Journal of Mathematical Imaging and Vision (2025) 67:7
https://doi.org/10.1007/s10851-024-01215-9

A Structured L-BFGS Method with Diagonal Scaling and Its Application to Image Registration

Florian Mannel[1] · Hari Om Aggrawal[1]

Received: 1 May 2024 / Accepted: 6 November 2024 / Published online: 30 December 2024

Abstract

We devise an L-BFGS method for optimization problems in which the objective is the sum of two functions, where the Hessian of the first function is computationally unavailable while the Hessian of the second function has a computationally available approximation that allows for cheap matrix–vector products. This is a prototypical setting for many inverse problems. The proposed L-BFGS method exploits the structure of the objective to construct a more accurate Hessian approximation than in standard L-BFGS. In contrast with existing works on structured L-BFGS, we choose the first part of the seed matrix, which approximates the Hessian of the first function, as a diagonal matrix rather than a multiple of the identity. We derive two suitable formulas for the coefficients of the diagonal matrix and show that this boosts performance on real-life image registration problems, which are highly non-convex inverse problems. The new method converges globally and linearly on non-convex problems under mild assumptions in a general Hilbert space setting, making it applicable to a broad class of inverse problems. An implementation of the method is freely available.

Keywords Structured L-BFGS · Seed matrix · Diagonal scaling · Non-convex optimization · Inverse problems · Medical image registration

Mathematics Subject Classification 65J22 · 65K05 · 65K10 · 90C06 · 90C26 · 90C30 · 90C48 · 90C53 · 90C90

1 Introduction

1.1 A Structured L-BFGS Method with Diagonal Scaling

In this paper, we study a new L-BFGS-type method for unconstrained optimization problems

$$\min_{x\in\mathcal{X}} \mathcal{J}(x)$$

with a cost function of the form

$$\mathcal{J}:\mathcal{X}\to\mathbb{R},\qquad \mathcal{J}(x)=\mathcal{D}(x)+\mathcal{S}(x). \tag{1}$$

Among others, this is a prototypical setting for inverse problems, where $\mathcal{D}:\mathcal{X}\to\mathbb{R}$ represents a data-fitting term, $\mathcal{S}:\mathcal{X}\to\mathbb{R}$ a regularizer, and $\mathcal{X}$ a Hilbert space. In this setting, it is often the case that, after discretization, the data-fitting term has an ill-conditioned and dense Hessian for which even matrix–vector multiplications are computationally expensive, whereas the Hessian of the regularizer is positive definite, well-conditioned, and sparse with computationally cheap matrix–vector products. The L-BFGS method [19, 48, 56] is one of the most widely used algorithms for large-scale inverse problems, but it does not take advantage of the splitting in (1). In our recent work [50], we introduced an L-BFGS-type method called TULIP (structured L-BFGS method for inverse problems) that exploits different structural properties of the two terms in (1). We proved the method's global and linear convergence including for the case that the cost function $\mathcal{J}$ is non-convex with singular Hessian, and we demonstrated in numerical experiments that it outperforms standard L-BFGS as well as other structured L-BFGS-type methods on real-world image registration problems.

✉ Florian Mannel
florian.mannel@uni-luebeck.de

Hari Om Aggrawal
hariom85@gmail.com

[1] Institute of Mathematics and Image Computing, University of Lübeck, Maria-Goeppert-Straße 3, 23562 Lübeck, Germany

In this paper, we present ROSE (structured L-BFGS method with flexible seed matrix), an algorithm that further improves the numerical performance of TULIP for image registration problems while offering identical convergence guarantees.

The improvement, which we demonstrate in the numerical experiments in Sect. 4, can be attributed primarily to the fact that in ROSE the Hessian of $\mathcal{D}$ is approximated in the seed matrix by a diagonal matrix rather than a multiple of the identity. Let us explain this in more detail.

In the classical L-BFGS method with memory length $\ell \in \mathbb{N}_0$, the L-BFGS operator $B_k \approx \nabla^2 \mathcal{J}(x_k)$ is obtained as $B_k := B_k^{(\ell)}$ from the recursion

$$B_k^{(j+1)} := B_k^{(j)} + \mathrm{Update}\big(s_{m+j}, y_{m+j}, B_k^{(j)}\big), \qquad j = 0, \ldots, \ell - 1.$$

Here, $m := \max\{0, k - \ell\}$, the stored update vectors are $\{(s_j, y_j)\}_{j=m}^{k-1}$, where $s_j := x_{j+1} - x_j$ and $y_j := \nabla \mathcal{J}(x_{j+1}) - \nabla \mathcal{J}(x_j)$, and they satisfy $y_j^T s_j > 0$ for all j. For $(s, y) \in \mathcal{X} \times \mathcal{X}$ with $y^T s > 0$ and positive definite B, the update is given by

$$\mathrm{Update}\big(s, y, B\big) := \frac{yy^T}{y^T s} - \frac{Bss^T B^T}{s^T Bs}.$$

It is a great advantage of the classical L-BFGS method that if the seed matrix $B_k^{(0)}$ is a multiple of the identity $\tau_k I$ for some $\tau_k > 0$, then the search direction $d_k = -B_k^{-1} \nabla \mathcal{J}(x_k)$ can be computed matrix-free and without having to solve a linear system. In practice, this is efficiently realized through the *two-loop recursion* (e.g., [15, Fig. 1], [57, Algorithm 7.4]), enabling the use of L-BFGS for large-scale problems. On the other hand, this choice of the seed matrix does not take into account the structure (1) and, in turn, does not benefit from the convenient properties of $\nabla^2 \mathcal{S}(x_k)$. To change this, in TULIP the seed matrix in iteration k is taken to be

$$B_k^{(0)} = \tau_k I + S_k,$$

where S_k approximates $\nabla^2 \mathcal{S}(x_k)$ and is selected in such a way that $B_k^{(0)}$ is positive definite and linear systems involving $B_k^{(0)}$ can be solved cheaply at least approximately. While the choice $B_k^{(0)} = \tau_k I + S_k$ is expected to make B_k a better approximation of $\nabla^2 \mathcal{J}(x_k)$ and improve the rate of convergence, the computation of $d_k = -B_k^{-1} \nabla \mathcal{J}(x_k)$ now requires the solution of a linear system involving $B_k^{(0)}$. In this paper, we consider seed matrices of the more general form

$$B_k^{(0)} = D_k + S_k, \tag{2}$$

where D_k and S_k are chosen in such a way that $B_k^{(0)}$ is positive definite and linear systems involving $B_k^{(0)}$ can be solved cheaply at least approximately. It is clear that this generalizes TULIP. The main focus in this paper is on the choice of D_k as a diagonal matrix. In particular, we propose two formulas for the entries of the diagonal matrix in Sect. 2 and we compare them numerically in Sect. 4. We also use the convergence theory developed in [50] for TULIP to obtain convergence results for ROSE in Sect. 3.

In view of the structure (1), the matrices $\tau_k I$ and S_k, respectively, D_k and S_k, may be regarded as approximations of the Hessians $\nabla^2 \mathcal{D}(x_k)$ and $\nabla^2 \mathcal{S}(x_k)$, respectively. It is therefore expected that the approximation quality of $B_k \approx \nabla^2 \mathcal{J}(x_k)$ increases from L-BFGS to TULIP to ROSE. Consequently, L-BFGS should usually require more iterations than TULIP, which should require more iterations than ROSE. Since the increase in approximation quality in the structured methods TULIP and ROSE comes at the cost of (inexactly) solving one linear system per iteration, the question arises whether the structured approach actually lowers the run time in comparison with standard L-BFGS. In [50], the answer was affirmative for TULIP when we considered a test set of 22 real-world problems from medical image registration. In the present paper, we find that ROSE is significantly faster than TULIP on the same set of test problems. These problems are large-scale and highly non-convex inverse problems that involve various data-fitting terms and regularizers, see Sect. 1.2, suggesting that ROSE is a promising method also for other inverse problems.

1.2 Application to Image Registration

Image registration is a highly ill-posed inverse problem where the regularizer $\mathcal{S}$ plays an important role to estimate a meaningful transformation field to align the images. These regularizers are generally designed well-structured and have a Hessian that is cheap to compute. Examples include a quadratic first order [16], a quadratic second order [29] and a non-quadratic first order [18] regularizer, commonly referred to as elastic, curvature and hyperelastic regularizer, respectively. We emphasize that the hyperelastic regularizer is non-convex. On the other hand, the data-fidelity measures $\mathcal{D}$ are highly non-convex and their Hessians are very dense and ill-conditioned. The sum of squared differences, mutual information [64] and normalized gradient fields [34] are the most commmonly used fidelity measures for registration. The structure of the Hessian in the image registration problems motivated us to apply structured L-BFGS methods including ROSE. In the numerical experiments in Sect. 4, we consider 22 real-world image registration problems that use the aforementioned fidelity measures and regularizers to register mono-modal and multi-modal images, estimating small to large deformations. While landmark constrained registration problems are not included in the 22 test cases, they still fit in the framework of this paper. For further details we refer the reader to Sect. 4.

1.3 Related Work

The idea to exploit the problem structure for constructing a better approximation of $\nabla^2 \mathcal{J}(x_k)$ and use it as a seed matrix in L-BFGS appears in the numerical studies [1, 2, 17, 35, 38, 40, 49, 50, 54, 66] that address a wide range of real-life problems. On the considered large-scale problems, the authors report significant speed ups over all methods that are used for comparison, including standard L-BFGS, Gauss–Newton, and truncated Newton. Among those contributions, only [49, 66] employ a diagonal seed matrix other than a scaled identity, but they do not consider the structure (1). Thus, to the best of our knowledge, Algorithm ROSE is the first structured L-BFGS method with proper diagonal scaling. Another important difference to the present paper is that except for [50], convergence rates are only studied numerically.

The convergence results of this work also hold for memory size zero. In this case, no updates are applied; hence, $B_k = B_k^{(0)} = D_k + S_k$. For the choice $S_k = 0$, this may be regarded as a method with *diagonal Barzilai–Borwein step size* [60]. Again, however, it seems that this work is the first that integrates diagonal Barzilai–Borwein step sizes into a *structured* method. We emphasize that the two choices proposed for D_k in Sect. 2 are inspired by [60], but are still somewhat different. As a matter of fact, our proposals do not agree with any of the choices for diagonal scaling that we have found in the literature.

For $\mathcal{S} \equiv 0$ and $S_k = 0$ for all k, ROSE is an unstructured L-BFGS method that uses a diagonal seed matrix. This class of methods has been studied in [9, 14, 21, 31, 43, 46, 48, 53, 61, 63], but convergence is usually shown for strongly convex objectives or not at all, except in [46], where global convergence is proved for the non-convex case in the sense that $\liminf_{k\to\infty} \|\nabla \mathcal{J}(x_k)\| = 0$. In contrast, we have $\lim_{k\to\infty} \|\nabla \mathcal{J}(x_k)\| = 0$ in that case and a linear rate of convergence, cf. Theorems 3.3,3.7 and 3.9. That is, in addition to its contribution in the structured setting, which is the main focus of this work, ROSE also provides convergence guarantees that are stronger than existing ones for unstructured objectives. Non-diagonal seed matrices have also been considered, for instance in [3], where the seed matrix itself contains low-rank updates.

In *diagonal quasi-Newton methods*, the Hessian is approximated by a diagonal matrix. In contrast with an L-BFGS-based approach, however, low-rank updates are not applied to the diagonal matrix. References include [5, 7, 8, 28, 44, 45, 68], but we are not aware of works that embed this approach in a structured method.

Despite the large amount of works that involve diagonal matrices, it seems that one of the two choices that we propose for D_k in [2] has not been considered before, while the other one may be regarded as a generalization of the diagonal matrix that appears in [6] within an unstructured diagonal quasi-Newton method. As mentioned above, however, there are no low-rank updates applied to the diagonal matrix in [6], which differs from our approach. The numerical experiments in [6] show that the method requires less CPU time than BFGS, but more than L-BFGS. On the theoretical side, [6] proves global convergence for strictly convex quadratics.

Structured variants of *full memory* quasi-Newton methods have been studied in various settings, cf. e.g., [4, 22, 24, 25, 27, 37, 42, 51, 52, 65], most often in the context of least squares problems [22, 32, 36, 55, 67]. However, they do not allow for a seed matrix, so by design they are somewhat different from Algorithm ROSE.

1.4 Main Contributions

The main contributions of this paper are that

- we present ROSE, the first structured L-BFGS method with diagonal scaling. Additionally, the specific diagonal scaling that we propose is new even for unstructured L-BFGS;
- we obtain global and linear convergence of ROSE for non-convex problems without assuming invertibility of the Hessian, cf. Theorems 3.3 and 3.7. Such strong results are not available for other structured L-BFGS methods except for our own method TULIP [50];
- we show that ROSE outperforms TULIP on real-world image registration problems. This is significant since TULIP outperforms standard L-BFGS as well as competing structured L-BFGS methods on the same set of problems [50];
- we work in Hilbert space, which is rarely done for L-BFGS and also for diagonal scaling. This is valuable for instance because infinite-dimensional Hilbert spaces are a natural setting for many inverse problems.

1.5 Code Availability

An implementation of our structured L-BFGS method that includes an example from the numerical section of this paper is freely available at https://github.com/hariagr/SLBFGS.

1.6 Organization and Notation

The paper is organized as follows. In Sect. 2, we introduce ROSE. Section 3 collects the convergence results and Sect. 4

contains the numerical experiments. Conclusions are drawn in Sect. 5.

We use $\mathbb{N} = \{1, 2, 3, \ldots\}$ and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. The scalar product of $x, y \in \mathcal{X}$ is indicated by $x^T y$, and for $x \in \mathcal{X}$ the linear functional $y \mapsto x^T y$ is denoted by x^T. The induced norm is $\|x\|$. We write $M \in \mathcal{L}(\mathcal{X})$ if $M : \mathcal{X} \to \mathcal{X}$ is a bounded linear mapping. The notation $M \in \mathcal{L}_{\ge 0}(\mathcal{X})$ means that $M \in \mathcal{L}(\mathcal{X})$ is symmetric and positive semi-definite. For $M \in \mathcal{L}(\mathcal{X})$, we define

$$\lambda(M) := \inf_{\|v\|=1} \|Mv\| \qquad \text{and} \qquad \Lambda(M) := \sup_{\|v\|=1} \|Mv\|.$$

Apparently, we have $0 \le \lambda(M) \le \Lambda(M)$. If $M \in \mathcal{L}_{\ge 0}(\mathcal{X})$, then $\lambda(M) = \inf_{\|v\|=1} v^T M v$ and $\Lambda(M) = \|M\| = \sup_{\|v\|=1} v^T M v$. Furthermore, if $M \in \mathcal{L}_{\ge 0}(\mathcal{X})$ is positive definite, then it is invertible, M^{-1} is symmetric, and $\lambda(M^{-1}) = \Lambda(M)^{-1} > 0$ as well as $\Lambda(M^{-1}) = \lambda(M)^{-1} > 0$. If $\mathcal{X}$ is finite dimensional, then $\lambda(M)$ and $\Lambda(M)$ are the smallest and largest eigenvalue of $M \in \mathcal{L}_{\ge 0}$.

2 The Structured L-BFGS method ROSE

Algorithm ROSE: Structured inverse L-BFGS method with flexible seed matrix

```
   Input: $x_0 \in \mathcal{X}$, $\epsilon \ge 0$, $\ell \in \mathbb{N}_0$, $c_0 \ge 0$, $C_0 \in [c_0, \infty]$,
          $c_s, c_1, c_2 > 0$
1  Choose $S_0 \in \mathcal{L}_{\ge 0}(\mathcal{X})$
2  Let $D_0 = \tau I$ for some $\tau > 0$
3  for $k = 0, 1, 2, \ldots$ do
4      Let $m := \max\{0, k - \ell\}$
5      Let $B_k^{(0)} := D_k + S_k$   // choice of seed matrix
6      Compute $d_k := -B_k^{-1} \nabla \mathcal{J}(x_k)$ from $B_k^{(0)}$ and the stored pairs
       $\{(s_j, y_j)\}_{j=m}^{k-1}$ using the two-loop recursion [57,
       Algorithm 7.4]
7      Compute step length $\alpha_k > 0$ using a line search
8      Let $s_k := \alpha_k d_k$, $x_{k+1} := x_k + s_k$, $y_k := \nabla \mathcal{J}(x_{k+1}) - \nabla \mathcal{J}(x_k)$
9      if $y_k^T s_k > c_s \|s_k\|^2$ then append $(s_k, y_k)$ to storage
       // cautious update 1
10     if $k \ge \ell$ then remove $(s_m, y_m)$ from storage
11     if $\|\nabla \mathcal{J}(x_{k+1})\| \le \epsilon$ then output $x_{k+1}$ and break
12     Choose $S_{k+1} \in \mathcal{L}_{\ge 0}(\mathcal{X})$
13     Let $z_k := y_k - S_{k+1} s_k$
14     Let $\omega_{k+1}^l := \min\{c_0, c_1 \|\nabla \mathcal{J}(x_{k+1})\|^{c_2}\}$ and
       $\omega_{k+1}^u := \max\{C_0, (c_1 \|\nabla \mathcal{J}(x_{k+1})\|^{c_2})^{-1}\}$
15     if $z_k^T s_k > 0$ then let $T_{k+1} := \left[\omega_{k+1}^l, \omega_{k+1}^u\right]$ else let
       $T_{k+1} := \left[\omega_{k+1}^l, P_{k+1}(\tau_{k+1}^g)\right]$
16     Choose $D_{k+1} \in \mathcal{L}_{\ge 0}(\mathcal{X})$ such that
       $\lambda(D_{k+1}), \Lambda(D_{k+1}) \in T_{k+1}$   // cautious update 2
17 end
```

The proposed structured L-BFGS algorithm with seed matrix $B_k^{(0)} = D_k + S_k$ is summarized as Algorithm ROSE (Structured inverse L-BFGS method with flexible seed matrix).

Here, $P_{k+1} : \mathbb{R} \to \mathbb{R}$ projects onto $[\omega_{k+1}^l, \omega_{k+1}^u]$, i.e., $P_{k+1}(t) := \min\{\omega_{k+1}^u, \max\{\omega_{k+1}^l, t\}\}$. The scalar τ_{k+1}^g is defined in Definition 2.2.

The main differences between Algorithm ROSE and standard L-BFGS are the choice of the seed matrix $B_k^{(0)}$ and the two cautious updates that affect if (s_k, y_k) enters the storage and that restrict the choice of D_{k+1}. We stress that the two-loop recursion in Line 6 requires solving a linear system whose system matrix is $B_k^{(0)}$. In standard L-BFGS, where $B_k^{(0)} = \tau_k I$, this is a trivial task, but in ROSE, where $B_k^{(0)} = D_k + S_k$, this is usually non-trivial. As discussed in Sect. 1.1, it is therefore important to choose D_k and S_k in such a way that linear systems with $B_k^{(0)}$ are relatively cheap to solve either directly or iteratively. In the numerical experiments, we choose D_k as a diagonal matrix with (sufficiently) positive entries, so for many regularizers $\mathcal{S}$ the choice $S_k = \nabla^2 \mathcal{S}(x^k)$ will result in linear systems that are cheap to solve. In Line 16, we see that the interval T_{k+1} is used to restrict the spectrum of D_{k+1}, which is called *cautious updating*. Cautious updating is critical for the strong convergence properties of ROSE; cf. Section 2.2 for details.

In comparison with Algorithm TULIP from [50], ROSE is more general. Specifically, the interval T_{k+1} is larger in ROSE and D_{k+1} in TULIP is restricted to multiples of the identity. In the following subsections, we offer two choices for D_{k+1} and provide further commentary on ROSE.

2.1 Choice of D_k

A key element of ROSE is to use

$$B_{k+1}^{(0)} = D_{k+1} + S_{k+1} \tag{3}$$

as seed matrix, where D_{k+1} is a symmetric positive semi-definite operator. In classical L-BFGS, we have $D_{k+1} = \tau_{k+1} I$ for some $\tau_{k+1} > 0$ and $S_{k+1} = 0$ for all k, with $\tau_{k+1} = y_k^T s_k / \|y_k\|^2$ being the most popular choice. This choice as well as some others can be derived from the Oren–Luenberger scaling strategy [58], which postulates that $B_{k+1}^{(0)}$ should satisfy the secant equation $y_k = B_{k+1}^{(0)} s_k$ in a least squares sense. In the structured setting of this paper, where $B_{k+1}^{(0)} = D_{k+1} + S_{k+1}$, the secant equation reads

$$D_{k+1} s_k - z_k = 0, \qquad \text{where} \qquad z_k := y_k - S_{k+1} s_k. \tag{4}$$

If D_{k+1} is invertible, this equation has equivalent forms such as $D_{k+1}^{1/2} s_k - D_{k+1}^{-1/2} z_k = 0$ or $s_k - D_{k+1}^{-1} z_k = 0$. This motivates to choose D_{k+1} in such a way that it minimizes the associated least squares problem, e.g., $\|D_{k+1} s_k - z_k\|$ or $\|D_{k+1}^{1/2} s_k - D_{k+1}^{-1/2} z_k\|$. On the other hand, it is well known that to maintain positive definiteness of B_{k+1}, the seed matrix $B_{k+1}^{(0)}$ has to be positive definite. In fact, to prove convergence of the algorithm, the positive definiteness of B_{k+1} is not strong enough; it is also necessary to appropriately control the condition number of B_{k+1} so that it does not go to infinity too quickly. For the problems that we are interested in, it is reasonable to expect that the approximations S_{k+1} of the Hessians $\nabla^2\mathcal{S}(x_{k+1})$ have well-behaved condition numbers. In this setting, due to Line 9, the condition number of B_{k+1} can be controlled by ensuring that $\lambda(D_{k+1})$ and $\Lambda(D_{k+1})$ belong to an appropriate interval, denoted T_{k+1} in ROSE, cf. Line 15. We comment further on T_{k+1} when we discuss cautious updating in Sect. 2.2.

As it turns out, the convergence analysis of [50] goes through for Algorithm ROSE without further specification of D_{k+1}. That is, the convergence results that we obtain hold for any $D_{k+1} \in \mathcal{L}_{\geq 0}(\mathcal{X})$ satisfying $\lambda(D_{k+1}), \Lambda(D_{k+1}) \in T_{k+1}$. However, to make the method efficient in practical computations it is crucial that linear systems with $B_{k+1}^{(0)}$ can be solved, at least approximately, in an efficient way. Thus, although the convergence analysis does not require a specific structure of D_{k+1}, we are mainly interested in choices that lead to a seed matrix $B_{k+1}^{(0)}$ with favorable properties (for iterative solvers). Recall that we focus on the case where S_{k+1} is (some sort of approximation of) a regularizer and hence positive definite, well-conditioned, and cheap to evaluate in any given direction. It is then clear that we can choose D_{k+1} as any symmetric positive semi-definite operator that is cheap to evaluate in all directions. In [50] we focused on $D_{k+1} = \tau_{k+1}I$, which is the most common choice for classical L-BFGS, but here we consider the more general situation that D_{k+1} is a *diagonal operator* with respect to a fixed orthonormal basis $(e_j)_{j\in\mathcal{I}} \subset \mathcal{X}$, where $\mathcal{I} = \{1, 2, \ldots, n\}$ if $\mathcal{X}$ is n dimensional and $\mathcal{I} = \mathbb{N}$ if it is infinite dimensional. We say that $D_{k+1} \in \mathcal{L}(\mathcal{X})$ is diagonal with respect to $(e_j)_{j\in\mathcal{I}}$ if there is a sequence $(\gamma_j^{k+1}) \in \ell^\infty(\mathcal{I})$ such that $D_{k+1} = \sum_{j\in\mathcal{I}} \gamma_j^{k+1} e_j e_j^T$. In the following, we will usually suppress the k-dependency of the coefficients (γ_j^{k+1}) and write (γ_j) instead. From $\|D_{k+1}\| = \sup_{j\in\mathcal{I}} \gamma_j$ we infer that D_{k+1} is bounded if and only if $\sup_{j\in\mathcal{I}} \gamma_j < \infty$. Moreover, D_{k+1} is positive semi-definite if and only if $\gamma_j \geq 0$ for all $j \in \mathcal{I}$. As $\lambda(D_{k+1}) = \inf_j \gamma_j$ and $\Lambda(D_{k+1}) = \sup_j \gamma_j$, it is easy to ensure $\lambda(D_{k+1}) \in T_{k+1}$ and $\Lambda(D_{k+1}) \in T_{k+1}$ in Line 16 of Algorithm ROSE. If we set $\gamma_j = \tau_{k+1}$ for all j, we recover the classical choice $D_{k+1} = \tau_{k+1}I$. Next we provide two possible choices for the diagonal elements (γ_j) of $D_{k+1} = \sum_{j\in\mathbb{N}} \gamma_j e_j e_j^T$.

The first choice for D_{k+1} As outlined above, we want to minimize the residual $\|D_{k+1}s_k - z_k\|$. This leads to $\gamma_j = \frac{z_k^T e_j}{s_k^T e_j}$ if $s_k^T e_j \neq 0$, and γ_j arbitrary if $s_k^T e_j = 0$. Since we also want to ensure $\lambda(D_{k+1}), \Lambda(D_{k+1}) \in T_{k+1}$, cf. Line 16 in ROSE, we project each γ_j onto T_{k+1}. This is equivalent to saying that (γ_j) minimizes the constrained least squares problem $\|D_{k+1}s_k - z_k\|$ s.t $\lambda(D_{k+1}) \in T_{k+1} \wedge \Lambda(D_{k+1}) \in T_{k+1}$. Writing $\hat{P}_{k+1} : \mathbb{R} \to \mathbb{R}$ for the projection onto T_{k+1} this yields

$$\begin{cases} \gamma_j = \hat{P}_{k+1}\left(\frac{z_k^T e_j}{s_k^T e_j}\right) & \text{if } s_k^T e_j \neq 0, \\ \gamma_j \in T_{k+1} & \text{if } s_k^T e_j = 0. \end{cases} \tag{5}$$

Of course, in finite dimensions with the canonical basis $(e_j)_j$, the scalar products $z_k^T e_j$ and $s_k^T e_j$ are simply the j-th component of the vectors z_k and s_k, respectively.

The second choice for D_{k+1} Next we determine the coefficients (γ_j) that minimize $\|D_{k+1}^{1/2}s_k - D_{k+1}^{-1/2}z_k\|$ s.t. $\lambda(D_{k+1}) \in T_{k+1} \wedge \Lambda(D_{k+1}) \in T_{k+1}$. Since $\|D_{k+1}^{1/2}s_k - D_{k+1}^{-1/2}z_k\|$ is minimal for $\gamma_j = \left|\frac{z_k^T e_j}{s_k^T e_j}\right|$ if $s_k^T e_j \neq 0$, and γ_j arbitrary if $s_k^T e_j = 0$, the coefficients (γ_j) are optimal if

$$\begin{cases} \gamma_j = \hat{P}_{k+1}\left(\left|\frac{z_k^T e_j}{s_k^T e_j}\right|\right) & \text{if } s_k^T e_j \neq 0, \\ \gamma_j \in T_{k+1} & \text{if } s_k^T e_j = 0. \end{cases} \tag{6}$$

The following relation between the coefficients of (5) and (6) is obvious.

Lemma 2.1 *Let $(s_k, z_k) \in \mathcal{X} \times \mathcal{X}$, $(e_j)_{j\in\mathcal{I}} \subset \mathcal{X}$ for some $\mathcal{I} \subset \mathbb{N}$, and $T_{k+1} \subset \mathbb{R}$. Define $(\gamma_j)_{j\in\mathcal{I}}$ according to (5) and $(\hat{\gamma}_j)_{j\in\mathcal{I}}$ according to (6), choosing $\gamma_j \leq \hat{\gamma}_j$ for any $j \in \mathcal{I}$ with $s_k^T e_j = 0$. Then $\gamma_j \leq \hat{\gamma}_j$ for all $j \in \mathcal{I}$.*

Relationship to scaled identity

For $D_{k+1} = \tau_{k+1}I$, the following three scaling factors are particularly interesting, and they also play a role in this paper.

Definition 2.2 For $(s_k, z_k) \in \mathcal{X} \times \mathcal{X}$ with $s_k \neq 0$ let $\rho_k := z_k^T s_k$ and define

$$\tau_{k+1}^s := \frac{\rho_k}{\|s_k\|^2}, \qquad \tau_{k+1}^g := \frac{\|z_k\|}{\|s_k\|}, \qquad \tau_{k+1}^z := \frac{\|z_k\|^2}{\rho_k},$$

where τ_{k+1}^z is only defined if $\rho_k \neq 0$.

We remark that τ_{k+1}^s and τ_{k+1}^z are the so-called Barzilai–Borwein scaling factors introduced by Barzilai and Borwein in their seminal work [11].

It is easy to check that for $\rho_k > 0$ the scaling factors τ_{k+1}^s, τ_{k+1}^z and τ_{k+1}^g are the minimizers of the least squares problems $\|\tau s_k - z_k\|$, $\|\sqrt{\tau}s_k - z_k/\sqrt{\tau}\|$ and $\|s_k - z_k/\tau\|$, respectively. Correspondingly, there holds $0 < \tau_{k+1}^s \leq \tau_{k+1}^g \leq \tau_{k+1}^z$ if $\rho_k > 0$. Note that the least squares problems are identical to those outlined for (4) if $D_{k+1} = \tau_{k+1}I$.

Thus, the coefficients (γ_j) from (5) and (6) as well as those introduced in 2.2 all realize optimal least squares fits of s_k to z_k, but each in a different sense. In doing so, they all approximately realize the secant equation $D_{k+1}s_k = z_k$. The secant equation plays a fundamental role in quasi-Newton methods [23, 57].

It follows that the choices (5) and (6) considered for (γ_j) in this paper correspond to τ^s_{k+1} and τ^g_{k+1} in the scalar setting. We emphasize that for structured L-BFGS, τ^g_{k+1} has emerged as the most effective choice in the scalar setting, cf., e.g., [38, 40, 50].

Next we observe that the scalars τ^s_{k+1}, τ^g_{k+1} and τ^z_{k+1} provide inner approximations for the spectrum of the *average Hessian* $\overline{\nabla^2\mathcal{D}_k}$. For convenience, we state this only for quadratic $\mathcal{S}$. The proof is similar to [31, Section 4.1], hence omitted. As an obvious consequence, we conclude that the spectrum of D_{k+1} approximates from within the spectrum of the average Hessian if we additionally impose τ^s_{k+1} as lower and τ^z_{k+1} as upper bound for T_{k+1}, which may not hold without these bounds. The latter approximation property serves as a motivation to impose these bounds in the numerical experiments.

Lemma 2.3 *Let $\mathcal{D} : \mathcal{X} \to \mathbb{R}$ be twice continuously differentiable and $\mathcal{S} : \mathcal{X} \to \mathbb{R}$ be quadratic with Hessian $S \in \mathcal{L}(\mathcal{X}, \mathcal{X})$. Let $x_k, x_{k+1} \in \mathcal{X}$ be such that $z_k^T s_k > 0$, where $y_k := \nabla\mathcal{J}(x_{k+1}) - \nabla\mathcal{J}(x_k)$, $s_k := x_{k+1} - x_k$ and $z_k := y_k - S_{k+1}s_k$ with $S_{k+1} := S$. Let*

$$\overline{\nabla^2\mathcal{D}_k} := \int_0^1 \nabla^2\mathcal{D}(x_k + ts_k)\,\mathrm{d}t \qquad (7)$$

be positive semi-definite. Then the scalars from Definition 2.2 satisfy

$$\lambda(\overline{\nabla^2\mathcal{D}_k}) \le \tau^s_{k+1} \le \tau^g_{k+1} \le \tau^z_{k+1} \le \Lambda(\overline{\nabla^2\mathcal{D}_k}).$$

If, in addition, $(e_j)_{j\in\mathcal{I}} \subset \mathcal{X}$ is an orthonormal basis of $\mathcal{X}$ and we project onto $\hat{T}_{k+1} := [\tau^s_{k+1}, \tau^z_{k+1}]$ instead of T_{k+1} in the formulas (5) or (6), then the diagonal operator $D_{k+1} := \sum_{j\in\mathcal{I}} \gamma_j e_j e_j^T$ with $(\gamma_j)_{j\in\mathcal{I}}$ according to (5) or (6) satisfies

$$\lambda(\overline{\nabla^2\mathcal{D}_k}) \le \lambda(D_{k+1}) \le \Lambda(D_{k+1}) \le \Lambda(\overline{\nabla^2\mathcal{D}_k}).$$

2.2 Cautious Updating

Algorithm ROSE uses *cautious updating* [47] both for the decision whether (s_k, y_k) is stored and for the choice of the seed matrix $B^{(0)}_{k+1}$, the latter through requiring $\lambda(D_{k+1}), \Lambda(D_{k+1}) \in T_{k+1}$ in Line 16, which effectively safeguards $\|D_{k+1}\|$ and $\|D^{-1}_{k+1}\|$ from becoming too small or too large in relation to $\nabla\mathcal{J}(x_{k+1})$. Combined, these two techniques yield sufficient control over the condition number of B_k to prove, without convexity assumptions on $\mathcal{J}$, that $\lim_{k\to\infty} \nabla\mathcal{J}(x_k) = 0$, cf. Theorem 3.3, and that $(\mathcal{J}(x_k))$ converges q-linearly, cf. Theorems 3.7 and 3.9. It is important to note that for $\nabla\mathcal{J}(x_k) \to 0$ the lower bound ω^l_{k+1} and the upper bound ω^u_{k+1} that appear in the definition of T_{k+1} satisfy $\omega^l_{k+1} \to 0$ and $\omega^u_{k+1} \to \infty$, respectively. Asymptotically, this allows D_{k+1} to have arbitrarily small positive eigenvalues and arbitrarily large positive eigenvalues. This is more flexible than safeguarding with a fixed positive number which would artificially restrict the spectrum of B_{k+1}.

Cautious updating has previously been used in L-BFGS, for instance in [12, 13, 46, 50]. Except for our own work [50], however, the cautious updating that we use in the present paper differs from that in the aforementioned references. Most importantly, only [50] proves linear convergence in a non-convex setting, and it is also the only contribution that considers cautious updating for a *structured* L-BFGS method.

2.3 The Line Search

The step length α_k in classical L-BFGS is often computed in such a way that it satisfies the weak or the strong Wolfe–Powell conditions. However, some authors determine α_k by backtracking until the Armijo condition holds. For *structured* L-BFGS, the authors of [1, 50] observed in numerical experiments that Armijo is preferable, so we have good reason to include all these line searches. For later reference, let us make the line searches explicit. For Armijo with backtracking, we fix constants $\beta, \sigma \in (0, 1)$. The step size $\alpha_k > 0$ for the iterate x_k with associated descent direction d_k is obtained by successively trying for α_k the numbers $1, \beta, \beta^2, \beta^3, \ldots$ and accepting the first one for which $x_{k+1} = x_k + \alpha_k d_k$ satisfies

$$\mathcal{J}(x_{k+1}) \le \mathcal{J}(x_k) + \alpha_k\sigma\nabla\mathcal{J}(x_k)^T d_k. \qquad (8)$$

For the Wolfe–Powell conditions, respectively, the strong Wolfe–Powell conditions we additionally fix $\eta \in (\sigma, 1)$. A step size $\alpha_k > 0$ is accepted if it satisfies (8) and

$$\nabla\mathcal{J}(x_{k+1})^T d_k \ge \eta\nabla\mathcal{J}(x_k)^T d_k, \qquad \text{respectively,} \qquad (9)$$
$$\left|\nabla\mathcal{J}(x_{k+1})^T d_k\right| \le \eta\left|\nabla\mathcal{J}(x_k)^T d_k\right|.$$

As is common practice when working with the Wolfe–Powell conditions (weak or strong), the first trial step size for α_k is always the full step $\alpha_k = 1$.

3 Convergence Results

This section presents convergence results for Algorithm ROSE. Specifically, global convergence is addressed in Sect. 3.1, linear convergence in Sect. 3.2, and finite convergence on suitable quadratics in Sect. 3.3.

As it turns out, the convergence analysis developed in [50] for Algorithm TULIP can be generalized to cover Algorithm ROSE. By doing so, we essentially obtain the same convergence results for ROSE in Sects. 3.1 and 3.2 as for TULIP in [50]. Since the changes required in the proofs are straightforward, we only spell them out for the proof of Theorem 3.3 1). To distinguish the convergence properties of ROSE from those of TULIP, we show in Sect. 3.3 that for a particular class of functions, ROSE converges after finitely many iterations, whereas TULIP does not.

3.1 Global Convergence of Algorithm ROSE

For x_0 in Algorithm ROSE, we define the level set, respectively, the extended level set by

$$\Omega := \left\{ x \in \mathcal{X} : \ \mathcal{J}(x) \leq \mathcal{J}(x_0) \right\}$$
$$\text{and} \qquad \Omega_\delta := \Omega + \mathbb{B}_\delta(0), \quad \text{where } \delta > 0.$$

The global convergence of Algorithm ROSE holds under the following assumption.

Assumption 3.1 1) The objective $\mathcal{J} : \mathcal{X} \to \mathbb{R}$ is continuously differentiable and bounded below.

2) The gradient of $\mathcal{J}$ is Lipschitz continuous in Ω with constant $L > 0$, i.e., there holds $\|\nabla\mathcal{J}(x) - \nabla\mathcal{J}(\hat{x})\| \leq L\|x - \hat{x}\|$ for all $x, \hat{x} \in \Omega$.
3) The sequence $(\|S_k\|)$ in Algorithm ROSE is bounded.
4) The step size α_k is, for all k, computed by Armijo with backtracking (8) or according to the Wolfe–Powell conditions (9). In the first case, we suppose in addition that there is $\delta > 0$ such that $\mathcal{J}$ or $\nabla\mathcal{J}$ is uniformly continuous in Ω_δ.
5) The value $c_0 = 0$ is only chosen in Algorithm ROSE if with this choice there holds $\sup_k \|(B_k^{(0)})^{-1}\| < \infty$ (which is, for instance, the case if $(\|S_k^{-1}\|)$ is bounded).
6) The value $C_0 = \infty$ is only chosen in Algorithm ROSE if any of the following holds:

- Line 15 is replaced by "Let $T_{k+1} := [\omega_{k+1}^l, P_{k+1}(\tau_{k+1}^g)]$."
- $\mathcal{J}$ is twice continuously differentiable, $\overline{G_k} := \overline{\nabla^2\mathcal{J}_k} - S_k$ is symmetric positive semi-definite for all k, and $(\|\overline{G_k}\|)$ is bounded. For quadratic $\mathcal{S}$ and $S_k = \nabla^2\mathcal{S}(x_k)$ for all k, we can also replace $\overline{G_k}$ in the preceding sentence by $\overline{\nabla^2\mathcal{D}_k} := \int_0^1 \nabla^2\mathcal{D}(x_k + ts_k)\,\mathrm{d}t$.

Remark 3.2 Parts 1)–4) of Assumption 3.1 are more general than the assumptions that are typically used in the literature to analyze the convergence of L-BFGS. For instance, it is often assumed that the objective $\mathcal{J}$ is twice continuously differentiable with bounded level sets and only the Wolfe–Powell line search is discussed. Let us point out two benefits of our more general setting to illustrate its usefulness. First, in our numerical experience with image registration problems, Armijo with backtracking is often more effective than Wolfe–Powell, and in particular, we use it in the numerical experiments of this paper. On the other hand, Wolfe–Powell is the standard line search for L-BFGS, so it is important to include it, too. Second, our weaker differentiability requirements cover the situation that $\mathcal{J}$ contains a penalty term, e.g., the classical quadratic penalty $\sum_i \max\{0, g_i(x)\}^2$ (here, the g_i stem from inequality constraints $g_i(x) \leq 0$). Since penalty terms are frequently used in image registration [39, Chapter 15], allowing penalty terms in the objective is a desirable feature.

Next we comment on some aspects of 3), 5) and 6). The sequence $(\|S_k\|)$ is for instance bounded if we select $S_k = \nabla^2\mathcal{S}(x_k)$ for all k, (x_k) is bounded and $\nabla^2\mathcal{S}$ is Hölder continuous in Ω. The sequence $(\|(B_k^{(0)})^{-1}\|)$ is for instance bounded if we select $S_k = \nabla^2\mathcal{S}(x_k)$ for all k and the regularizer $\mathcal{S}$ is strongly convex. We stress, however, that the boundedness of $(\|(B_k^{(0)})^{-1}\|)$ is only required if we want to choose $c_0 = 0$ in ROSE. Yet, all convergence results hold for $c_0 > 0$ and the numerical results in Sect. 4 are obtained with $c_0 = 10^{-6}$ and include non-convex regularizers for which a positive semi-definite approximation of the Hessian is available for S_k. Note that $c_0 = 0$ implies $\omega_{k+1}^l = 0$ for all k, while $C_0 = \infty$ implies $\omega_{k+1}^u = \infty$ for all k. That is, for $c_0 = 0$, respectively, $C_0 = \infty$, the lower safeguard is zero, resp., the upper safeguard is irrelevant, just as in standard L-BFGS. Observe in this context that if $\mathcal{S} \equiv 0$ and $\mathcal{D}$ is a strongly convex C^2 function with Lipschitz continuous gradient in Ω, then Assumption 3.1 holds with $c_0 = 0$ and $C_0 = \infty$; hence, we can use $S_k = 0$ for all k. In this case, we recover classical L-BFGS if c_s is smaller than the modulus of convexity of $\mathcal{D}$.

We now state the global convergence of ROSE in the sense $\lim_{k\to\infty} \|\nabla\mathcal{J}(x_k)\| = 0$, without convexity of the objective. For L-BFGS-type methods, this strong form of global convergence has rarely been shown in the literature in non-convex settings, cf. the discussion in [50].

Theorem 3.3 *Let Assumption 3.1 hold. Then:*

1) *Algorithm ROSE is well-defined.*
2) *If Algorithm ROSE is applied with $\epsilon = 0$, then it either terminates after finitely many iterations with an x_k that satisfies $\nabla\mathcal{J}(x_k) = 0$ or it generates a sequence (x_k) such that $(\mathcal{J}(x_k))$ is strictly monotonically decreasing*

and convergent and there holds

$$\lim_{k\to\infty} \|\nabla \mathcal{J}(x_k)\| = 0. \tag{10}$$

In particular, every cluster point of (x_k) *is stationary.*

3) *If Algorithm* ROSE *is applied with* $\epsilon > 0$*, then it terminates after finitely many iterations with an* x_k *that satisfies* $\|\nabla \mathcal{J}(x_k)\| \le \epsilon$.

Proof The claim 1) can be established similarly as in [50, Lemma 4.3], while 2) follows as in [50, Theorem 4.8] and 3) is an obvious consequence of 2).

Let us spell out the two changes required in the proof of [50, Lemma 4.3] to obtain 1). First, the intervals $[\tau^z, \tau^s]$ and $[\tau^z, \tau^g]$ that appear in the proof in [50] have to be replaced by T_{k+1}. It is then argued in the proof that these intervals are non-empty, but this is obvious for T_{k+1}. Second, it is used that B_k is positive definite, so we have to establish this here.

Lemma 4.1 from [50] yields that B_k is positive definite if $B_k^{(0)}$ is. Since $B_k^{(0)} = D_k + S_k$ by Line 5 of ROSE and since D_k and S_k are positive semi-definite by Line 16 and Line 12, $B_k^{(0)}$ is positive semi-definite. To argue that it is actually positive definite, we have to distinguish two cases. If the constant $c_0 \ge 0$ in ROSE is positive, we infer that unless x_k is stationary (in which case the algorithm terminates before generating B_k, so there is nothing to show), there holds $\omega_k^l > 0$. As ω_k^l is the lower bound of the interval T_k (Line 15), it follows from $\lambda(D_k) \in T_k$ (Line 16) that $\lambda(D_k) \ge \omega_k^l > 0$, so D_k is positive definite, hence $B_k^{(0)}$ is, too. By Assumption 3.1 5) the choice $c_0 = 0$ is only made if $B_k^{(0)}$ is invertible, so $B_k^{(0)}$ is positive definite in this case, too. □

Remark 3.4 1) If (x_k) is bounded, the uniform continuity in Assumption 3.1 4) can be dropped for finite dimensional $\mathcal{X}$.

2) Note that while Theorem 3.3 states that cluster points of (x_k) are necessarily stationary, it it does not ensure that cluster points exist. If $\mathcal{X}$ is finite dimensional, then the boundedness of (x_k) is sufficient for that existence. If $\mathcal{X}$ is infinite dimensional, then it is more delicate to ensure the existence of cluster points. However, if (x_k) is bounded and $\nabla\mathcal{J}$ is weakly continuous, then the existence of *weak* cluster points is guaranteed and it is easy to show that every weak cluster point is stationary.

3.2 Rate of Convergence of Algorithm ROSE

The convergence rate of the classical L-BFGS method is q-linear for the objective and r-linear for the iterates under *global* strong convexity of $\mathcal{J}$, cf. [48], and sublinear for non-convex objectives [12]. For structured L-BFGS, we established linear convergence for non-convex objectives in [50]. A close inspection of the results from [50] reveals that they essentially apply to Algorithm ROSE, too. This yields two results on the rate of convergence. First we obtain under a Kurdyka–Łojasiewicz-type inequality, which is weaker than *local* strong convexity, that the objective converges q-linearly and the iterates and their gradients converge r-linearly. Second, the same type of convergence also holds if there is a cluster point in whose neighborhood $\mathcal{J}$ is strongly convex, which is the classical sufficient optimality condition of second order. Both results rely on the following assumption.

Assumption 3.5 1) Assumption 3.1 holds.

2) Algorithm ROSE is applied with $\epsilon = 0$ and does not terminate finitely.
3) The sequences $(\|B_k\|)$ and $(\|B_k^{-1}\|)$ are bounded.
4) If the Armijo condition with backtracking is used for step size selection in Algorithm ROSE, there is $\delta > 0$ such that $\mathcal{J}$ is uniformly continuous in Ω_δ or $\nabla\mathcal{J}$ is Lipschitz continuous in Ω_δ.

Remark 3.6 The boundedness assumption 3) is easy to satisfy in the structured setting of this paper. Specifically, it follows as in [50] that $(\|B_k\|)$ is bounded if at least one of the two statements in Assumption 3.1 6) holds. Notably, the first of those statements only limits the size of the interval T_{k+1} and does not involve convexity of the objective. The boundedness of $(\|B_k^{-1}\|)$ is, for instance, guaranteed if (S_k) is chosen uniformly positive definite. If $\mathcal{S}$ is strongly convex, this holds for $S_k = \nabla^2\mathcal{S}(x_k)$, but more sophisticated choices may be available for the problem at hand. If a positive semi-definite approximation of $\nabla^2\mathcal{S}(x_k)$ is available, we may choose it as S_k and, if necessary, add δI with a small $\delta > 0$ to ensure boundedness of $(\|B_k^{-1}\|)$. In any case, the data-fitting term $\mathcal{D}$ in (1) can clearly be non-convex.

3.2.1 Linear Convergence Under a Kurdyka–Łojasiewicz-type Inequality

In this subsection, we state the linear convergence of Algorithm ROSE based on a Kurdyka–Łojasiewicz-type inequality. To introduce this inequality, let us consider the sequence (x_k) generated by Algorithm ROSE and recall from Theorem 3.3 that $(\mathcal{J}(x_k))$ is strictly monotonically decreasing and that $\mathcal{J}^* := \lim_{k\to\infty}\mathcal{J}(x_k)$ exists. We demand that there are $\bar{k}, \mu > 0$ such that

$$\mathcal{J}(x_k) - \mathcal{J}^* \le \frac{1}{\mu}\|\nabla\mathcal{J}(x_k)\|^2 \qquad \forall k \ge \bar{k}. \tag{11}$$

Comments on how this inequality relates to other Kurdyka–Łojasiewicz-type-inequalities can be found in [50]. It is not difficult to check that well-known *error bound conditions* like the one in [62, Assumption 2] imply (11). Thus, the following result holds in particular under any of those error bound conditions. The significance of (11) is that it allows for minimizers that are neither locally unique nor have a regular Hessian, while still resulting in linear convergence. We recall that the parameter σ appears in the Armijo condition (8).

Theorem 3.7 *Let Assumption 3.5 and (11) hold. Then there exists x^* such that*

1) *there hold $\nabla\mathcal{J}(x^*) = 0$ and $\mathcal{J}^* = \mathcal{J}(x^*)$;*
2) *the iterates (x^k) converge r-linearly to x^*;*
3) *the gradients $(\nabla\mathcal{J}(x_k))$ converge r-linearly to zero;*
4) *the function values $(\mathcal{J}(x_k))$ converge q-linearly to $\mathcal{J}(x^*)$. Specifically, we have*

$$\mathcal{J}(x_{k+1}) - \mathcal{J}(x^*) \leq \left(1 - \frac{\sigma\alpha_k\mu}{\|B_k\|}\right)\Big[\mathcal{J}(x_k) - \mathcal{J}(x^*)\Big] \quad \forall k \geq \bar{k}. \tag{12}$$

The supremum of the term in round brackets is strictly smaller than 1.

Proof Identical to the proof of [50, Theorem 4.7]. □

Remark 3.8 As in Remark 3.4, the statements concerning Ω_δ in Assumption 3.1 and in Assumption 3.5 can be dropped if $\mathcal{X}$ is finite dimensional and (x_k) is bounded.

3.2.2 Linear Convergence Under Local Strong Convexity

We now derive linear convergence under a different set of assumptions than in Theorem 3.7.

For the special case $\mathcal{S} \equiv 0$ and $S_k = 0$ for all k, the following result may be viewed as an improved version of the classical convergence result [48, Thm. 7.1] from Liu and Nocedal on L-BFGS, the most notable improvement being that strong convexity is required only locally.

Theorem 3.9 *Let Assumption 3.5 hold except for the statements concerning Ω_δ. Let (x_k) have a cluster point x^* such that $\mathcal{J}|_{\mathcal{N}}$ is μ-strongly convex, where $\mathcal{N} \subset \Omega$ is a convex neighborhood of x^*. Then*

1) *there holds $\mathcal{J}(x^*) + \mu\|x - x^*\|^2 \leq \mathcal{J}(x)$ for all $x \in \mathcal{N}$;*
2) *the iterates (x_k) converge r-linearly to x^*;*
3) *the gradients $(\nabla\mathcal{J}(x_k))$ converge r-linearly to zero;*
4) *the function values $(\mathcal{J}(x_k))$ converge q-linearly to $\mathcal{J}(x^*)$. Specifically, if $\bar{k}$ is such that $x_k \in \mathcal{N}$ for all $k \geq \bar{k}$, then we have*

$$\mathcal{J}(x_{k+1}) - \mathcal{J}(x^*) \leq \left(1 - \frac{2\sigma\alpha_k\mu}{\|B_k\|}\right)\Big[\mathcal{J}(x_k) - \mathcal{J}(x^*)\Big] \quad \forall k \geq \bar{k}. \tag{13}$$

The supremum of the term in round brackets is strictly smaller than 1.

Proof Identical to the proof of [50, Theorem 4.9]. □

Remark 3.10 If $\mathcal{J}$ is μ-strongly convex in the convex level set Ω, then (13) holds for $\bar{k} = 0$.

3.3 Finite Convergence on Suitable Quadratics

In Sects. 3.1 and 3.2, we have established convergence results for ROSE for fairly general objective functions. We have also pointed out that these results are essentially identical to those derived in [50] for TULIP. In this subsection, we show in a model setting that the better Hessian approximation of ROSE results in ROSE requiring fewer iterations than TULIP. Specifically, we establish in this section that for certain quadratic objective functions, ROSE with $\ell = 0$ can find the exact minimizer after a finite number of iterations, whereas this does not hold for TULIP. We confirm these results in the numerical experiments for quadratics in Sect. 4.2. By extension, we expect that in related settings, ROSE requires much fewer iterations than TULIP. Indeed, for the real-world image registration problems in Sect. 4.1.1 we observe this to be the case.

The key property of our model setting is that for some k it enables the seed matrix $B_{k+1}^{(0)}$ of ROSE to approximate the Hessian $\nabla^2\mathcal{J}(x_{k+1})$ *exactly*, yielding that x_{k+2} is the *exact* minimizer if $\ell = 0$. Since $B_{k+1}^{(0)} = D_{k+1} + S_{k+1}$, where we choose $S_{k+1} = \nabla^2\mathcal{S}(x_{k+1})$ and D_{k+1} is a diagonal matrix with diagonal elements that satisfy $\lambda(D_{k+1}), \Lambda(D_{k+1}) \in T_{k+1}$, cf. Line 16, we can only ensure $B_{k+1}^{(0)} = \nabla^2\mathcal{J}(x_{k+1})$ if $\mathcal{D}$ has a diagonal Hessian $\nabla^2\mathcal{D}$ such that $\lambda(\nabla^2\mathcal{D}), \Lambda(\nabla^2\mathcal{D}) \in T_{k+1}$. As this severely limits the class of addressable objective functions, we emphasize again that the goal is not to prove another convergence result under general assumptions, but to show rigorously that ROSE can require significantly fewer iterations than TULIP.

We have now motivated the majority of assumptions that are required to prove the first result. We recall that the objective function has the form $\mathcal{J} = \mathcal{S} + \mathcal{D}$.

Lemma 3.11 *Let Assumption 3.1 hold. Let $\mathcal{D}, \mathcal{S} : \mathcal{X} \to \mathbb{R}$ be convex quadratics with positive semi-definite Hessians $D, S \in \mathcal{L}(\mathcal{X})$. Let $(e_j)_{j\in\mathcal{I}} \subset \mathcal{X}$ be an orthonormal basis of $\mathcal{X}$ and suppose that D is diagonal wrt. (e_j), i.e., $D = \sum_{j\in\mathcal{I}} \hat{\gamma}_j e_j e_j^T$ with $(\hat{\gamma}_j) \subset [0, \infty)$. Consider ROSE for some $k \in \mathbb{N}_0$ with $S_{k+1} := S$. Then:*

1) *(5) and (6) yield the same diagonal operator $D_{k+1} = \sum_{j\in\mathcal{I}} \gamma_j e_j e_j^T$ with $(\gamma_j) \subset [0, \infty)$.*
2) *For any j such that $\hat{\gamma}_j \in T_{k+1}$ and $s_k^T e_j \neq 0$, there holds $\gamma_j = \hat{\gamma}_j$, i.e., $B_{k+1}^{(0)} e_j = \nabla^2\mathcal{J} e_j$.*

Proof Proof of 1):

Comparing the formulas (5) and (6), it suffices to show that either $z_k^T e_j$ and $s_k^T e_j$ have identical signs or $z_k^T e_j = 0$.

From

$$z_k = y_k - Ss_k = \nabla\mathcal{D}(x_{k+1}) - \nabla\mathcal{D}(x_k) = Ds_k$$

it follows that $z_k^T e_j = \hat{\gamma}_j s_k^T e_j$. Thus, either $z_k^T e_j = 0$ (if $\hat{\gamma}_j = 0$) or $z_k^T e_j$ and $s_k^T e_j$ have identical signs (if $\hat{\gamma}_j > 0$).

Proof of 2):

The formula $z_k^T e_j = \hat{\gamma}_j s_k^T e_j$ derived in the proof of 1) implies that $\frac{z_k^T e_j}{s_k^T e_j} = \hat{\gamma}_j$ for any j with $s_k^T e_j \neq 0$. Since $\hat{\gamma}_j \in T_{k+1}$,

it follows from (5) that $\gamma_j = \hat{\gamma}_j$ for these j. □

Observe that for $\ell = 0$ the approximation property of Lemma 3.11 2) is particularly strong in that $B_{k+1}e_j = \nabla^2\mathcal{J}e_j$. Since $\mathcal{J}$ is quadratic, Taylor expansion yields

$$\begin{aligned}\nabla\mathcal{J}(x_{k+1}+d_{k+1})^T e_j &= \nabla\mathcal{J}(x_{k+1})^T e_j + (d_{k+1})^T\nabla^2\mathcal{J}e_j \\ &= \nabla\mathcal{J}(x_{k+1})^T e_j - \nabla\mathcal{J}(x_{k+1})e_j = 0,\end{aligned}$$

where the second equality relies on $B_{k+1}d_{k+1} = -\nabla\mathcal{J}(x_{k+1})$. This shows that if $s_k^T e_j \neq 0$ for all j, then $\nabla\mathcal{J}(x_{k+1} + d_{k+1}) = 0$, i.e., $x_{k+1} + d_{k+1}$ is the minimizer of $\mathcal{J}$. It is not difficult to show that in this case, step size $\alpha_{k+1} = 1$ is chosen, and thus, ROSE terminates with x^{k+2} being the minimizer. Essentially, we have established the following result.

Corollary 3.12 *Let Assumption 3.1 hold. Let $\mathcal{D}, \mathcal{S} : \mathcal{X} \to \mathbb{R}$ be convex quadratics with Hessians $D, S \in \mathcal{L}(\mathcal{X})$. Let $(e_j)_{j\in\mathcal{I}} \subset \mathcal{X}$ be an orthonormal basis of $\mathcal{X}$ and suppose that D is diagonal wrt. (e_j) and positive definite.*

Consider Algorithm ROSE with $\ell = 0$ and the choice $S_k := S$ for all k. Then: There is $\hat{K} \in \mathbb{N}_0$ such that if $s_K^T e_j \neq 0$ for all $j \in \mathcal{I}$ and some $K \geq \hat{K}$, then ROSE terminates for $k = K + 1$ in Line 11 with the global minimizer of $\mathcal{J}$. If the constants c_0, C_0 in ROSE satisfy $c_0 \leq \lambda(D)$ and $C_0 \geq \Lambda(D)$, then $\hat{K} = 0$.

Proof Our prior discussion establishes the claims, but recall that it was based on Lemma 3.11 2) whose application requires $\lambda(D), \Lambda(D) \in T_{k+1}$. The specified choice for c_0, C_0 guarantees that this requirement is satisfied because it implies $T_{k+1} = [\omega_{k+1}^l, \omega_{k+1}^u] \supset [c_0, C_0] \supset [\lambda(D), \Lambda(D)]$. Here, we used that $z_k^T s_k = s_k^T Ds_k > 0$ because D is positive definite by assumption. Without knowledge about c_0 and C_0, it still holds that $\omega_{k+1}^l \to 0$ and $\omega_{k+1}^u \to \infty$ for $k \to \infty$, hence $\lambda(D), \Lambda(D) \in [\omega_{k+1}^l, \omega_{k+1}^u] = T_{k+1}$ holds for all sufficiently large k. The limits for ω_{k+1}^l and ω_{k+1}^u follow from the fact that $\nabla\mathcal{J}(x_k) \to 0$, which is established in Theorem 3.3. □

The assumption that $s_k^T e_j \neq 0$ for all j is difficult to guarantee in general, so let us also prove a result without this assumption. The price to pay is that we need S to be diagonal. This is helpful because if $B_k = D_k + S$ is diagonal, then $s_k^T e_j = 0$ is equivalent to $\nabla\mathcal{J}(x_k)^T e_j = 0$. Since $\nabla^2\mathcal{J}$ is also diagonal in this setting, we obtain $\nabla\mathcal{J}(x_{k+1})^T e_j = 0$ by Taylor expansion, which in turn gives $d_{k+1}^T e_j = 0$. Hence, $\nabla\mathcal{J}(x_{k+1}+d_{k+1})^T e_j = \nabla\mathcal{J}(x_{k+1})^T e_j + (d_{k+1})^T\nabla^2\mathcal{J}e_j = 0$, the same result as for those indices j with $s_k^T e_j \neq 0$. Necessarily, then, $\nabla\mathcal{J}(x_{k+1} + d_{k+1})^T e_j = 0$ for all j, thus $\nabla\mathcal{J}(x_{k+1} + d_{k+1}) = 0$. As before, $\alpha_{k+1} = 1$ is selected and ROSE terminates with x_{k+2}, which is the minimizer of $\mathcal{J}$. These observations yield the following result.

Lemma 3.13 *Let Assumption 3.1 hold. Let $\mathcal{D}, \mathcal{S} : \mathcal{X} \to \mathbb{R}$ be convex quadratics with diagonal Hessians $D, S \in \mathcal{L}(\mathcal{X})$ wrt. the orthonormal basis $(e_j)_{j\in\mathcal{I}}$. Also, let D be positive definite. Consider Algorithm ROSE with $\ell = 0$ and the choice $S_k := S$ for all k. Let $K \in \mathbb{N}_0$ be the smallest number such that $\omega_K^l \leq \lambda(D)$ and $\omega_K^u \geq \Lambda(D)$ are satisfied. Then ROSE terminates for $k = K+1$ in Line 11 with the global minimizer of $\mathcal{J}$. If the constants c_0, C_0 in ROSE satisfy $c_0 \leq \lambda(D)$ and $C_0 \geq \Lambda(D)$, then $K = 0$.*

The arguments of this subsection remain valid under small perturbations of $\nabla^2\mathcal{J}$ and B_k, although the final iterate will no longer be the *exact* minimizer. The precise statement, whose proof is omitted for brevity, reads as follows. We recall that ϵ is the termination tolerance in Algorithm ROSE.

Lemma 3.14 *Let Assumption 3.1 hold. Let $\mathcal{D} : \mathcal{X} \to \mathbb{R}$ be twice continuously differentiable and strongly convex with Hessian $\nabla^2\mathcal{D}(x) = D + T_1(x)$ for all $x \in \mathcal{X}$, where $D \in \mathcal{L}(\mathcal{X})$ is diagonal wrt. the orthonormal basis $(e_j)_{j\in\mathcal{I}}$ and positive definite. Let $\mathcal{S} : \mathcal{X} \to \mathbb{R}$ be twice continuously differentiable with Hessian $\nabla^2\mathcal{S}(x) = S + T_2(x) + T_3(x)$ for all $x \in \mathcal{X}$, where $S + T_2(x)$ is positive semi-definite for all $x \in \mathcal{X}$. Let $\hat{x} \in \mathcal{X}$ and $\hat{\Omega} := \{x \in \mathcal{X} : \mathcal{J}(x) \leq \mathcal{J}(\hat{x})\}$. Consider Algorithm ROSE with $\ell = 0$ and the choice $S_k := S + T_2(x_k)$ for all k. Then: For all $\epsilon > 0$ there is $\delta > 0$ such that if $\sup_{x\in\hat{\Omega}}\|T_1(x)\| + \|T_2(x)\| + \|T_3(x)\| \leq \delta$, then for any $x_0 \in \hat{\Omega}$, ROSE terminates for $k = K + 1$ in Line 11 with x_{K+2} satisfying $\|\nabla\mathcal{J}(x_{K+2})\| \leq \epsilon$, where $K \in \mathbb{N}_0$ is the smallest number such that $\omega_K^l \leq \lambda(D)$ and $\omega_K^u \geq \Lambda(D)$ are satisfied. If $c_0 \leq \lambda(D)$ and $C_0 \geq \Lambda(D)$, then $K = 0$.*

Remark 3.15 1) Let us specialize Lemma 3.14 to a strongly convex regularizer of the form $\mathcal{S}(x) = \alpha s(x)$, $\alpha > 0$, $S_k := \alpha\nabla^2 s(x_k)$ for all k, and a quadratic function $\mathcal{D}$ with a positive definite and diagonal Hessian. Lemma 3.14 with $T_1(x) := T_3(x) := S := 0$ and $T_2(x) := \alpha\nabla^2 s(x_k)$ yields that as α decreases we expect ROSE with $\ell = 0$ to terminate *earlier* because $\sup_{x\in\hat{\Omega}}\|T_1(x)\| + \|T_2(x)\| +$

$\|T_3(x)\|$ becomes smaller. In particular, if we allow $\alpha = 0$ then ROSE with $\ell = 0$ requires only 2 iterations if c_0 and $1/C_0$ are small enough, and the same statement holds if $\alpha > 0$ is sufficiently small. In contrast, for larger values of α or $\ell > 0$ more than two iterations may be required. Similarly, if c_0 and $1/C_0$ are not sufficiently small or the interval T_{k+1} is restricted in such a way that $\lambda(D)$ or $\Lambda(D)$ do not belong to it, then ROSE with $\ell = 0$ will not terminate in two iterations. The numerical results in Sect. 4.2 match these expectations quite well, cf. Table 5.

2) Let us briefly discuss what convergence behavior we expect of ROSE for $\ell > 0$. For $\ell = 0$, the key point is that a good approximation $B_{k+1}^{(0)}$ of $\nabla^2\mathcal{J}(x_{k+1})$ translates into B_{k+1} being a good approximation of $\nabla^2\mathcal{J}(x_{k+1})$. For $\ell > 0$ the identity $B_{k+1}^{(0)} = B_{k+1}$ is no longer true, so although $B_{k+1}^{(0)} = \nabla^2\mathcal{J}(x_{k+1})$ remains valid, the rank-two updates actually destroy the perfect approximation and guarantee that $B_{k+1} \neq \nabla^2\mathcal{J}(x_{k+1})$. On the other hand, as ℓ increases ROSE becomes more similar to a BFGS-type method, so there should be a tendency of improving (linear) convergence rates, which suggests lower iteration numbers. By observing that the rank-two updates modify the eigenvalues of B_{k+1} vs. $B_{k+1}^{(0)}$, we may further suspect that the updates are most helpful in settings where $B_{k+1}^{(0)}$ is unable to approximate the spectrum of $\nabla^2\mathcal{J}$ well, for instance if T_{k+1} is too small. Thus, if T_{k+1} does not include $\lambda(D)$ or $\Lambda(D)$ (by a certain margin), we expect that the performance of ROSE with $\ell = 0$ is worse than with $\ell > 0$, regardless of the value of α. Again, the numerical results in Sect. 4.2 are very well aligned with these expectations, cf. Table 5.

3) It is clear that if $\nabla^2\mathcal{D}$ is diagonal but not a scalar multiple of the identity, then the seed matrix of TULIP cannot agree with the exact Hessian. Therefore, the iteration numbers of TULIP may be significantly larger than those of ROSE in the setting of Lemma 3.13. Table 5 confirms this. By extension, we also expect lower iteration numbers from ROSE if $\mathcal{D}$ has a Hessian that is close to a diagonal matrix (at least near the minimizer) or ill-conditioned, cf. also Lemma 3.14. The results of the real-world experiments in Sect. 4.1.1 support this view.

4 Numerical Experiments

The numerical experiments are divided into two parts. First we consider real-life image registration problems to demonstrate the practical merits of ROSE; then, we study quadratic model problems to illustrate its convergence properties in an academic setting. All experiments are performed in MATLAB (R2023b) on an Apple M1 Pro with 32GB of RAM.

4.1 Real-life Image Registration Problems

We first demonstrate the effectiveness of Algorithm ROSE over its predecessor TULIP from [50] using the same 22 real-life large-scale highly non-convex and ill-posed image registration problems that were used in [50] to compare TULIP with L-BFGS and other structured L-BFGS methods. A description of these problems is provided next.

4.1.1 Problems Under Consideration

Registration problems are generally highly non-convex and ill-posed. Given a pair of images T and R, the goal is to find a transformation field ϕ such that the transformed image $T(\phi)$ is similar to R, i.e., $T \circ \phi \approx R$. To determine ϕ, we solve an unconstrained optimization problem

$$\min_{\phi} \mathcal{J}(\phi) = \mathcal{D}(\phi; T, R) + \alpha S(\phi), \tag{14}$$

where $\mathcal{D}$ measures the similarity between the transformed image $T(\phi)$ and R. The regularizer $\mathcal{S} = \alpha S$ guarantees that the problem is solvable and it enforces smoothness in the field. We follow the “discretize-then optimize” approach. To obtain a clearer comparison, we work with a fixed discretization instead of a multilevel approach. In consequence, for each problem the transformation field satisfies $\phi \in \mathbb{R}^n$ with a fixed n that defines the number of variables. The values of n are provided in Table 1.

An important quality measure for the registration is the *target registration error (TRE)*. It measures the distance between the landmark locations in the reference image and in the target image transformed with the transformation field ϕ.

The 22 test cases, listed in Table 1, cover many different registration models, estimating small to large deformations. The three-dimensional (3D) lung CT images are from the well-known DIR dataset [20, 41], and the rest of the datasets are from [54]; in Fig. 5 we display five of the datasets together with registration results. 2D-Disc images are an academic example with large deformations from [18]. The six test cases for the datasets 2D-PET-CT and 2D-MRI-Head perform multi-modal registration of images from two different modalities.

Let us also point out that the test cases do not include landmark constrained registration problems, but that they fit in the framework of this paper if the constraints are eliminated as proposed in [33], resulting in an unconstrained optimization problem formulated in the basis of the constraint set.

The test cases comprise the fidelity measures sum of squared difference (SSD), mutual information (MI) [64] and normalized gradient fields (NGF) [34]. They include a quadratic first-order (Elas) [16], a quadratic second-order (Curv) [29], and a non-quadratic first-order (H-elas) [18]

Table 1 We use 22 non-quadratic image registration problems as test cases (TC)

TC	Dataset	n	$\mathcal{D}$	$\mathcal{S}$	α	Initial TRE
1	2D-Hands	32 768	SSD	Curv	$1.5 \cdot 10^3$	1.04 (0.62)
2	2D-Hands	32 768	SSD	Elas	$1.5 \cdot 10^3$	1.04 (0.62)
3	2D-Hands	32 768	SSD	H-elas	$(10^3, 20)$	1.04 (0.62)
4	2D-Hands	32 768	NGF	Curv	0.01	1.04 (0.62)
5	2D-Hands	32 768	NGF	Elas	1	1.04 (0.62)
6	2D-Hands	32 768	NGF	H-elas	(1, 1)	1.04 (0.62)
7	2D-Hands	32 768	MI	Curv	$5 \cdot 10^{-3}$	1.04 (0.62)
8	2D-Hands	32 768	MI	Elas	$5 \cdot 10^{-3}$	1.04 (0.62)
9	2D-Hands	32 768	MI	H-elas	$(10^{-3}, 1)$	1.04 (0.62)
10	2D-PET-CT	32 768	MI	Elas	10^{-4}	N.A
11	2D-PET-CT	32 768	MI	Curv	0.1	N.A
12	2D-PET-CT	32 768	NGF	Elas	0.05	N.A
13	2D-PET-CT	32 768	NGF	Curv	10	N.A
14	2D-MRI-Head	32 768	MI	Elas	10^{-3}	N.A
15	2D-MRI-Head	32 768	NGF	Elas	0.1	N.A
16	2D-Disc	512	SSD	H-elas	(100, 20)	N.A
17	3D-Brain	49 152	SSD	H-elas	(100, 10, 100)	N.A
18	3D-Lung	294 912	NGF	Curv	100	3.89 (2.78)
19	3D-Lung	307 200	NGF	Curv	100	9.83 (4.86)
20	3D-Lung	319 488	NGF	Curv	100	6.94 (4.05)
21	3D-Lung	331 776	NGF	Curv	100	7.48 (5.51)
22	3D-Lung	344 064	NGF	Curv	100	4.34 (3.90)

Data-fidelity (MI, NGF, SSD) and regularization (Curvature, Elasticity, Hyperelasticity) are denoted by $\mathcal{D}$ and $\mathcal{S}$, respectively; the regularization parameter is α, cf. (14). The problem size is $n = d \cdot \prod_{k=1}^{d} m_k$, where $d \in \{2, 3\}$ denotes the data dimensionality (2D or 3D) and m_k the corresponding data resolution. The data resolution for Hands, PET-CT, and MRI data is 128×128, for Disc data 16×16, Lung data $64 \times 64 \times X$ where $X \in [24, 28]$, and Brain data $32 \times 16 \times 32$. The last column (Initial TRE) reports the initial target registration error and in brackets the standard deviation, cf. Sect. 4.1.1. In Fig. 5 we display five of the data sets together with registration results

regularizer. We stress that the hyperelastic regularizer (H-elas) is non-convex. Also, computing the full Hessian of the hyperelastic regularizer is expensive. As proposed in [18], we therefore use a Gauss–Newton-like approximation of the Hessian for S_k. In particular, this approximation is positive semi-definite. It is important to note here that the convergence analysis of ROSE allows for $S_k \neq \nabla^2 \mathcal{S}(x_k)$.

4.1.2 Algorithmic Settings

For TULIP and ROSE, we use $S_k = \nabla^2 \mathcal{S}(x_k)$, except for the hyperelastic regularizer where S_k only approximates $\nabla^2 \mathcal{S}(x_k)$. The seed matrix in TULIP is $B_k^{(0)} = \tau_k I + S_k$, and we compute τ_k by the adaptive approach from [50], which is the best performing method of [50]. Since we use only one approach for τ_k, we will refer to it simply as TULIP. In ROSE, we have $B_k^{(0)} = D_k + S_k$, and we determine the diagonal entries of D_k according to (5), respectively, (6), which we refer to as ROSE-Ds and ROSE-Dg, respectively. The linear system in the two-loop recursion is solved inexactly and matrix-free using MINRES [59] with a Jacobi preconditioner. Unless stated otherwise, MINRES is terminated after 50 iterations or when the relative residual falls below 10^{-2}. For the 22 image registration problems under consideration, these settings were identified in [50] to work well for TULIP and to outperform the preconditioned conjugate gradients method.

The image processing operations are carried out matrix-free with the open-source image registration toolbox FAIR [54].

For the stopping criteria of the optimization methods, we follow [54, p. 78]. That is, we stop if all of the conditions

- $|\mathcal{J}(x_k) - \mathcal{J}(x_{k-1})| \leq 10^{-5}\big(1 + |\mathcal{J}(x_0)|\big)$,
- $\|x_k - x_{k-1}\| \leq 10^{-3}\big(1 + \|x_k\|\big)$,
- $\|\nabla \mathcal{J}(x_k)\| \leq 10^{-3}\big(1 + |\mathcal{J}(x_0)|\big)$

are satisfied. We consistently use $\ell = 5$ in ROSE and TULIP. We employ the Armijo line search routine from FAIR [54] with parameters $LSmaxIter = 50$ and $LSreduction =$

Table 2 Parameter values for Algorithms ROSE and ES

Algorithm ROSE					Algorithm ES				
c_s	c_0	C_0	c_1	c_2	ε_0	ε_1	η_0	η_1	η_2
10^{-9}	10^{-6}	10^{6}	10^{-6}	1	10^{-3}	10^{-4}	10	30	50

10^{-4}, where the latter corresponds to σ in (8). We do not consider the Wolfe–Powell line search because it does not work as well on image registration problems [50]. The remaining parameter values of Algorithms ROSE and TULIP are specified in Table 2, which also contains parameter values for Algorithm ES that we introduce below.

4.1.3 Evaluation Measures

We select run time and solution accuracy as the main criteria to evaluate the performance of the algorithms, where the solution accuracy is measured with target registration error (TRE) [30] discussed in Sect. 4.1.1.

To visualize the performance, we use the performance profiles of Dolan and Moré [26] which allow to compare the performance of several optimization methods on a given set of problems with respect to a performance metric (e.g., run time). Denote by S the set of methods, by P the set of problems, and by $t_{p,s} \in (0, \infty]$ the value that method $s \in S$ achieves on problem $p \in P$ in the performance metric, where a smaller value of $t_{p,s}$ is better and $t_{p,s} = \infty$ indicates that algorithm s did not solve p. The performance profile for s is the function $\rho_s : [1, \infty) \to [0, 1]$ given by

$$\rho_s(\tau) = \frac{\left|\left\{p \in P : r_{p,s} \leq \tau\right\}\right|}{|P|}, \qquad \text{where}$$

$$r_{p,s} = \frac{t_{p,s}}{\min\left\{t_{p,\sigma} : \sigma \in S\right\}}.$$

We see that ρ_s is the cumulative distribution function with respect to the performance metric t. Note that τ in $\rho_s(\tau)$ is not related to the scaling factor τ used for seed matrices, but both are standard notation.

4.1.4 Results

Experimental comparison of different choices for $\mathbf{D_k}$

We pair the two schemes ROSE-Ds and ROSE-Dg with various restrictions of T_{k+1}. Here, by restriction we mean that we choose $\lambda(D_{k+1})$, $\Lambda(D_{k+1})$ in Line 16 of ROSE from a sub-interval $\hat{T}_{k+1} := [a_{k+1}, b_{k+1}] \cap T_{k+1}$ (this is compatible with the algorithm since it still ensures $\lambda(D_{k+1}), \Lambda(D_{k+1}) \in T_{k+1}$). Specifically, we are interested in sub-intervals that use τ^s_{k+1} as lower and τ^z_{k+1} as upper bound because this guarantees that the spectrum of D_{k+1} is related to that of the (average) Hessian of the data fidelity term, cf. Lemma 2.3.

Table 3 Diagonal approximation schemes and choice of bounds a, b for $\hat{T} = [a, b] \cap T$

No	diagonal approximation	lower bound a	upper bound b
1	ROSE-Ds: Formula (5)	ω^l	ω^u
2		ω^l	$\min(\lvert\tau^z\rvert, \omega^u)$
3		$\max(\lvert\tau^s\rvert, \omega^l)$	$\min(\lvert\tau^z\rvert, \omega^u)$
4	ROSE-Dg: Formula (6)	ω^l	ω^u
5		ω^l	$\min(\lvert\tau^z\rvert, \omega^u)$
6		$\max(\lvert\tau^s\rvert, \omega^l)$	$\min(\lvert\tau^z\rvert, \omega^u)$

We work with the absolute values of τ^s and τ^z to account for the case $s^T z < 0$. It is easy to see that $|\tau^s| \leq |\tau^z|$. We point out that the parameters for the cautious updates are chosen such that $\omega^l \ll |\tau^s|$ and $\omega^u \gg |\tau^z|$

Omitting the index $k + 1$, we detail different combinations of lower and upper bounds that we use for $\hat{T}_{k+1}$ in Table 3.

Figure 1 shows that all variants of ROSE are either faster or at least similar to TULIP. All variants of ROSE-Dg outperform the adaptive version of TULIP, while only one variant of ROSE-Ds is faster than TULIP.

In both ROSE-Ds and ROSE-Dg, the variant with lower bound $|\tau^s|$ is the fastest but yields the lowest accuracy. Less run time is attributable to lower objective value reduction. The other two variants with lower bound ω^l are almost identical in performance, but the one with upper bound $|\tau^z|$ has a slight advantage in terms of accuracy and run time, which is why we focus next on further improving this variant's run time by managing the accuracy of the linear solver over the iterations.

Effect of linear solver on run-time performance

While a diagonal choice of D_k infuses more information into the structured L-BFGS method compared to a scaled identity, the run time is below the anticipated level. Figure 2b confirms that the diagonal schemes require fewer function evaluations than the scaled identity, but indicates that for the diagonal choice the linear solver requires a much higher number of total iterations. For instance, it requires more than double the amount of iterations for 70% of the problems.This is the main reason for its underperformance..

Since ROSE-Dg is superior to ROSE-Ds, which is consistent with the literature on structured L-BFGS [38, 40, 50], we exclude ROSE-Ds from the remaining experiments on image registration.

Improved run-time performance with earlier stopping

We reduce the computational time required by the linear solver by stopping earlier. Specifically, instead of allowing a maximum of 50 iterations as before, we switch between 10, 30, and 50 iterations, respectively, depending on whether ROSE makes good progress or not. The details are provided in Algorithm ES.

The underlying idea in Algorithm ES is that while far away from a local minimum, a crude approximation of the search direction is enough to obtain a sufficient decrease in the objective value. Therefore, in this case MINRES stops after only η_0 iterations, where $\eta_0 = 10$ in our experiments. As the rate of change in the objective function value decreases, the maximal number of MINRES iterations is increased to obtain a more accurate estimate of the search direction.

Figure 3 compares ROSE-Dg to TULIP, both with and without Algorithm ES. As displayed, ROSE is faster than TULIP on approximately 80% of the problems with almost similar performance in terms of accuracy. In contrast with ROSE, TULIP does not benefit from using ES, suggesting that the lower approximation quality of the seed matrix in TULIP combined with even earlier stopping produces descent directions of poor quality.

Overall performance Table 4 reports the total run time and average target registration error on the 22 image registration problems. The two variants of ROSE-Dg clearly outperform the two variants of TULIP. In particular, the best

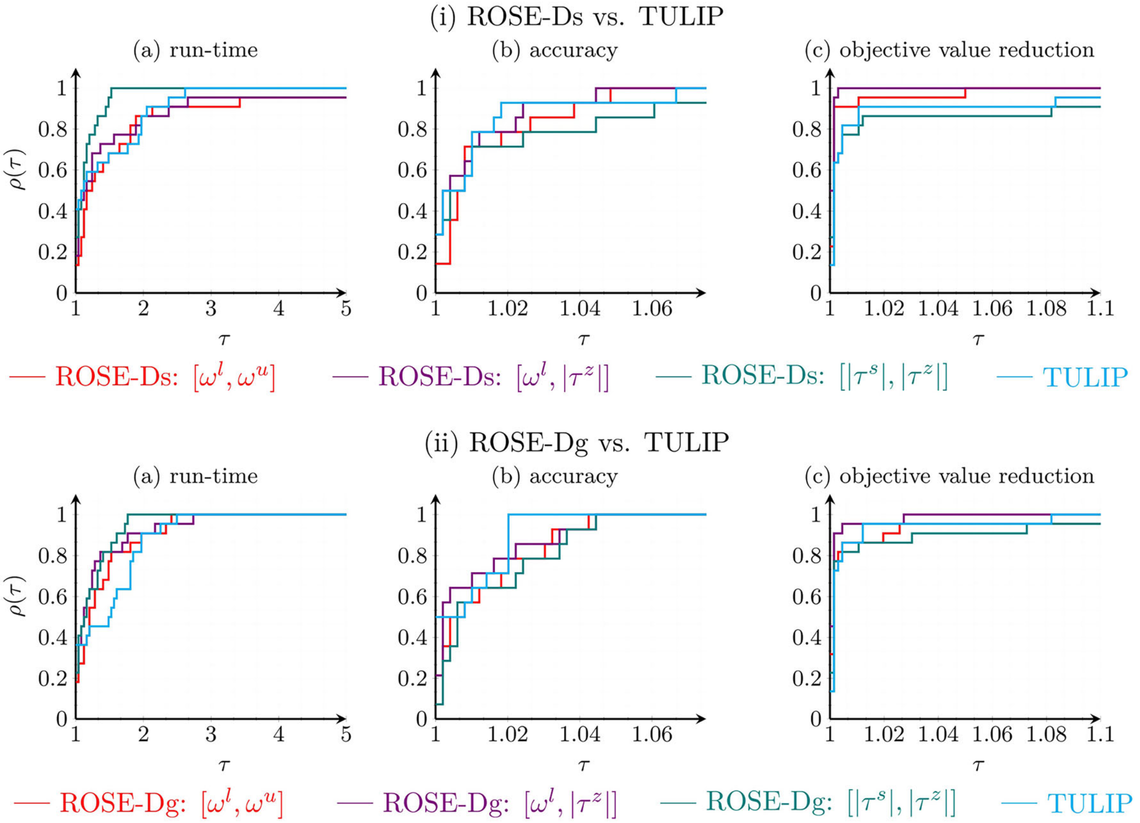

Fig. 1 Performance profiles for different variants of ROSE compared with TULIP. Each variant of ROSE is combined with three different choices of $\hat{T}_{k+1}$

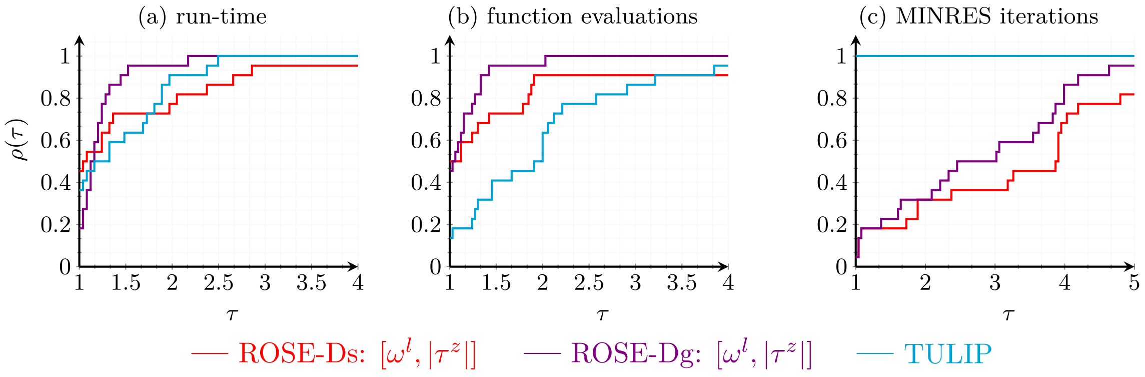

Fig. 2 Performance profiles comparing the best variant of ROSE-Ds and ROSE-Dg to TULIP

Algorithm ES: Early stopping criteria for linear solver; here, ι_k is the number of allowed MINRES iterations at the $(k+1)$-th iteration of ROSE

```
Input: $0 < \varepsilon_1 < \varepsilon_0 << 1$, $0 < \eta_0 < \eta_1 < \eta_2$
if $|\mathcal{J}(x_{k+1}) - \mathcal{J}(x_k)| \leq \varepsilon_1 |\mathcal{J}(x_k)|$ then let $\iota_k := \eta_2$        // small progress → many iterations
else if $|\mathcal{J}(x_{k+1}) - \mathcal{J}(x_k)| \leq \varepsilon_0 |\mathcal{J}(x_k)|$ then let $\iota_k := \eta_1$      // medium progress → in between
else let $\iota_k := \eta_0$                                                                     // large progress → few iterations
```

method ROSE-Dg-ES improves meaningfully over the best variant of TULIP. We recall from [50] that this variant of TULIP outperforms standard L-BFGS by a wide margin on the 22 problems under consideration.

Convergence rate In Fig. 4, we assess the rate of convergence of ROSE in comparison with TULIP.

Unsurprisingly, the diagonal choice of D_k used in ROSE enables much faster convergence than the scalar multiple of the identity used in TULIP.

Visualization of the registration Figure 5 displays the two registration results with the largest TRE for ROSE-Dg and TULIP. It also depicts registration results on three more datasets where landmarks are not available to measure TRE. The objective value reduction in Fig. 5 is very similar for ROSE-Dg and TULIP, which means that the final objective values are close to each other. This suggests that the run-time improvement of ROSE-Dg over TULIP does not affect the quality of the registration.

4.2 Quadratic Problems

We turn to the academic setting of strictly convex quadratics to illustrate what the ideal problem type is for ROSE. This helps to explain the outperformance of ROSE over TULIP observed for image registration in Sect. 4.1. We study how the regularization parameter and the number of L-BFGS update vectors affect convergence of ROSE, and we validate the results of Sect. 3.3.

4.2.1 Problem Under Consideration

We seek the unique minimizer $x^* := (1, 1, \ldots, 1)^T \in \mathbb{R}^{16}$ of the strictly convex quadratic $\mathcal{J}(x) := 0.5(x - x^*)^T [D + \alpha S] (x - x^*)$, where D and S are SPD and $\alpha > 0$. We let D be the diagonal matrix $D_{jj} := \exp(-j)$ with exponentially decaying eigenvalues. This matrix is ill-conditioned with a condition number around 10^7,

reflecting the Hessian of a typical data fidelity term in inverse problems. For S, we use the classical five-point stencil finite difference discretization of the Laplacian with zero boundary conditions on the unit square [10, Section 1.4.3]. We investigate three setups: *weakly* ($\alpha = 10^{-5}$), *mildly* ($\alpha = 10^{-3}$) and *strongly* ($\alpha = 10^{-1}$) regularized problems.

4.2.2 Algorithmic Settings

We use $x_0 = 0$, $S_k = \alpha S$, and we terminate if $\|\nabla \mathcal{J}(x_k)\| \leq 10^{-13}$. The linear system in the two-loop recursion is solved with MATLAB'S backslash. We use the same algorithmic parameters for ROSE and TULIP and the same three variants of $\hat{T}_{k+1}$ as in Sect. 4.1, cf. Tables 2 and 3. ROSE-Ds and ROSE-Dg agree in this example since the formulas (5) and (6) yield identical coefficients, cf. Lemma 3.11 1), so we simply call it ROSE.

4.2.3 Results

Iteration numbers, average number of line search steps, and run times are summarized in Table 5. As expected based on the considerations of Sect. 3.3, the iteration numbers decrease for increasing ℓ, except for ROSE when going from $\ell = 0$ to $\ell = 3$ in case of smaller α and a sufficiently large interval $\hat{T}_{k+1}$, cf. Remark 3.15 2).

Table 5 confirms that, as proved in Lemma 3.14, the problem structure implies that ROSE with $\ell = 0$ terminates after two iterations if α is small enough and $\hat{T}_{k+1}$ is large enough, cf. also Remark 3.15 1).

We repeat that in this section we have deliberately chosen a problem structure that is very favorable for ROSE ($\mathcal{D}$ and $\mathcal{S}$ are strongly convex quadratics, $\mathcal{D}$ has a diagonal Hessian), since our aim is to illustrate in a clear fashion that the better Hessian approximation that ROSE generates can reduce the number of iterations significantly in comparison to TULIP, enhancing the practical performance. Indeed, ROSE consistently requires fewer iterations and less run time than TULIP. The variants of ROSE with $[\omega^l, \omega^u]$ and $[\omega^l, |\tau^z|]$ are more effective than with $[|\tau^s|, |\tau^z|]$, which is in line with the results from Sect. 4.1. In consequence, the outperformance in run time of ROSE over TULIP displayed in Table 5 is substantial for the two variants with larger intervals. In this respect, we recall that the larger the interval, the better D_{k+1} can potentially approximate the spectrum of $\nabla^2 \mathcal{D}$, which is particularly helpful if $\nabla^2 \mathcal{D}$ is ill-conditioned.

In this example, by obtaining a good approximation of $\nabla^2 \mathcal{D}$, ROSE is able to substantially outperform TULIP for weak and mild regularization. As an aside we note that like TULIP, standard L-BFGS struggles on such poorly conditioned problems. All in all, ROSE is robust across the full range of regularization values α.

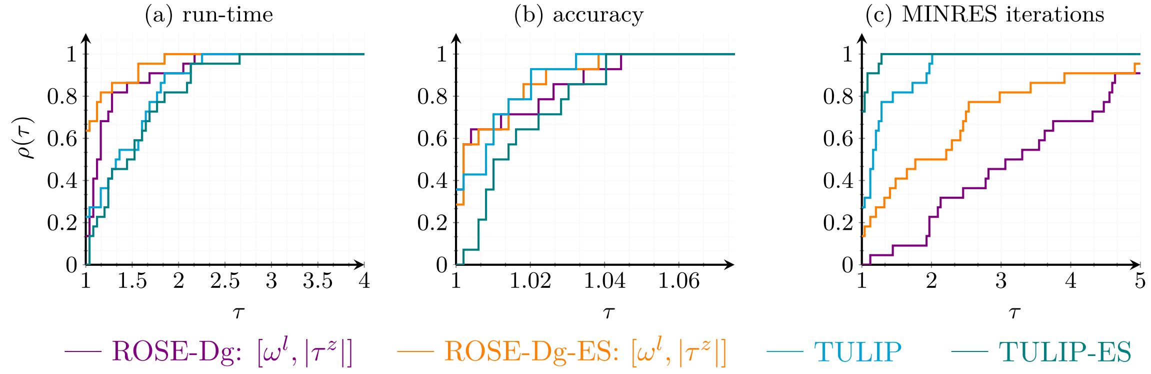

Fig. 3 Performance profiles for ROSE-Dg and TULIP with and without Algorithm ES

Table 4 Performance table

Measures	ROSE-Dg	ROSE-Dg-ES	TULIP	TULIP-ES
total run time (sec.)	624	**614**	904	949
average TRE	0.537	**0.536**	0.538	0.542

The best results are highlighted in bold and are achieved by Rose-Dg-ES

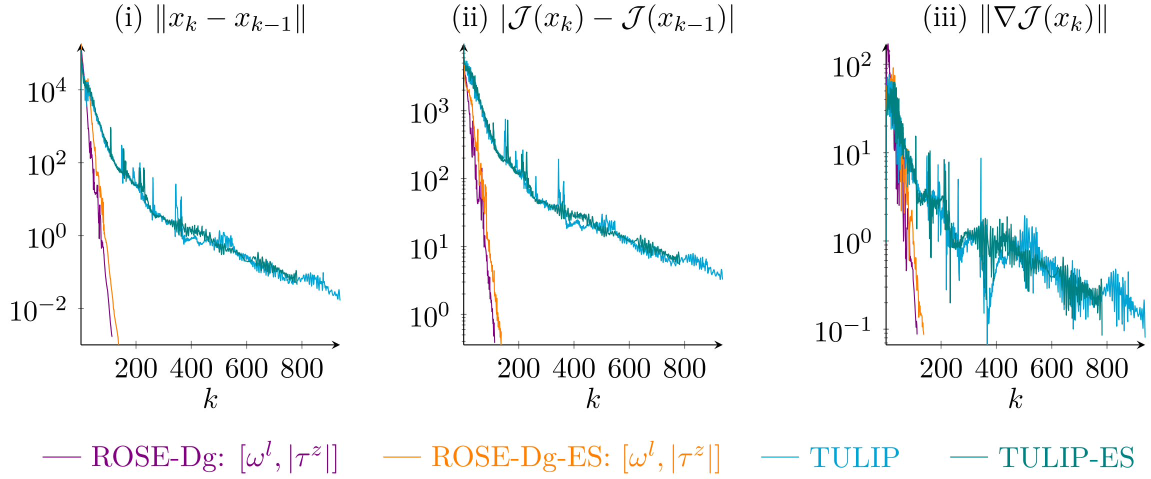

Fig. 4 Convergence behavior of ROSE and TULIP on one image registration problem. The diagonal choices for D_k used in ROSE result in higher rates of convergence compared to the scalar multiple of the identity used in TULIP

5 Conclusions

We have presented ROSE, a structured L-BFGS scheme that allows the first part of the structured seed matrix to be diagonal. We derived two choices for the diagonal part and we compared them to each other numerically. We found that the choice (6) related to the geometric mean is substantially more effective.

ROSE is well suited for structured large-scale optimization problems, including many inverse problems. It comes with strong convergence guarantees, ensuring global and linear convergence in Hilbert space even for non-convex objective functions and absent invertibility of the Hessian. These convergence results do not require the first part of the seed matrix to be diagonal, but they hold in particular for the two proposed diagonal choices. The underlying assumptions are especially mild if the objective includes a regularizer for which a computationally cheap and uniformly positive definite Hessian approximation is available.

In the numerical experiments, we have demonstrated on large-scale real-world inverse problems from medical image registration that ROSE outperforms the structured L-BFGS method TULIP from [50], which in combination with the findings of [50] implies that it also exceeds other structured L-BFGS methods and standard L-BFGS on these problems. In comparison with a scaled identity as in TULIP, the diagonal scaling used in ROSE will usually increase the time required by the iterative linear solver to satisfy the same stopping criterion. On the other hand, diagonal scaling improves the quality of the search directions, so

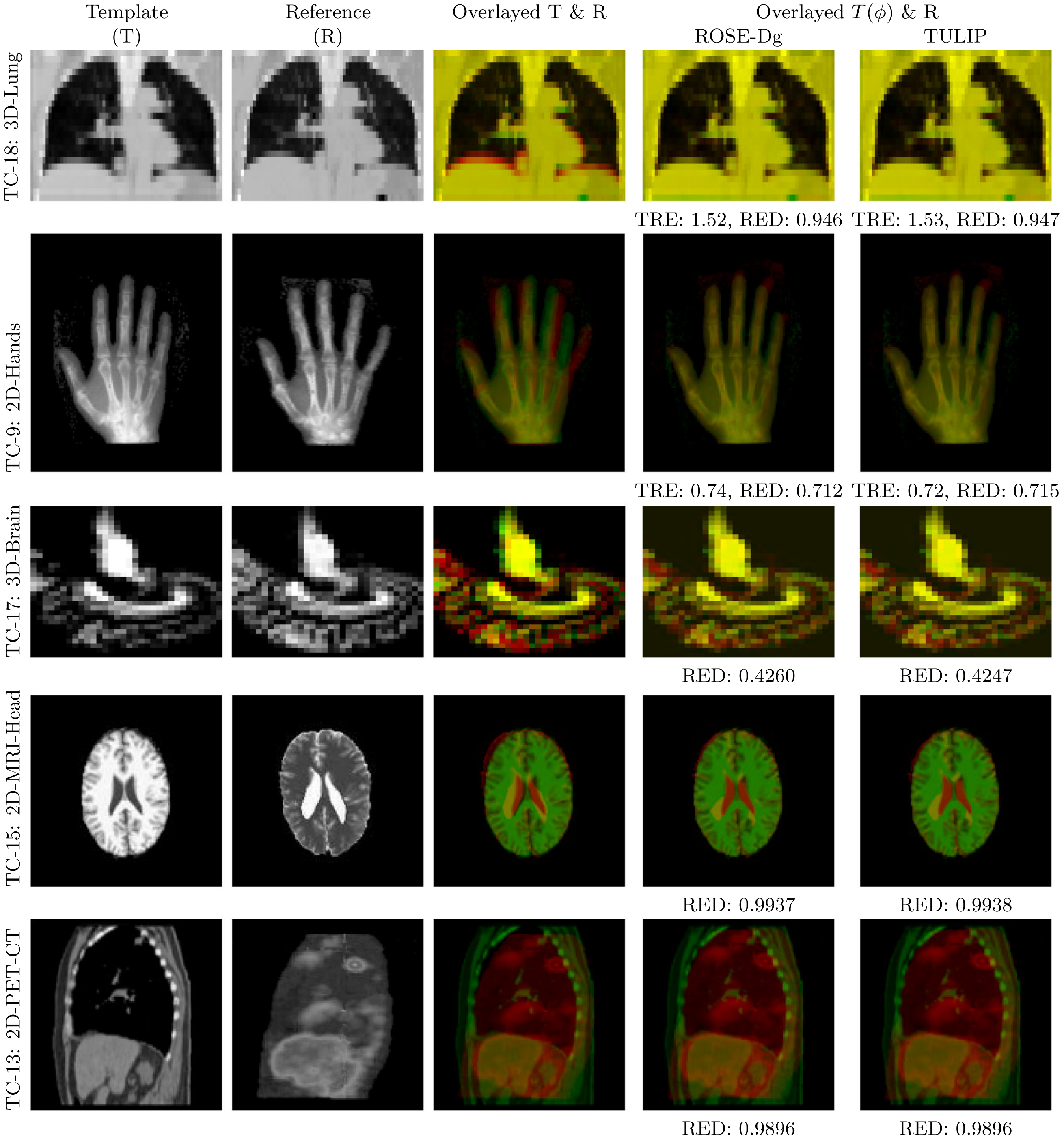

Fig. 5 Registration results for the test cases (TC) 18, 9, 17, 15 and 13; see Table 1. The first two columns show Template (T) and Reference (R) images in the gray scale, the third column and the last two columns show the difference between T and R before and after registration through a two-channel colored composite image, respectively. The composite image shows T and R overlaid in the green and red channels, respectively. Each color channel is scaled between 0 to 1 similar to the gray scale images. Solution accuracy measure TRE is available only for TC-18 and TC-9, where ROSE-Dg and TULIP achieve similar values. The relative objective value reductions (RED) of ROSE-Dg and TULIP are consistently close to each other. This indicates that the solution quality of ROSE-Dg is comparable to that of TULIP although ROSE-Dg requires significantly less run-time on average

Table 5 Comparison of ROSE and TULIP on quadratic problems with different regularization strengths α and different numbers of update vectors ℓ.

	$\alpha = 10^{-5}$					$\alpha = 10^{-3}$					$\alpha = 10^{-1}$				
Method / ℓ	0	3	5	10	∞	0	3	5	10	∞	0	3	5	10	∞
Number of iterations															
TULIP	5000	3088	2222	1106	576	5000	551	409	177	62	284	35	35	22	19
ROSE: $[\omega^l, \omega^u]$	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3
ROSE: $[\omega^l, \lvert\tau^z\rvert]$	5	12	9	8	8	6	12	9	8	8	10	7	6	6	6
ROSE: $[\lvert\tau^s\rvert, \lvert\tau^z\rvert]$	107	51	46	47	38	85	47	31	32	23	37	25	20	15	14
Average number of line searches per iteration															
TULIP	1.01	1.14	1.15	1.16	1.15	1.00	1.11	1.11	1.22	1.39	1.02	1.09	1.06	1.05	1.00
ROSE: $[\omega^l, \omega^u]$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ROSE: $[\omega^l, \lvert\tau^z\rvert]$	1.40	1.08	1.00	1.00	1.00	1.33	1.08	1.00	1.00	1.00	1.10	1.00	1.00	1.00	1.00
ROSE: $[\lvert\tau^s\rvert, \lvert\tau^z\rvert]$	1.62	1.29	1.39	1.15	1.03	1.25	1.11	1.03	1.09	1.00	1.05	1.04	1.05	1.00	1.00
Run time (in milliseconds)															
TULIP	543.2	340.5	248.1	127.9	69.9	542.5	61.9	46.4	21.7	8.7	31.7	5.2	5.1	3.7	3.5
ROSE: $[\omega^l, \omega^u]$	2.9	1.6	1.5	1.5	1.5	1.6	1.5	1.5	1.4	1.4	1.9	1.6	1.6	1.6	1.6
ROSE: $[\omega^l, \lvert\tau^z\rvert]$	2.3	3.0	2.5	2.4	2.3	2.1	2.9	2.4	2.3	2.3	2.6	2.2	2.1	2.0	2.0
ROSE: $[\lvert\tau^s\rvert, \lvert\tau^z\rvert]$	16.6	8.5	7.8	7.9	6.7	13.5	8.1	5.4	5.8	4.5	6.2	4.7	4.0	3.3	3.2

ROSE clearly outperforms TULIP in run time

earlier stopping of the linear solver is appropriate. If this is taken into account, diagonal scaling outperforms scalar scaling in our experiments in that it obtains solutions of comparable accuracy in lower run time. ROSE can be implemented matrix-free and an implementation is available at https://github.com/hariagr/SLBFGS.

Future work should assess the numerical performance of Algorithm ROSE on additional classes of inverse problems. It would also be worthwhile to incorporate other diagonal scalings in ROSE and compare their numerical performance.

Author Contributions All authors have contributed equally to the work.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests

References

1. Ablin, P., Cardoso, J.-F., Gramfort, A.: Faster independent component analysis by preconditioning with hessian approximations. IEEE Trans. Signal Process. **66**(15), 4040–4049 (2018). https://doi.org/10.1109/TSP.2018.2844203
2. Aggrawal, H.O., Modersitzki, J.: Hessian initialization strategies for ℓ-BFGS solving non-linear inverse problems. In: Scale space and variational methods in computer vision, SSVM, 2021. Proceedings, pp. 216–228. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-75549-2_18
3. Al-Baali, M.: Improved Hessian approximations for the limited memory BFGS method. Numer. Algorithms **22**(1), 99–112 (1999). https://doi.org/10.1023/A:1019142304382
4. Amini, K., Rizi, A.G.: A new structured quasi-Newton algorithm using partial information on Hessian. J. Comput. Appl. Math. **234**(3), 805–811 (2010). https://doi.org/10.1016/j.cam.2010.01.044
5. Aminifard, Z., Babaie-Kafaki, S.: A diagonally scaled Newton-type proximal method for minimization of the models with nonsmooth composite cost functions. Comput. Appl. Math. **42**(8), 12 (2023). https://doi.org/10.1007/s40314-023-02494-5
6. Andrei, N.: A diagonal quasi-Newton updating method for unconstrained optimization. Numer. Algorithms 81(2), 575–590 (2019). https://doi.org/10.1007/s11075-018-0562-7
7. Andrei, N.: A diagonal quasi-Newton updating method based on minimizing the measure function of Byrd and Nocedal for unconstrained optimization. Optimization **67**(9), 1553–1568 (2018). https://doi.org/10.1080/02331934.2018.1482298
8. Andrei, N.: A new accelerated diagonal quasi-Newton updating method with scaled forward finite differences directional derivative for unconstrained optimization. Optimization **70**(2), 345–360 (2020). https://doi.org/10.1080/02331934.2020.1712391

9. Babaie-Kafaki, S., Aminifard, Z., Ghafoori, S.: Nonmonotone diagonally scaled limited-memory BFGS methods with application to compressive sensing based on a penalty model. Appl. Numer. Math. **181**, 618–629 (2022). https://doi.org/10.1016/j.apnum.2022.07.008
10. Bartels, S.: *Numerical approximation of partial differential equations*, volume 64 of Texts in Applied Mathematics Cham: Springer, 2016. https://doi.org/10.1007/978-3-319-32354-1
11. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**(1), 141–148 (1988). https://doi.org/10.1093/imanum/8.1.141
12. Berahas, A.S., Jahani, M., Richtárik, P., Takáč, M.: Quasi-Newton methods for machine learning: forget the past, just sample. Optim. Methods Softw. **37**(5), 1668–1704 (2022). https://doi.org/10.1080/10556788.2021.1977806
13. Berahas, A.S., Takáč, M.: A robust multi-batch L-BFGS method for machine learning. Optim. Methods Softw. **35**(1), 191–219 (2019). https://doi.org/10.1080/10556788.2019.1658107
14. Biglari, F.: Dynamic scaling on the limited memory BFGS method. Eur. J. Oper. Res. **243**(3), 697–702 (2014). https://doi.org/10.1016/j.ejor.2014.12.050
15. Boggs, P.T., Byrd, R.H.: Adaptive, limited-memory BFGS algorithms for unconstrained optimization. SIAM J. Optim. **29**(2), 1282–1299 (2019). https://doi.org/10.1137/16M1065100
16. Broit, C.: Optimal registration of deformed images. In: PhD thesis, University of Pennsylvania, 1981. URL: https://repository.upenn.edu/dissertations/AAI8207933
17. Brust, J.J., Di, Z.W., Leyffer, S., Petra, C.G.: Compact representations of structured BFGS matrices. Comput. Optim. Appl. **80**(1), 55–88 (2021). https://doi.org/10.1007/s10589-021-00297-0
18. Burger, M., Modersitzki, J., Ruthotto, L.: A hyperelastic regularization energy for image registration. SIAM J. Sci. Comput. **35**(1), b132–b148 (2013). https://doi.org/10.1137/110835955
19. Byrd, R.H., Nocedal, J., Schnabel, R.B.: Representations of quasi-Newton matrices and their use in limited memory methods. Math. Program. **63**(2(A)), 129–156 (1994). https://doi.org/10.1007/BF01582063
20. Castillo, R., Castillo, E., Guerra, R., Johnson, V.E., McPhail, T., Garg, A.K., Guerrero, T.: A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets. Phys. Med. Biol. **54**(7), 1849–1870 (2019). https://doi.org/10.1088/0031-9155/54/7/001
21. Dener, A., Munson, T.: Accelerating limited-memory quasi-newton convergence for large-scale optimization. In: Rodrigues, J.M.F., Cardoso, P.J.S. (eds.) Computational Science–ICCS 2019, pp. 495–507. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22744-9_39
22. Dennis, J.E., Jr., Martinez, H.J., Tapia, R.A.: Convergence theory for the structured BFGS secant method with an application to nonlinear least squares. J. Optim. Theory Appl. **61**(2), 161–178 (1989). https://doi.org/10.1007/BF00962795
23. Dennis, J.E., Jr., Schnabel, R.B.: Least change secant updates for quasi-Newton methods. SIAM Rev. **21**, 443–459 (1979). https://doi.org/10.1137/1021091
24. Dennis, J.E., Jr., Walker, H.F.: Convergence theorems for least-change secant update methods. SIAM J. Numer. Anal. **18**, 949–987 (1981). https://doi.org/10.1137/0718067
25. Dennis, J.E., Jr., Walker, H.F.: Least-change sparse secant update methods with inaccurate secant conditions. SIAM J. Numer. Anal. **22**, 760–778 (1985). https://doi.org/10.1137/0722046
26. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002). https://doi.org/10.1007/s101070100263
27. Engels, J.R., Martínez, H.J.: Local and superlinear convergence for partially known quasi-Newton methods. SIAM J. Optim. **1**(1), 42–56 (1991). https://doi.org/10.1137/0801005
28. Enshaei, S., Leong, W.J., Farid, M.: Diagonal quasi-Newton method via variational principle under generalized Frobenius norm. Optim. Methods Softw. **31**(6), 1258–1271 (2016). https://doi.org/10.1080/10556788.2016.1196205
29. Fischer, B., Modersitzki, J.: A unified approach to fast image registration and a new curvature based registration technique. Linear Algebr. Appl. **380**, 107–124 (2004). https://doi.org/10.1016/j.laa.2003.10.021
30. Fitzpatrick, J.M., West, J.B.: The distribution of target registration error in rigid-body point-based registration. IEEE Trans. Med. Imaging **20**(9), 917–927 (2001). https://doi.org/10.1109/42.952729
31. Gilbert, J.C., Lemaréchal, C.: Some numerical experiments with variable-storage quasi-Newton algorithms. Math. Program. **45**(3(B)), 407–435 (1989). https://doi.org/10.1007/BF01589113
32. Haber, E.: Quasi-Newton methods for large-scale electromagnetic inverse problems. Inverse Probl. **21**(1), 305–323 (2004). https://doi.org/10.1088/0266-5611/21/1/019
33. Haber, E., Heldmann, S., Modersitzki, J.: A scale-space approach to landmark constrained image registration. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) Scale space and variational methods in computer vision, pp. 612–623. Springer, Berlin (2009). https://doi.org/10.1007/978-3-642-02256-2_51
34. Haber, E., Modersitzki, J.: Intensity gradient based registration and fusion of multi-modal images. In: Sam, A. (ed.) Medical image computing and computer-assisted intervention - MICCAI 2006, pp. 726–733. Springer, Berlin (2006). https://doi.org/10.1007/11866763_89
35. Heldmann, S.: Non-linear registration based on mutual information theory, numerics, and application. Logos-Verlag, Berlin (2006)
36. Huschens, J.: On the use of product structure in secant methods for nonlinear least squares problems. SIAM J. Optim. **4**(1), 108–129 (1994). https://doi.org/10.1137/0804005
37. Hwang, D.M., Kelley, C.T.: Convergence of Broyden's method in Banach spaces. SIAM J. Optim. **2**(3), 505–532 (1992). https://doi.org/10.1137/0802025
38. Jiang, L., Byrd, R. H., Eskow, E., Schnabel, R. B.: A preconditioned L-BFGS algorithm with application to molecular energy minimization. In: Computer Science Technical Reports. 919, (2004)
39. Kimmel, R., Tai, X.-C.: editors. Processing, analyzing and learning of images, shapes, and forms. Part 2, volume 20 of Handbook of Numerical Analysis. Amsterdam: Elsevier/North Holland, (2019). URL: www.sciencedirect.com/handbook/handbook-of-numerical-analysis/vol/20/suppl/C
40. Klemsa, J., Řezáč, J.: Parallel low-memory quasi-newton optimization algorithm for molecular structure. Chem. Phys. Lett. **584**, 10–13 (2013). https://doi.org/10.1016/j.cplett.2013.08.050
41. König, L., Rühaak, J., Derksen, A., Lellmann, J.: A matrix-free approach to parallel and memory-efficient deformable image registration. SIAM J. Sci. Comput. **40**(3), B858–B888 (2018). https://doi.org/10.1137/17m1125522
42. Laumen, M.: A Kantorovich theorem for the structured PSB update in Hilbert space. J. Optim. Theory Appl. **105**(2), 391–415 (2000). https://doi.org/10.1023/A:1004666019575
43. Leong, W.J., Chen, C.Y.: A class of diagonal preconditioners for limited memory BFGS method. Optim. Methods Softw. **28**(2), 379–392 (2013). https://doi.org/10.1080/10556788.2011.653356
44. Leong, W.J., Enshaei, S., Kek, S.L.: Diagonal quasi-Newton methods via least change updating principle with weighted Frobenius norm. Numer. Algorithms **86**(3), 1225–1241 (2021). https://doi.org/10.1007/s11075-020-00930-9
45. Leong, W. J., Farid, M., Hassan, M. A.: Scaling on diagonal quasi-Newton update for large-scale unconstrained optimization. *Bull. Malays. Math. Sci. Soc. (2)*, 35(2):247–256, (2012). Accessed at 22/03/2024. URL: https://math.usm.my/bulletin/pdf/v35n2/v35n2p2.pdf

46. Li, D., Wang, X., Huang, J.: Diagonal BFGS updates and applications to the limited memory BFGS method. Comput. Optim. Appl. **81**(3), 829–856 (2022). https://doi.org/10.1007/s10589-022-00353-3
47. Li, D.-H., Fukushima, M.: On the global convergence of the BFGS method for nonconvex unconstrained optimization problems. SIAM J. Optim. **11**(4), 1054–1064 (2001). https://doi.org/10.1137/S1052623499354242
48. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**(3(B)), 503–528 (1989). https://doi.org/10.1007/BF01589116
49. Liu, Q., Beller, S., Lei, W., Peter, D., Tromp, J.: Pre-conditioned BFGS-based uncertainty quantification in elastic full-waveform inversion. Geophys. J. Int. **228**(2), 796–815 (2022). https://doi.org/10.1093/gji/ggab375
50. Mannel, F., Aggrawal, H.O., Modersitzki, J.: A structured L-BFGS method and its application to inverse problems. Inverse Probl. **40**, 045022 (2024). https://doi.org/10.1088/1361-6420/ad2c31
51. Mannel, F., Rund, A.: A hybrid semismooth quasi-Newton method for nonsmooth optimal control with PDEs. Optim. Eng. **22**(4), 2087–2125 (2021). https://doi.org/10.1007/s11081-020-09523-w
52. Mannel, F., Rund, A.: A hybrid semismooth quasi-Newton method for structured nonsmooth operator equations in Banach spaces. J. Convex Anal. **29**(1), 183–204 (2022)
53. Marjugi, S. M., Leong, W. J.: Diagonal Hessian approximation for limited memory quasi-Newton via variational principle. J. Appl. Math. **2013**, 8 (2013). Id/No 5 https://doi.org/10.1155/2013/523476
54. Modersitzki, J.: *FAIR. Flexible algorithms for image registration*, volume 6 of *Fundam. Algorithms*. Philadelphia, PA: SIAM,https://doi.org/10.1137/1.9780898718843
55. Mohammad, H., Waziri, M.Y.: Structured two-point stepsize gradient methods for nonlinear least squares. J. Optim. Theory Appl. **181**(1), 298–317 (2019). https://doi.org/10.1007/s10957-018-1434-y
56. Nocedal, J.: Updating quasi-Newton matrices with limited storage. Math. Comput. **35**, 773–782 (1980). https://doi.org/10.2307/2006193
57. Nocedal, J., Wright, S.J.: Numerical optimization, 2nd edn. Springer, New York (2006). https://doi.org/10.1007/978-0-387-40065-5.
58. Oren, S.S.: Perspectives on self-scaling variable metric algorithms. J. Optim. Theory Appl. **37**, 137–147 (1982). https://doi.org/10.1007/BF00934764
59. Paige, C.C., Saunders, M.A.: Solution of sparse indefinite systems of linear equations. SIAM J. Numer. Anal. **12**(4), 617–629 (1975). https://doi.org/10.1137/0712047
60. Park, Y., Dhar, S., Boyd, S., Shah, M.: Variable metric proximal gradient method with diagonal barzilai-borwein stepsize. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3597–3601,https://doi.org/10.1109/ICASSP40776.2020.9054193
61. Sahari, M.L., Khaldi, R.: Quasi-Newton type of diagonal updating for the L-BFGS method. Acta Math. Univ. Comen. New Ser. **78**(2), 173–181 (2009)
62. Tseng, P., Yun, S.: A coordinate gradient descent method for nonsmooth separable minimization. Math. Program. **117**(1–2(B)), 387–423 (2009). https://doi.org/10.1007/s10107-007-0170-0
63. Veersé, F., Auroux, D., Fisher, M.: Limited-memory BFGS diagonal preconditioners for a data assimilation problem in meteorology. Optim. Eng. **1**(3), 323–339 (2000). https://doi.org/10.1023/A:1010030224033 newpage
64. Viola, P.: Alignment by maximization of mutual information. In: PhD thesis, Massachusetts Institute of Technology, (1995)
65. Yabe, H., Yamaki, N.: Local and superlinear convergence of structured quasi-Newton methods for nonlinear optimization. J. Oper. Res. Soc. Japan. **39**(4), 541–557 (1996). https://doi.org/10.15807/jorsj.39.541
66. Yang, H., Gunzburger, M., Ju, L.: Fast spherical centroidal Voronoi mesh generation: a Lloyd-preconditioned LBFGS method in parallel. J. Comput. Phys. **367**, 235–252 (2018). https://doi.org/10.1016/j.jcp.2018.04.034
67. Zhou, W., Chen, X.: Global convergence of a new hybrid Gauss-Newton structured BFGS method for nonlinear least squares problems. SIAM J. Optim. **20**(5), 2422 (2010). https://doi.org/10.1137/090748470
68. Zhu, M., Nazareth, J.L., Wolkowicz, H.: The quasi-Cauchy relation and diagonal updating. SIAM J. Optim. **9**(4), 1192–1204 (1999). https://doi.org/10.1137/S1052623498331793