Big). \end{aligned} \tag{7}$$

Here, ν_n^{β} corresponds to the measure related to the integration weights $(\beta_k)_{k=1,\dots,n}$, see [2, Assumption 2.4]. Now, denoting by $C_2 > 0$ a constant depending on the Lipschitz constant L_{f_2} of f_2, we decompose the last term:

$$\begin{aligned} &|\hat{f}_n - \hat{f}_k| \\ &\le |\hat{f}_n - f_n| + |\hat{f}_k - f_k| + |f_n - f_k| \\ &\le |\hat{f}_n - f_n| + |\hat{f}_k - f_k| + L_{f_2}\|u_n - u_k\|_{\mathcal{U}} \\ &\le C_2\Big(\|u_n - u_k\|_{\mathcal{U}} + \sup_{y\in\mathcal{X}_2} \min_{i=1,\dots,n} \left(\|u_n - u_i\|_{\mathcal{U}} + \|y - y_i\|_{\mathcal{X}_2}\right) \\ &\quad + \sup_{y\in\mathcal{X}_2} \min_{i=1,\dots,k} \left(\|u_k - u_i\|_{\mathcal{U}} + \|y - y_i\|_{\mathcal{X}_2}\right) + d_W(\mu_2, \nu_n^{\alpha}) + d_W(\mu_2, \nu_k^{\alpha})\Big) \\ &= C_2\Big(\|u_n - u_k\|_{\mathcal{U}} + \sup_{y\in\mathcal{X}_2} Z_n(y) + \sup_{y\in\mathcal{X}_2} Z_k(y) + d_W(\mu_2, \nu_n^{\alpha}) + d_W(\mu_2, \nu_k^{\alpha})\Big). \end{aligned} \tag{8}$$

Assuming that the convergence of the sequence $(u_n)_{n\in\mathbb{N}}$ generated by the CSG method implies

$$\mathcal{O}\left(\sup_{y\in\mathcal{X}_2} Z_n(y)\right) = \mathcal{O}\left(\sup_{y\in\mathcal{X}_2} Z_k(y)\right) \quad \text{and} \quad \mathcal{O}\big(d_W(\mu_2, \nu_n^\alpha)\big) = \mathcal{O}\big(d_W(\mu_2, \nu_k^\alpha)\big),$$

we insert (8) into (7), to obtain

$$\big\|\nabla J(u_n) - \hat{G}_n\big\| \leq C(C_1, C_2)\Big(d_W(\mu_1, \nu_n^\beta) + d_W(\mu_2, \nu_n^\alpha) + \sup_{x\in\mathcal{X}_1} Z_n(x) + \sup_{y\in\mathcal{X}_2} Z_n(y)\Big).$$

Therefore, by the same arguments as in Sect. 4.1, we conjecture

$$\begin{aligned}\big\|\nabla J(u_n) - \hat{G}_n\big\| &= \mathcal{O}\left(\ln(n)\cdot n^{-\frac{1}{\max\{2,\dim(\mathcal{X}_1),\dim(\mathcal{X}_2)\}}}\right),\\ \|u_{n+1} - u_n\|_{\mathcal{U}} &= \mathcal{O}\left(\ln(n)\cdot n^{-\frac{1}{\max\{2,\dim(\mathcal{X}_1),\dim(\mathcal{X}_2)\}}}\right).\end{aligned}$$

In conclusion, we conjecture that, assuming the objective function can be rewritten in terms of nested expectation values

$$J(u) = \int_{\mathcal{X}_1} f_1\left(u, x_1, \int_{\mathcal{X}_2} f_2\left(u, x_2, \int_{\mathcal{X}_3} f_3(\cdots)\mu_3(\mathrm{d}x_3)\right)\mu_2(\mathrm{d}x_2)\right)\mu_1(\mathrm{d}x_1),$$

the convergence rate of the CSG method depends only on the *largest* dimension of the occurring $\mathcal{X}_i$, which may be much lower when compared to $\dim(\mathcal{X})$.

Since this is again a claim and not a rigorous proof, we validate this assumption numerically. For this, we once more consider (5) and initialize it with 500 random starting points. This time, however, we utilize the fact that the objective function can be written as

$$J(u) = \frac{1}{2}\int_{\mathcal{X}} \|u - x\|_2^2 \mathrm{d}x = \frac{1}{2}\int_{\mathcal{X}} \Big(\sum_{i=1}^{d_\mathrm{r}} (u_i - x_i)^2\Big)\mathrm{d}x = \frac{1}{2}\sum_{i=1}^{d_\mathrm{r}} \int_{-\frac{1}{2}}^{\frac{1}{2}} (u_i - x_i)^2 \mathrm{d}x_i.$$

Thus, we can group the independent coordinates into subintegrals of arbitrary dimension, allowing us to study our claim for a large number of different regroupings without having to change the whole problem formulation. The results for several different decompositions and 500 random starting points in the case $d_\mathrm{r} = 100$ are shown in Fig. 22. The improved rates of convergence are clearly visible, independent on whether the subgroup dimensions are equal or not. As claimed above, the highest remaining dimension of integration determines the overall convergence rate of CSG.

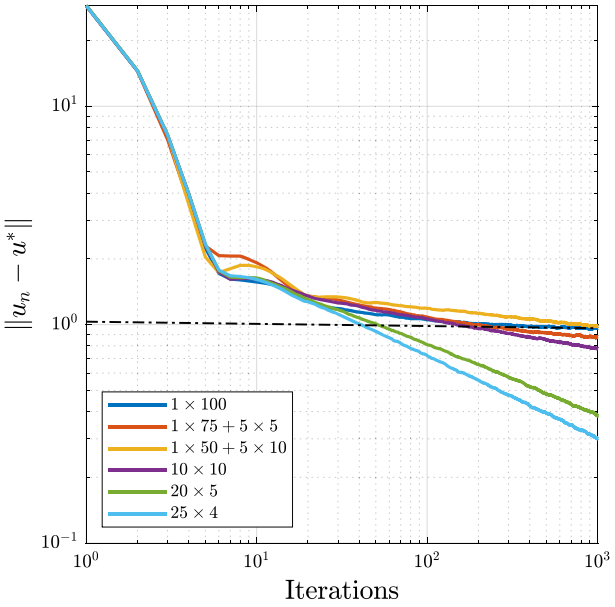

Fig. 22 Median total error $\|u_n - u^*\|_2$ of the CSG iterates for (5), for $d_\mathrm{r} = 100$. The integral over $\mathcal{X} = \left[-\frac{1}{2}, \frac{1}{2}\right]^{d_\mathrm{r}}$ has been decomposed into several integrals of smaller dimension. The labels in the bottom left give details about the decomposition, e.g., the orange line corresponds to splitting the whole integral into one integral of dimension 75 and 5 integrals of dimension 5. The dotted line indicates the expected rate of convergence obtained by the CSG method without splitting up the integral

5 Conclusion and outlook

In this contribution, we presented a numerical analysis of the CSG method. The practical performance of CSG was tested for two applications from nanoparticle design optimization with varying computational complexity. For the low-dimensional problem formulation, CSG was shown to perform superior when compared to the commercial *fmincon* blackbox solver. The high-dimensional setting provided an example, for which classic optimization schemes (stochastic as well as deterministic) from literature do not provide optimal solutions within reasonable time.

Convergence rates for CSG with constant step size were proposed and analytically motivated. They were shown to agree with numerically obtained convergence rates in several different instances. Moreover, in the case that the objective function admits additional structure, techniques to circumvent slow convergence for high dimensional integration domains were presented.

While the proposed convergence rates for CSG agree with our experimental results, it remains an open question if they can be proven rigorously. Furthermore, even though the choice of a metric for the nearest neighbor approximation in the integration weights is irrelevant for the convergence results, a problem specific metric could significantly improve the performance of CSG by exploiting additional structure, which might be lost by utilizing an arbitrary metric. How to automatically obtain such a metric during the optimization process requires further research.

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

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability statement In our numerical experiments related to convergence rates, only simple academic examples were used to visualize the theoretical results. These can be reproduced based on the given algorithms. For the nanoparticle design optimization, the corresponding data is available at https://doi.org/10.5281/zenodo.10032613.

Declaration

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. 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). https://doi.org/10.1007/s00158-020-02571-x
2. Grieshammer, M., Pflug, L., Stingl, M., Uihlein, A.: The continuous stochastic gradient method: part I–convergence theory. Comput. Optim. Appl. (2023). https://doi.org/10.1007/s10589-023-00542-8
3. Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. **22**, 400–407 (1951). https://doi.org/10.1214/aoms/1177729586
4. 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). https://doi.org/10.1007/s10107-016-1030-6
5. Zhao, Y., Xie, Z., Gu, H., Zhu, C., Gu, Z.: Bio-inspired variable structural color materials. Chem. Soc. Rev. **41**, 3297–3317 (2012). https://doi.org/10.1039/C2CS15267C
6. Wang, J., Sultan, U., Goerlitzer, E.S.A., Mbah, C.F., Engel, M.S., Vogel, N.: Structural color of colloidal clusters as a tool to investigate structure and dynamics. Adv. Funct. Mater. **30** (2019)
7. England, G.T., Russell, C., Shirman, E., Kay, T., Vogel, N., Aizenberg, J.: The optical Janus effect: asymmetric structural color reflection materials. Adv. Mater. **29** (2017). https://doi.org/10.1002/adma.201606876
8. Xiao, M., Hu, Z., Wang, Z., Li, Y., Tormo, A.D., Thomas, N.L., Wang, B., Gianneschi, N.C., Shawkey, M.D., Dhinojwala, A.: Bioinspired bright noniridescent photonic melanin supraballs. Sci. Adv. **3**(9), 1701151 (2017). https://doi.org/10.1126/sciadv.1701151
9. Goerlitzer, E.S.A., Klupp-Taylor, R.N., Vogel, N.: Bioinspired photonic pigments from colloidal self-assembly. Adv. Mater. **30**(28), 1706654 (2018). https://doi.org/10.1002/adma.201706654
10. Uihlein, A., Pflug, L., Stingl, M.: Optimizing color of particulate products. PAMM **22**(1), 202200047 (2023). https://doi.org/10.1002/pamm.202200047
11. Taylor, R.K., Seifrt, F., Zhuromskyy, O., Peschel, U., Leugering, G., Peukert, W.: Painting by numbers: Nanoparticle-based colorants in the post-empirical age. Adv. Mater. **23**(22–23), 2554–2570 (2011). https://doi.org/10.1002/adma.201100541
12. Buxbaum, G.: Industrial inorganic pigments. Wiley, New Jersey (2008) https://doi.org/10.1002/3527603735
13. Colorimetry, C.: Report no: Cie pub no 15. CIE Central Bureau, Vienna (2004)
14. CIE Commission Internationale de l'Éclairage Proceedings (1931)

15. Mishchenko, M.I., Travis, L.D., Lacis, A.A.: Scattering, Absorption, and Emission of Light by Small Particles. Cambridge University Press, Cambridge (2002)
16. DeVore, J.R.: Refractive indices of rutile and sphalerite. J. Opt. Soc. Am. **41**(6), 416–419 (1951). https://doi.org/10.1364/JOSA.41.000416
17. Purcell, E.M., Pennypacker, C.R.: Scattering and absorption of light by nonspherical dielectric grains. Astrophys. J. **186**, 705–714 (1973). https://doi.org/10.1086/152538
18. Yurkin, M.A., Hoekstra, A.G.: The discrete-dipole-approximation code ADDA: capabilities and known limitations. J. Quant. Spectrosc. Radiat. Transfer **112**(13), 2234–2247 (2011). https://doi.org/10.1016/j.jqsrt.2011.01.031
19. Nees, N., Pflug, L., Mann, B., Stingl, M.: Multi-material design optimization of optical properties of particulate products by discrete dipole approximation and sequential global programming. Struct. Multidiscip. Optim. (2022). https://doi.org/10.1007/s00158-022-03376-w
20. Mie, G.: Beiträge zur optik trüber medien, speziell kolloidaler metallösungen. Ann. Phys. **330**, 377–445 (1908). https://doi.org/10.1002/andp.19083300302
21. Hergert, W., Wriedt, T.: The Mie Theory: Basics and Applications. Springer Series in Optical Science. Springer, Berlin (2012). https://doi.org/10.1007/978-3-642-28738-1
22. Kubelka, P., Munk, F.: An article on optics of paint layers. Z. Tech. Phys. **12**(593–601), 259–274 (1931)
23. García-Valenzuela, A., Cuppo, F., Olivares, J.: An assessment of saunderson corrections to the diffuse reflectance of paint films. In: Journal of Physics: Conference Series, vol. 274, p. 012125 (2011). https://doi.org/10.1088/1742-6596/274/1/012125. IOP Publishing
24. on Illumination (CIE), I.C.: CIE 1964 colour-matching functions , 10 degree observer. International Commission on Illumination (CIE). https://doi.org/10.25039/cie.ds.sqksu2n5
25. Wiscombe, W.J.: Improved mie scattering algorithms. Appl. Opt. **19**(9), 1505–1509 (1980)
26. Wang, M., Fang, E.X., Liu, H.: Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. Math. Program. 161(1-2, Ser. A), 419–449 (2017). https://doi.org/10.1007/s10107-016-1017-3
27. Schäfer, J., Lee, S.-C., Kienle, A.: Calculation of the near fields for the scattering of electromagnetic waves by multiple infinite cylinders at perpendicular incidence. J. Quant. Spectrosc. Radiat. Transfer **113**(16), 2113–2123 (2012). https://doi.org/10.1016/j.jqsrt.2012.05.019
28. Draine, B.T., Flatau, P.J.: Discrete-dipole approximation for scattering calculations. JOSA A **11**(4), 1491–1499 (1994)
29. Sigmund, O.: Morphology-based black and white filters for topology optimization. Struct. Multidiscip. Optim. **33**(4), 401–424 (2007). https://doi.org/10.1007/s00158-006-0087-x
30. Caflisch, R.E.: Monte carlo and quasi-monte carlo methods. Acta Numer. **7**, 1–49 (1998). https://doi.org/10.1017/S0962492900002804
31. Burrough, P., McDonnell, R., Lloyd, C.: 8.11 nearest neighbours: Thiessen (dirichlet/voroni) polygons. Princ. Geograph. Inf. Syst. (2015)
32. Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. SIAM Rev. **60**(2), 223–311 (2018). https://doi.org/10.1137/16M1080173
33. Fournier, N., Guillin, A.: On the rate of convergence in wasserstein distance of the empirical measure. Probab. Theory Relat. Fields **162**(3), 707–738 (2015). https://doi.org/10.1007/s00440-014-0583-7
34. Beck, A.: First-order Methods in Optimization. MOS-SIAM Series on Optimization, vol. 25, p. 475. Society for Industrial and Applied Mathematics (SIAM): Mathematical Optimization Society, Philadelphia (2017). https://doi.org/10.1137/1.9781611974997.ch1