Journal of Computational and Applied Mathematics 310 (2017) 115–128

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Volume constrained 2-phase segmentation method utilizing a linear system solver based on the best uniform polynomial approximation of $x^{-1/2}$

Stanislav Harizanov [*], Svetozar Margenov, Pencho Marinov, Yavor Vutov

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev, bl. 25A, 1113 Sofia, Bulgaria

ARTICLE INFO

Article history:
Received 1 February 2016
Received in revised form 10 June 2016

MSC:
65F10
41A10
68W10
68U10

Keywords:
Primal–dual algorithm
Image segmentation
Best uniform approximation
Parallel algorithm
Constrained convex optimization
Image reconstruction

ABSTRACT

Volume constrained image segmentation aims at improving the quality of image reconstruction via incorporating physical information for the underline object of interest into the mathematical modeling of the segmentation problem. In this paper, we develop a general framework for 3D 2-phase image segmentation, based on constrained ℓ^2 minimization of a non-local regularizer, the Euler–Lagrange derivative of which is the discrete graph-Laplacian of a weighted graph, associated with the image voxels. It involves a convenient change of basis in the image domain, for which the optimization function is decomposed element-wise. Using univariate polynomial approximation techniques, we show that the transformation matrix does not need to be explicitly computed and its action is well approximated by a suitable matrix polynomial. The error is independent of the domain size, thus our approach is applicable to high resolution data. The model allows for adding arbitrary linear terms into the optimization function in order to increase the control on the output, in particular to add its scalar product with another, already known, segmentation vector. Such a "hybrid" process may significantly improve on the individual quality of each of the involved segmentations.

1. Introduction

Image segmentation is a challenging problem and an active research field in Computer Vision. Numerous approaches have been considered and analyzed. Mumford and Shah [1] proposed an energy functional, the minimization of which gave rise to meaningful 2D segmentation. The functional is neither convex nor smooth, its optimization is theoretically difficult and computationally expensive. Simplifications of the model [2–6] have later been proposed. Other 2-phase models, based on [4] are [7–9]. In [10], the regularity term in the Mumford–Shah functional is replaced by the Rudin–Osher–Fatemi (ROF) functional [11]. Graph-based image segmentation studies a suitably constructed weighted graph whose vertices are the image voxels. The phases are related to minimum spanning trees (MST) [12–14] or minimum graph cuts [15–18] and are efficiently numerically derived under the use of polynomial approximations [19,20]. Graph-Laplacian-based segmentation models [16,21–25] can be viewed as an intersection of the above techniques. Here, voxels are again parts of a graph structure,

[*] Corresponding author.
E-mail addresses: sharizanov@parallel.bas.bg (S. Harizanov), margenov@parallel.bas.bg (S. Margenov), pencho@bas.bg (P. Marinov), yavor@parallel.bas.bg (Y. Vutov).
URLs: http://parallel.bas.bg/~sharizanov/ (S. Harizanov), http://parallel.bas.bg/~margenov/CV.htm (S. Margenov), http://parallel.bas.bg/~pencho/pencho_v.htm (P. Marinov), http://parallel.bas.bg/~yavor/index_en.html (Y. Vutov).

http://dx.doi.org/10.1016/j.cam.2016.06.020
0377-0427/© 2016 Elsevier B.V. All rights reserved.

and the functional derivative of the regularization term is the weighted graph-Laplacian operator [26,27]. There are also other approaches, some of them considered classical (e.g., the K-means method and its modifications [28,29]) but they are outside of the scope of the paper.

Accurate segmentation of 3D porous media reconstructions is crucial for the numerical simulations and the computation of the material/object's macro characteristics at the next stage. Due to the highly irregular structure of the segmentation phases and the presence of noise in the image, the classical methods are not reliable and the results of different standard algorithms may differ drastically (even on up to 50% of the data). Furthermore, some of the pores as well as parts of their walls are very thin so high-resolution images are required. Thus, large-scale problems need to be solved in real time and the algorithms should be fast and cheap.

We investigate a new direction of CT data image segmentation, where the physical properties of the scanned specimen are incorporated in the process as constraints [14,25]. In particular, the volume of the solid phase can be determined from the material's density and weight measurements and can be a priori prescribed (mass conservation), while the solid phase itself, should be connected whenever the specimen is a single material piece. The volume constrained model in [25] follows graph-Laplacian-based regularization, has good denoising properties, but tends to oversmooth the solution, thus certain parts of the object (e.g., thin long pores) are lost during the process. The model in [14] uses MST techniques and additionally imposes solid phase connectivity, hence it is not as robust to noise as the former, but on the other hand is useful for deblurring since it produces non-smooth shapes. This paper establishes a general framework, which allows us to import information from any other segmentation as a constraint into the optimization function. Such "hybrid" processes, if carefully conducted, give to their conductor the opportunity to engineer the output object's properties. In the process, one needs to apply the principal square root $Q^{-1/2}$ of a sparse M-matrix Q (block of the graph-Laplacian matrix) to a small number of given vectors. To compute $Q^{-1/2}$ explicitly is numerically expensive, so we propose an algebraic approach, based on univariate uniform polynomial approximation, which addresses the derivation of the vector $Q^{-1/2}q$, without the need of having $Q^{-1/2}$ first.

The paper is organized as follows. In Section 2 a graph-Laplacian-related segmentation problem is formulated. In Section 3 the theoretical foundations of our new approach are built. In Section 4 a primal–dual parallel algorithm for numerically solving the proposed optimization problem is derived. In Section 5 the algorithm is numerically analyzed. Conclusions are drawn in Section 6.

2. Mathematical formulation of the problem

Let us first describe the problem and fix the notation. The latter follows those in [21,25] in order to facilitate the reader. We consider 3D gray-scale images $u : \Omega \to [0, 1]$, where the image domain Ω is a discrete box of dimensions n_1, n_2, and n_3, respectively. Although industrial images usually have much higher gray-scale intensity levels (e.g., 255/65 535 in 8-/16-bit formats, respectively) for the theoretical scope of this paper, it suffices to normalize them in the unit interval. For a simpler matrix–vector notation, we assume the image to be column-wise reshaped as a vector $u \in [0, 1]^{\mathbf{n}}$, with $\mathbf{n} = \text{card}(\Omega) = n_1n_2n_3$, and it will be clear from the context which representation of u we consider. The voxel index set is denoted by $\mathbb{I}_{\mathbf{n}} := \{1, \dots, \mathbf{n}\}$. The discrete segment membership vector $v \in \{0, 1\}^{\mathbf{n}}$ indicates to which class the ith voxel belongs for every $i \in \mathbb{I}_{\mathbf{n}}$. We consider only 2-phase segmentation, therefore unlike the multi-phase setting we can incorporate all the information in a single vector v. More precisely, $v_i = 0$ when the ith voxel belongs to the pore ("air") phase, and $v_i = 1$ when the ith voxel belongs to the solid ("material") phase. The index set is split into two disjoint subsets $\mathbb{I}_{\mathbf{n}} = L \cup U$ that correspond to labeled and unlabeled voxels, respectively. The labeled voxels are a very small set of voxels with a priori known/determined origin. They provide data fidelity information into the model, prevent the minimizer to become trivial (constant vector), and are used for the future characterization of the unlabeled (undetermined) voxels. Without loss of generality (after re-numeration) we consider $L = \{1, \dots, 2\ell\}$, $U = \{2\ell+1, \dots, \mathbf{n}\}$, and we split $v = (v_L, v_U)^T$. Furthermore, $L = L_0 \cup L_1$, where $L_0 := \{i \in L | v_L(i) = 0\} = \{1, \dots, \ell\}$, $L_1 := \{i \in L | v_L(i) = 1\} = \{\ell + 1, \dots, 2\ell\}$.

The indicator function ι_C of a nonempty set C is given by

$$\iota_C(x) = \begin{cases} 0 & \text{if } x \in C, \\ +\infty & \text{otherwise.} \end{cases}$$

When the set is non-empty, convex, and closed, ι_C is proper, convex, and lower semicontinuous (lsc). We denote by e the ones vector $(1, \dots, 1)^T$ of the appropriate dimension, and by $V_\tau(g)$ the hyperplane

$$V_\tau(g) := \{x \mid \langle x, g\rangle = \tau\}.$$

2.1. The weighted graph-Laplacian model

Nonlocal image processing [30,31] allows for simultaneous handling of various pieces of structural information and outperforms classical local processes on complex repetitive structures. In the continuous setting, when the domain $\bar{\Omega}$ is a compact subset of $\mathbb{R}^3$, the nonlocal gradient of $z : \bar{\Omega} \to \mathbb{R}$ at x is defined as

$$\nabla_w z(x) : \bar{\Omega} \mapsto \mathbb{R}, \qquad (\nabla_w z)(x, y) := (z(y) - z(x))\sqrt{w(x, y)}. \tag{1}$$

The weights $w(x, y) \geq 0$ are symmetric, nonzero only for few y, and can be interpreted as the inverse squared distance between x and y in a certain metric. In [22] the functional

$$J(z) := \frac{1}{4}\int_{\bar{\Omega}} |\nabla_w z|^2 dx \tag{2}$$

was used for regularization. It differs from the classical (local, rescaled) Mumford–Shah regularizer

$$J_{MS}(z) = \frac{1}{4}\left(\int_{\bar{\Omega}\setminus K} |\nabla z|^2 dx + \mu|K|\right),$$

since the curve length energy is not considered. The latter allows for efficiently capturing highly non-convex shapes and corners, which is vital for our application—meaningful segmentation of porous media. On the other hand, J is within the family of regularizing functionals [31]

$$J_\phi(z) := \frac{1}{4}\int_{\bar{\Omega}} \phi\left(|\nabla_w z|^2\right) dx, \quad \phi(\mathbb{R}^+) \subseteq \mathbb{R}^+, \; \phi(0) = 0,$$

with $\phi(s)$ convex in $\sqrt{s}$, and corresponds to $\phi(s) = s$. Therefore, it is related to the nonlocal version of the total-variation (NL-ROF) functional [11], derived for $\phi = \sqrt{s}$

$$J_{NL\text{-}ROF}(z) = \frac{1}{4}\int_{\bar{\Omega}} |\nabla_w z| dx = \frac{1}{4}\int_{\bar{\Omega}} \sqrt{\int_{\bar{\Omega}} (z(y) - z(x))^2 w(x, y) dy dx}.$$

The (Euler–Lagrange) variation with respect to z in (2) can be computed as

$$\begin{aligned}\partial_z J(z) &= -\frac{1}{2}\mathrm{div}_w(\nabla_w z(x)) = \frac{1}{2}\int_{\bar{\Omega}} ((\nabla_w z)(y, x) - (\nabla_w z)(x, y))\sqrt{w(x, y)} dy \\ &= \int_{\bar{\Omega}} (z(x) - z(y)) w(x, y) dy =: \triangle_w z(x),\end{aligned}$$

which is the natural continuous generalization of the weighted graph-Laplacian operator [26,27].

Using the adjoint relation, proved in [31]

$$\langle \nabla_w z, \nabla_w z\rangle = -\langle \mathrm{div}_w(\nabla_w z), z\rangle = 2\langle \triangle_w z, z\rangle,$$

we get the discretization of (2) in the form

$$J(v) = \frac{1}{2}\langle \triangle_w v, v\rangle = \frac{1}{4}\sum_{i,j=1}^{\mathbf{n}} w_{i,j}\big(v(i) - v(j)\big)^2. \tag{3}$$

In this paper we apply the functional (3) as a regularization term in our optimization function. For the graph-Laplacian operator $\triangle_w v$ we use the following matrix representation from [21]

$$(\triangle_w v)(i) = \sum_{j=1}^{\mathbf{n}} w_{i,j}\,(v(i) - v(j)) \qquad \Longrightarrow \qquad \triangle_w = D - W,$$

where

$$W := \begin{pmatrix} W_{LL} & W_{LU} \\ W_{UL} & W_{UU}\end{pmatrix}; \qquad D := \mathrm{diag}(d_i)_{i=1}^{\mathbf{n}}, \quad d_i := \sum_{j=1}^{\mathbf{n}} w_{i,j}, \;\; \forall i \in \mathbb{I}_{\mathbf{n}}. \tag{4}$$

The block structure is with respect to labeled/unlabeled voxels, thus W_{LL} is $2\ell \times 2\ell$, W_{LU} is $2\ell \times (\mathbf{n} - 2\ell)$, W_{UL} is $(\mathbf{n} - 2\ell) \times 2\ell$, and W_{UU} is $(\mathbf{n} - 2\ell) \times (\mathbf{n} - 2\ell)$.

Following [18], we incorporate data fidelity as foreground–background marks, thus the segment membership vector v can be derived as the solution of the constrained optimization problem

$$\underset{v\in\{0,1\}^{\mathbf{n}}}{\mathrm{argmin}}\; \frac{1}{2}\langle \triangle_w v, v\rangle \quad \text{subject to } v_L(i) = \begin{cases} 0, & i \in L_0, \\ 1, & i \in L_1. \end{cases} \tag{5}$$

In [22] the authors start with $v|_{L_0} = -1$, $v|_U = 0$, $v|_{L_1} = 1$ at time $t = 0$ and diffuse it, applying several steps of the time-discretized steepest descent method $v_{t+h} = -\triangle_w v_t$ for (3). Such an approach works well for simple phase boundaries (small curve length energy), but not for highly interacting phases as in the case of porous media. We prefer the use of strong constraints on the labeled data (5) as in [21,25].

2.2. Computing the graph's weights

In graph-related image processing, edge weights are commonly taken within the Gaussian class

$$w_{i,j} := \begin{cases} e^{-(h\|F(u,i)-F(u,j)\|_2)^2-\alpha\|i-j\|_2^2} & \text{if } j \in \mathcal{N}_i, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $\mathcal{N}_i$ is a neighborhood of interest for the ith voxel and F is a function on the gray-scale intensity (see [22] for further details). The parameter h serves as a soft threshold and defines which norm values are considered similar. The parameter α controls the spatial decay and avoids blurring of singular patches in the presence of white (Gaussian) noise. Varying the two parameters, we can change the type of the regularization induced by (3). In general, the whole family (6) is known to be robust against noise.

We consider weights similar to [21,25] under minor (but theoretically important) modifications. They are divided into three different groups. Local, purely spatial, isotropic weights W^{geo} link neighboring voxels in the image domain and address their "geometric" similarity. In terms of (6) we have

$$\mathcal{N}_i^{geo} = \{j : \|j-i\|_2 = 1\}, \quad h = 0, \quad \Longrightarrow \; w_{i,j}^{geo} := \begin{cases} e^{-\alpha} & \text{if } j \in \mathcal{N}_i^{geo}, \\ 0 & \text{otherwise.} \end{cases}$$

Since $|\mathcal{N}_i^{geo}| = 6$, after normalization all the positive geometric weights equal 1/6.

Semi-local weights W^{pho} link close-by *unlabeled* voxels with matching intensity profiles and address their structural similarity via

$$\begin{aligned} &F(u,i) = f(i) = \frac{1}{12}\Big(6u(i) + \sum_{j\in\mathcal{N}_i^{geo}} u(j)\Big) \\ &\mathcal{N}_i^{pho} = \{j : \|j-i\|_\infty \le r \& j \in U\} \end{aligned} \quad \Longrightarrow \; w_{i,j}^{pho} := \begin{cases} e^{-h^2(f(i)-f(j))^2-\alpha^{pho}\|i-j\|_2^2} & \text{if } j \in \mathcal{N}_i^{pho}, \\ 0 & \text{otherwise.} \end{cases}$$

For each i, only the m most similar voxels j (i.e., those that minimize $h^2(f(i)-f(j))^2+\alpha^{pho}\|i-j\|_2^2$) are chosen and all the other weights are set to zero. Some remarks are in order. In [21,22,30] vector-valued functions F, namely $F(u,i) = \{f(u_j),\ j \in B_i\}$, where B_i is a "patch" centered at i, are considered. Only 2D images were investigated and both patches B_i and windows $\mathcal{N}_i^{pho}$ were relatively large (5×5 patches in a 11×11 window for [22], and 11×11 patches in a 17×17 window for [21]). Such approach results in quite expensive (both in terms of time and memory) numerical computation of W^{pho} (which has by far the highest complexity in the algorithm, anyways) and it is also anisotropic, since patches are compared elementwise. For macro-structural images, such as architecture, texture, etc., where we have distant, oriented repetitive pattern (windows, arcs, ornaments) this is important and necessary. However, in this paper we deal with micro-, even nano-structures, where patterns are more local and non-oriented, while the images are of high-resolution. Therefore, we choose to follow a more local approach, where $B_i = \mathcal{N}_i^{geo}$, F is a scalar-valued weighted average of the patch intensities, and the window is a $5\times5\times5$ cube (i.e., $r = 2$). The constants h and α^{pho} can be chosen by the user. They influence the segmentation process and differ for different purposes. For example, increasing h smoothens the weights and distributes them close to uniformly, while increasing α^{pho} favors the geometrically closer voxels and impose stronger connectivity of the segmentation phases. In our experiments, we work with $h = \{1, 10\}$, and $\alpha^{pho} = \{0, 1, 5\}$. We also set $m = 3$.

Finally, non-local weights W^{lab} link the unlabeled voxels U to the labeled ones L and address their intensity similarity:

$$F(u,i) = f(i), \; \mathcal{N}_i^{lab} = L, \; \alpha = 0, \quad \Longrightarrow \; w_{i,j}^{lab} := \begin{cases} e^{-h^2(f(i)-f(j))^2} & \text{if } j \in L. \\ 0 & \text{otherwise.} \end{cases}$$

Considered alone in (5), W^{lab} gives rise to almost hard-thresholding (by construction f has certain local smoothing properties). It also guarantees that the weighted graph, associated to the image, is connected, or equivalently, that the graph-Laplacian $\triangle_w$ has a zero eigenvalue of multiplicity 1 (its kernel consists of all the constant vectors), which is important for existence and uniqueness of the segmentation. When introduced in [18], only one foreground and one background pixels were labeled, because a globally minimum cut of a graph with two terminals can be computed efficiently in low-order polynomial time. We apply convex optimization rather than graph-cut techniques, so there is no need to restrict ourselves to $|L| = 2$ and we use $|L| = 6$, instead. Therefore, we have the same cardinality for $\mathcal{N}_i^{geo}$ and $\mathcal{N}_i^{lab}$, which in a sense balances the contribution of the two weight groups.

As suggested in [21,30], we normalize the weights within each group so that they sum up to 1. Finally, the weighted matrix W is a symmetrized convex combination of the corresponding three matrices

$$W^* = \frac{1}{1+\nu^{\rm pho}} W^{geo} + \frac{\nu^{\rm pho}}{1+\nu^{\rm pho}} W^{pho}, \qquad \tilde{W} = \frac{1}{1+\nu^{\rm lab}} W^* + \frac{\nu^{\rm lab}}{1+\nu^{\rm lab}} W^{lab}, \qquad W = \min\left\{\tilde{W}, \tilde{W}^T\right\}. \tag{7}$$

W is non-negative, symmetric. The parameters $\nu^{\rm pho}$, $\nu^{\rm lab}$ are positive. W_{UU} is sparse, and (almost) row-normalized via $d_i \approx 1$ (and always $d_i \le 1$!), $\forall i \in U$. Since $\nu^{\rm lab} > 0$, $\sum_{j\in L} w_{i,j} > 0$, $\forall i \in U$,

$$Q := D_{UU} - W_{UU} \tag{8}$$

is strictly diagonally dominant with non-positive off-diagonal entries, thus an M-matrix. Note that, excluding the labeled voxels from $\mathcal{N}_i^{pho}$ was not considered before. This has negligible effect on the weights $\{w_{i,j} : i \in U,\ j \in L\}$ due to the low cardinality of L and the high compactness of the candidate set $\mathcal{N}_i^{pho}$. On the other hand, it allows for the unit vector e to be a good approximation of the smallest eigenvector (the one, corresponding to the smallest eigenvalue) of Q.

The choices for $\tilde{W}$ and W differ from those in [21,25]. There, they had

$$\tilde{W} = \max\left\{\frac{\nu^{\mathrm{lab}}}{1+\nu^{\mathrm{lab}}}W^{lab}, \frac{1}{1+\nu^{\mathrm{lab}}}W^{*}\right\}, \qquad W = \max\left\{\tilde{W}, \tilde{W}^T\right\}.$$

The first modification is, again, negligible, especially when we set $\nu^{\mathrm{pho}} \gg 1$ and decrease the impact of W^{geo} on W and since we disabled interaction between the positive elements of W^{pho} and W^{lab}. Taking the smaller weight instead of the larger one between $\tilde{w}_{i,j}$ and $\tilde{w}_{j,i}$, however, is conceptually different. By construction both W^{geo} and W^{lab} are symmetric, so $\tilde{w}_{i,j} \ll \tilde{w}_{j,i}$ implies $w_{i,j}^{pho} \ll w_{j,i}^{pho}$, thus within $\mathcal{N}_i^{pho}$, i has better structural matches than j within $\mathcal{N}_j^{pho}$. The latter suggests that $\mathcal{N}_j^{pho}$ is noisy and the gray-scale intensity information around j is mistrustful. In such a case, it is better to decrease its impact on the segmentation process rather than to increase it, so taking minimum over maximum should be more robust against noise. There is also a theoretical justification for our choice. The sparseness of the matrix W is under control (e.g., at most $|\mathcal{N}_i^{geo}| + m + |L| = 12 + m$ elements per row may be non-zero), as well as the spectral condition number of the submatrix Q. We choose $m = 3$ in order to compensate for the potential loss of weight and for finally having $\sum_j w_{i,j}^{pho} < 1$. In this way, we guarantee that the weights $w_{i,j}^{pho}$ will not be dominated by the ones from the other two groups. We experimentally compared both choices for W for the model in [25] and the segmentation outputs were basically the same, meaning that the proposed modifications indeed can be considered negligible from a practical point of view.

3. Constrained convex optimization, problem relaxation and useful reformulation

In this section we use a convenient change of basis to reformulate the general model (5), which allows us to enrich the model and to fully split the variables in its optimization function, without decreasing its numerical efficiency. If we relax problem (5), allowing for the entries of v to lie in the whole unit interval, it admits a unique solution $\bar{v}$, given by (see [21, Theorem 3.2.] for details)

$$Q\,\bar{v}_U = W_{UL}v_L =: q. \tag{9}$$

Here Q is as in (8). The restriction $\mathbf{0} \le \bar{v}_U \le \mathbf{1}$ is guaranteed by the component-wise non-negativity of both Q^{-1} and q (Q^{-1} is a monotone operator, since Q is an M-matrix). To binary discretize $\bar{v}_U$ back, hard thresholding with respect to the middle value 0.5 is typically used.

Such segmentation methods work fine for well-separated, smooth phases, but is often poor and unreliable when applied to noisy (homogeneous) porous media, where the pore phase consists of multiple, non-structured, possibly not even connected pores of various size and shape, that "cut" through the material. Recently, we proposed and investigated in [25] the volume-constrained problem

$$\operatorname*{argmin}_{v\in\{0,1\}^{\mathbf{n}}} \frac{1}{2}\langle \triangle_w\, v, v\rangle \quad \text{subject to } v_L(i) = \begin{cases} 0, & i \in L_0, \\ 1, & i \in L_1; \end{cases} \quad \|v\|_0 = N, \tag{10}$$

for CT data segmentation, where N is the physical volume of the reconstructed specimen. The ℓ_0 pseudo-norm is non-convex and the problem (10) is NP-Hard [32]. In the binary case, the problem is equivalent to

$$\operatorname*{argmin}_{v\in\{0,1\}^{\mathbf{n}}} \frac{1}{2}\langle \triangle_w\, v, v\rangle \quad \text{subject to } v_L(i) = \begin{cases} 0, & i \in L_0, \\ 1, & i \in L_1; \end{cases} \quad e^T v = N, \tag{11}$$

which, in turn can be relaxed to

$$\operatorname*{argmin}_{\mathbf{0}\le v_U \le \mathbf{1}} \frac{1}{2}\langle Q\,v_U, v_U\rangle - \langle v_U, q\rangle \quad \text{subject to } \langle v_U, e\rangle = N_1. \tag{12}$$

Here $N_1 := N - \ell$, and later $\mathbf{n}_1 = \mathbf{n} - 2\ell$.

Problem (12) belongs to the class of equality and inequality constrained Quadratic Programming (QP). In general, dealing with both types of constraints gives rise to slow and expensive iterative algorithms. In [25] we took advantage of the special properties of Q, we neglected the inequality constraint $\mathbf{0} \le v_U \le \mathbf{1}$ and we solved directly the remaining equality constrained QP, using Schur complement and conjugate gradient (CG) method. The minimizer

$$\bar{v}_U = Q^{-1/2}\left(Q^{-1/2}q + \frac{N_1 - \langle Q^{-1/2}q, Q^{-1/2}e\rangle}{\langle Q^{-1/2}e, Q^{-1/2}e\rangle}Q^{-1/2}e\right), \tag{13}$$

is nothing but the preimage w.r.t. $Q^{-1/2}$ of the orthogonal projection of $Q^{-1/2}q$ onto the hyperplane $V_{N_1}\left(Q^{-1/2}e\right)$ (compare with (9)). In most of the numerical computations we performed, the projection was in (or at least very close to) the positive orthant $\mathbb{R}^{\mathbf{n}_1}_{\geq 0}$, which combined with the monotonicity of Q^{-1} automatically lead to $\mathbf{0} \leq \bar{v}_U \leq \mathbf{1}$.

However, during our numerical experiments we also witnessed that the solution $\bar{v}_U$ is quite smooth, the differences between its entries are small, and the volume-constrained effect is not well-presented, since before digitalization, the largest N entries of $\bar{v}_U$ (which at the end define our solid phase) contributed only up to 30%–35% to the scalar product $\langle \bar{v}_U, e\rangle = N_1$. Furthermore, the "cut" between the top Nth and the top $N+1$st entries is usually in the order of the machine error, so the same segmentation algorithm performed on the same input image may produce different (in 2–4 voxels) outputs. Such instability is not welcomed and the purpose of this paper is to address and attack that issue.

The data fidelity term in (12) is hidden in q through the action of W_{UL} onto v_L. This action is not component-wise, but integral in nature, so it seems that, unless all the free parameters ν^{pho}, ν^{lab}, h, α^{pho}, used for generating W are properly tuned, the regularization term $\langle Q\bar{v}_U, \bar{v}_U\rangle$ prevails and the output $\bar{v}_U$ is oversmoothened. On the other hand, when we increase the data fidelity role of q in (12) via enlarging ν^{lab}, we end up with close-to-direct, non-supervised segmentation.

The above numerical observations motivated us to search for a way to add an additional data fidelity term in (12). QP is not a suitable platform for our mathematical modeling, because adding additional constraints is quite expensive there, and at each iteration step we need to apply CG to a possibly ill-conditioned large-scale matrix. We propose the following alternative

$$\underset{\mathbf{0}\leq Q^{-1/2}v'_U\leq \mathbf{1}}{\operatorname{argmin}} \ \frac{1}{2}\langle v'_U, v'_U\rangle - \langle v'_U, q'\rangle - \langle v'_U, s'\rangle \quad \text{subject to } \langle v'_U, e'\rangle = N_1. \tag{14}$$

Here, $v'_U = Q^{1/2}v_U$, $q' = Q^{-1/2}q$, $e' = Q^{-1/2}e$, and $s' = Q^{-1/2}s$ where $s \geq \mathbf{0}$ incorporates additional data fidelity information (e.g., s could store the intensity values of the input image or to be the segment membership vector of another (arbitrary!) segmentation process). Note that, once the constraint $\mathbf{0} \leq Q^{-1/2}v'_U \leq \mathbf{1}$ is properly reformulated, problem (14) gives rise to constrained convex optimization, where only the identity matrix I is involved, thus it allows for implementing fully parallel and computationally cheap numerical solvers. The only additional work is during the preprocessing and postprocessing stages, where the vectors q', e', s', respectively $\bar{v}_U$ need to be derived. This derivation is the subject of the rest of the section.

3.1. The principal square root of Q

It is well-known that the inverse of a sparse matrix, as well as its square root, are usually dense. In [33,34] efficient methods for computing the pth root of a *general* matrix are discussed. Other fractional powers are attacked in [35,36]. For symmetric, positive definite (SPD) sparse matrices either these algorithms can be modified and accelerated or specialized approaches can be pursued [37,38]. The class of M-matrices (actually, we are interested only in its proper subclass of Stieltjes matrices, since the symmetry of Q is crucial for the reformulation (14)) is even a small subset of the SPD class, where the objects satisfy additional properties, so not only all the SPD techniques are applicable, but there is room for even more restrictive research directions. In this paper, we choose to apply classical univariate polynomial algebra to the matrix setting in order to approximate the action of $Q^{-1/2}$, without the necessity of explicitly computing $Q^{-1/2}$.

Denote by $Q = AD_QA^*$ the SVD decomposition of Q. The symmetry of Q implies that D_Q is a diagonal matrix, while A is real-valued and unitary, thus $A^* = A^T = A^{-1}$. The positive-definiteness of Q implies

$$D_Q = \operatorname{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_{\mathbf{n}_1}\}, \qquad \lambda_i > 0.$$

The inverse of Q is $Q^{-1} = AD_Q^{-1}A^T$ and the principle square root of Q^{-1} is

$$Q^{-1/2} := AD'_QA^T, \qquad D'_Q = \operatorname{diag}\left\{\frac{1}{\sqrt{\lambda_1}}, \frac{1}{\sqrt{\lambda_2}}, \ldots, \frac{1}{\sqrt{\lambda_{\mathbf{n}_1}}}\right\}. \tag{15}$$

Before proving our main result, we want to go back to the definition of the weight matrix W. Note that the normalization constant 1, which was used for each of the class-matrices W^{geo}, W^{pho}, W^{lab}, is a degree of freedom for the optimization problem (10) and when rescaling it, we simply rescale J, but do not alter the minimizer. We use this trick to normalize the spectrum of Q.

Theorem 3.1. *Let Q be defined as in (9) where the weight matrix W is generated as in Section 2.2 with weights, normalized to $\frac{1+\nu^{\text{lab}}}{2+\nu^{\text{lab}}}$. Then, the spectrum of Q lies inside the interval*

$$[\underline{\lambda}, \bar{\lambda}] \subset \left[\frac{\nu^{\text{lab}}}{2+\nu^{\text{lab}}}, 1\right] =: I_{lab}. \tag{16}$$

Let $P \in \Pi_k(\mathbb{R})$ be a univariate polynomial of degree k, such that

$$\left|P_k(x) - \frac{1}{\sqrt{x}}\right| < \varepsilon, \quad \forall x \in \left[\frac{\nu^{\text{lab}}}{2+\nu^{\text{lab}}}, 1\right]. \tag{17}$$

Then for every vector $s \in \mathbb{R}^{\mathbf{n}_1}$, *we have*

$$\frac{\left\|P_k(Q)s - Q^{-1/2}s\right\|_{Q^{1/2}}}{\|s\|_{Q^{1/2}}} < \varepsilon. \tag{18}$$

Proof. By construction of W in (7) and Q in (4)–(9), we have for every $i \in U$

$$q_{i,i} - \sum_{j\neq i}|q_{i,j}| = \sum_{j\in L} w_{i,j} \geq \sum_{j} w^{lab}_{i,j} = \frac{1+\nu^{\text{lab}}}{2+\nu^{\text{lab}}}\,\frac{\nu^{\text{lab}}}{1+\nu^{\text{lab}}} = \frac{\nu^{\text{lab}}}{2+\nu^{\text{lab}}}$$

$$q_{i,i} + \sum_{j\neq i}|q_{i,j}| = 2\sum_{j} w_{i,j} - \sum_{j\in L} w_{i,j} \leq 2\frac{1+\nu^{\text{lab}}}{2+\nu^{\text{lab}}} - \frac{\nu^{\text{lab}}}{2+\nu^{\text{lab}}} = 1.$$

Therefore the first part of Theorem 3.1 is a direct application of the Gershgorin circle theorem.

For the second part, denote by a_l the normalized eigenvector, corresponding to λ_l, $l = 1, \dots, \mathbf{n}_1$. Since Q is symmetric, $\{a_l\}$ forms an orthonormal basis of $\mathbb{R}^{\mathbf{n}_1}$ with respect to which s can be written as

$$s = \sum_l \beta_l a_l.$$

Let $\hat{x}' = P_k(Q)s$, while $x' = Q^{-1/2}s$. We deduce

$$\begin{aligned}\|\hat{x} - x\|^2_{Q^{1/2}} &:= \left[\left(P_k(Q) - Q^{-1/2}\right)\sum_l \beta_l a_l\right]^T Q^{1/2}\left(P_k(Q) - Q^{-1/2}\right)\sum_l \beta_l a_l\\ &= \sum_l \left(P_k(\lambda_l) - \frac{1}{\sqrt{\lambda_l}}\right)^2 \beta_l^2 a_l^T Q^{1/2} a_l \leq \varepsilon^2\|s\|^2_{Q^{1/2}}.\end{aligned}$$

The proof is completed. In the last computation we used the definition (15) of the principal square root of Q, which implies that a_l is an eigenvector also of $Q^{-1/2}$ for the eigenvalue $1/\sqrt{\lambda_l}$. □

3.2. Choosing suitable polynomial approximation for $x^{-1/2}$

Theorem 3.1 gives us an (computationally) efficient way to circumvent the numerically expensive explicit computation of $Q^{-1/2}$. Different classes of polynomials can be considered for the error estimation (17) (see for example [39–42]). In [43] it was proved that the principal square root of a nonsingular M-matrix is again an M-matrix, meaning that $Q^{-1/2}$ is a monotone operator. Therefore, for the well-posedness and the meaningfulness of the problem (14), it is important that the approximated vectors $\hat{q}'$, $\hat{s}'$, and $\hat{e}'$ remain non-negative. Indeed, assume for example that $\hat{q}'_i < 0$ for some $i \in U$, then so is its preimage $q'' = Q^{1/2}q'$. Thus, we actually attempt to solve the original problem (12) for the vector q'' with $q''_i < 0$, which is nonsense since q'' cannot serve anymore as a data fidelity measure and it forces $\bar{v}_i \gg 1$. The argument for s' is the same, while in the case of e', we lose the equivalence between $\langle \bar{v}_U, e''\rangle$ and $\|\bar{v}_U\|_1$.

Unfortunately, we cannot theoretically guarantee for any non-linear approximation $P_k(Q)$, $k \geq 2$ element-wise positivity of the entries. What we do in practice is to explicitly check $\hat{q}', \hat{s}', \hat{e}'$, to project them on $\mathbb{R}^{\mathbf{n}_1}_{\geq 0}$ if necessary (when all the negative entries are very small in absolute value), or to change the polynomial P_k (when some of the negative entries are substantially large in absolute value). To prevent this from happening, we avoid as much as possible oscillatory polynomials, because there, even in the univariate setting, computations become instable and untrustworthy, thus the probability to encounter negative values significantly grows. Because of that, even for matrices of known spectrum, we do not recommend the use of interpolatory polynomials $P_k(x)$. Based on our numerical experiments, when $\underline{\lambda}$ is well separated from 0 the polynomials $U_k(\underline{\lambda}, x)$ of degree k of best uniform approximation of $1/\sqrt{x}$ in $[\underline{\lambda}, 1]$ behave well even for $k > 20$. When $\underline{\lambda}$ approaches zero, one may try with the polynomials of best Hausdorff approximation, which are less oscillatory, but for them the sup error $\|P_k(x) - 1/\sqrt{x}\|_\infty$ grows exponentially with $\underline{\lambda}$. In practice, we did not work with $\nu^{\text{lab}} < 1/12$, thus, due to (16), with $\underline{\lambda} < 0.04$.

Using a modified Remez' algorithm from [40] with Hausdorff distance parameter set to zero, we generated the polynomials $U_k(\underline{\lambda}, \cdot)$ for $\underline{\lambda} \in \{0.01, 0.04, 0.05, 0.10, 0.15, \dots, 0.35\}$ and $k \in \{1, 2, \dots, 26, 30, 40, 50, 60\}$. For stable numerical computations we use the Chebyshev basis functions

$$T_0(s) = 1,\ T_1(s) = s, \dots,\ T_k(s) = \cos(k\ \arccos(s))$$

in $[-1, 1]$ and their three-term recurrence relation

$$T_k(s) = 2s * T_{k-1}(s) - T_{k-2}(s). \tag{19}$$

The linear transform that sends $[-1, 1]$ in our interval of interest $[\underline{\lambda}, 1]$ is given via

$$s = \frac{2}{1-\underline{\lambda}} x - \frac{1+\underline{\lambda}}{1-\underline{\lambda}}.$$

Unlike the classical $\{x^k\}$ basis, because of (19), we observe that the coefficients $\{c_\ell\}_0^k$ of $U_k(\underline{\lambda}, s)$ tend to zero monotonically and with alternating sign. The only exception is the leading coefficient c_k, which also has an alternating sign, but we could have $|c_k| > |c_{k-1}|$. For all of the computed polynomials, we have $|c_k| < 2|c_{k-1}|$, guaranteeing $|c_k| = \mathcal{O}(|c_{k-1}|)$. Furthermore, the coefficient sequence of the lower-degree polynomials seem to approximate well the corresponding subsequence of the higher-degree ones, meaning that truncation errors might be harmless. We illustrate it via the following example:

$$\begin{aligned}
U_4(0.04, s) &= 1.92T_0(s) - 1.374T_1(s) + 0.705T_2(s) - 0.398T_3(s) + 0.375T_4(s)\\
U_6(0.04, s) &= 1.92T_0(s) - 1.374T_1(s) + 0.704T_2(s) - 0.396T_3(s) + 0.233T_4(s) - 0.141T_5(s) + 0.142T_6(s)\\
U_{60}(0.04, s) &= 1.92T_0(s) - 1.374T_1(s) + 0.704T_2(s) - 0.396T_3(s) + 0.233T_4(s) - \dots .
\end{aligned}$$

The last two coefficients of $U_{60}(0.04, s)$ are respectively, $c_{59} \approx -1.3 \cdot 10^{-11}$ and $c_{60} \approx 1.6 \cdot 10^{-11}$.

4. ADMM-based algorithm derivation

In this section we address the question how to proceed regarding the box constraint $\mathbf{0} \le Q^{-1/2} v'_U \le \mathbf{1}$ in (14). The lower bound can be reformulated straightforwardly in terms of v'_U due to the monotonicity of $Q^{-1/2}$. It is equivalent to $\mathbf{0} \le v'_U$. For the upper bound, we use Theorem 3.1. Since $\left[\frac{\nu^{\text{lab}}}{2+\nu^{\text{lab}}}, 1\right]$ covers the spectrum of Q, the interval $\left[1, \sqrt{\frac{2+\nu^{\text{lab}}}{\nu^{\text{lab}}}}\right]$ covers the spectrum of $Q^{-1/2}$. Therefore

$$Q^{-1/2}\sqrt{\frac{\nu^{\text{lab}}}{2+\nu^{\text{lab}}}} e \le \bar{\lambda}(Q^{-1/2})\sqrt{\frac{\nu^{\text{lab}}}{2+\nu^{\text{lab}}}} e = e \qquad \Rightarrow \quad \text{if } v'_U \in \left[0, \sqrt{\frac{\nu^{\text{lab}}}{2+\nu^{\text{lab}}}}\right]^{\mathbf{n}_1} \quad \Rightarrow \quad Q^{-1/2} v'_U \subset [0, 1]^{\mathbf{n}_1}.$$

Even though we can only prove inclusion, for $\nu^{\text{pho}} \gg 1$, $Q^{-1/2} v'_U$ is observed to cover most of the unit cube. In this case W^{geo} plays a secondary role in W, thus in the generation of Q, and, as discussed earlier, e approximates closely the eigenvector with respect to $\frac{\nu^{\text{lab}}}{2+\nu^{\text{lab}}}$, implying that $Q^{-1/2}\sqrt{\frac{\nu^{\text{lab}}}{2+\nu^{\text{lab}}}} e \approx e$.

Following [44], we rewrite (14) into its equivalent form

$$\underset{v'_U \in \mathbb{R}^{\mathbf{n}_1}, x \in \mathbb{R}^{3\mathbf{n}_1}}{\operatorname{argmin}} \left\{ \langle 0, v'_U \rangle + \iota_{V_{N_1}(e')}(x_1) + \frac{1}{2}\|x_2\|_2^2 - \langle x_2, q' + s' \rangle + \iota_{\left[0, \sqrt{\frac{\nu^{\text{lab}}}{2+\nu^{\text{lab}}}}\right]^{\mathbf{n}_1}}(x_3) \right\} \quad \text{s.t.} \begin{pmatrix} I \\ I \\ I \end{pmatrix} v'_U = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

The first summand $\langle 0, v'_U \rangle$ may seem redundant, but it allows for the full splitting of the remaining terms so they do not interact directly (only via $x_i = v'_U$, $i = 1, 2, 3$). The convergence rate of the applied Primal–Dual numerical algorithms improves significantly. In this paper, we use the alternating direction methods of multipliers (ADMM).

```
Algorithm 1 (ADMM): Input: $u$ – gray-scale image. $\nu^{\text{pho}}, \nu^{\text{lab}}, h, \alpha^{pho}, s, \ell, N$.
Preprocessing:
  1. Choose $L_1, L_0$.
  2. Generate $W$ and compute $Q$ and $q$.
  3. Choose a polynomial $P_k(x)$ and compute $e' = P_k(Q)e$, $q' = P_k(Q)q$, $s' = P_k(Q)s$.
  4. Project onto $\mathbb{R}^{\mathbf{n}_1}_{\ge 0}$: $e' = \max\{e', \mathbf{0}\}$, $q' = \max\{q', \mathbf{0}\}$, $s' = \max\{s', \mathbf{0}\}$.

Initialization: $q^{(0)}_{1,2,3} = 0$, $x^{(0)}_{1,2,3} = \sqrt{\frac{\nu^{\text{lab}}}{2+\nu^{\text{lab}}}} e$, $\gamma \in (0, 1)$.
For $k = 0, 1, \dots$ repeat until a stopping criterion is reached
  1. $(v'_U)^{(k+1)} = \frac{1}{3}\left(\left(x_1^{(k)} - q_1^{(k)}\right) + \left(x_2^{(k)} - q_2^{(k)}\right) + \left(x_3^{(k)} - q_3^{(k)}\right)\right)$
  2. $x_1^{(k+1)} = q_1^{(k)} + (v'_U)^{(k+1)} + \frac{N_1 - \langle q_1^{(k)} + (v'_U)^{(k+1)}, e' \rangle}{\langle e', e' \rangle} e'$
  3. $x_2^{(k+1)} = \gamma \left( \frac{\gamma}{1-\gamma}(q' + s') + q_2^{(k)} + (v'_U)^{(k+1)} \right)$
```

4. $x_3^{(k+1)} = \min\left\{\sqrt{\frac{\nu^{\mathrm{lab}}}{2+\nu^{\mathrm{lab}}}}\mathbf{1}, \max\left\{\mathbf{0}, q_3^{(k)} + (v_U')^{(k+1)}\right\}\right\}$

5. $q_i^{(k+1)} = q_i^{(k)} + (v_U')^{(k+1)} - x_i^{(k+1)},\ i = 1, 2, 3.$

Postprocessing:

1. $\bar{v}_U = P_k(Q)\bar{v}_U'$.

Output: Segment membership vector with N ones, corresponding to L_1 and the N_1 largest entries of $\bar{v}_U$.

Remarks are in order. Both used constrained sets are convex and closed, while the function $1/2\|\cdot\|_2^2 - \langle\cdot, q' + s'\rangle$ is quadratic, so every summand in the optimization functional is proper, convex, lsc. The proposed algorithm converges for every $\gamma \in (0, 1)$ as long as the optimization functional itself is proper (see [45] for proof), i.e., when

$$V_{N_1}(e') \cap \left[0, \sqrt{\frac{\nu^{\mathrm{lab}}}{2+\nu^{\mathrm{lab}}}}\right]^{\mathbf{n}_1} \neq \emptyset.$$

We cannot theoretically guarantee this in advance, so during the postprocessing stage we check whether $\bar{v}_U'$ belongs to each of the constrained sets. If not, we change the approximating polynomial P_k and start over again. Note, that the classical (scaled) version of ADMM converges for every $\bar{\gamma} > 0$, but for simplicity of the exposition, we used the transformation $\gamma = \bar{\gamma}/(1+\bar{\gamma})$. All steps in the iterating part, but Step 2, are performed voxel-wise and independently of each other, while the only "global" procedures in Step 2 are vector–vector multiplications. Therefore, the main stage of the algorithm allows for efficient parallel implementation. It is implemented and parallelized using OpenMP [46]. The number of non-zero elements per row for the matrix Q is uniformly bounded by $|\mathcal{N}_i^{geo}| + |\mathcal{N}_i^{pho}| + 1 = 10$, while the degree of P_k is commonly taken within the range $k = \{7, 8\}$. Hence, the overall computational complexity of the approximated action of the operator $Q^{-1/2}$ via the use of the operator $P_k(Q)$ is $O(\mathbf{n})$. Combined with the independence of the error estimate (18) from the matrix dimension, the algorithm is applicable to large-scale problems of any size, provided the weight matrix W is efficiently generated or known a priori.

5. Numerical tests

The first test we perform is related to the numerical stability of the matrix–vector computations in $P_k(G)a$. Results for the polynomials $U_k(0.04, s)$, written in Chebyshev basis, are summarized in Table 1. The test matrices $GS = AD^{GS}A^T$ have uniformly distributed spectrum in [0.04, 1], i.e., $d_{i,i}^{GS} = \lambda_i^{GS} = 1 - \frac{24i}{25m}$, m being the size of GS. A random (uniformly sampled in [0, 1]) matrix $\tilde{A}$ of size $m \times m$ has been generated, and then orthonormalized by Gram–Schmidt. The result is saved in A. The m-dimensional vector a is also generated at random, uniformly in [0, 1]. We observe that, when using the three-term recurrence relation of the basis functions, the approximation of $GS^{-1/2}a$ via $U_k(0.04, GS)a$ for polynomial degrees k up to even 40, and matrix sizes m up to 500, behaves in a stable manner and satisfies the theoretical bound (18). We used the notation

$$\varepsilon_k = \sup_{x\in[0.04,1]} |U_k(0.04, x) - x^{-1/2}|, \qquad \text{error} = \frac{\|U_k(0.04, GS)a - GS^{-1/2}a\|_{GS^{1/2}}}{\|a\|_{GS^{1/2}}}.$$

For $k = 60$ and $m = 500$ numerical problems already appear, but in our segmentation algorithm we usually work with $k = 7, 8$, anyway. The matrix GS is not necessary an M-matrix, so we do not check for monotonicity of the approximating operator. Finally, the theoretically proven dimension-invariance of the error (18) in Theorem 3.1 is now numerically verified by the results in Table 1, since the 1D one ε_k and the corresponding $\mathbf{n}$-D ones $\text{error}_{\mathbf{n}}$ have the same order of magnitude for various choices of k and $\mathbf{n}$.

As a continuation of the work in [14,25], we keep on testing an artificially polluted microstructure of a trabecular bone tissue, based on [47]. The true image has size 64 × 64 × 64. 50 604 of its voxels are bone material (porosity 80.7%). It was convoluted with a Gaussian kernel ($\sigma = 2$). Then, 10% white (Gaussian) noise was added to derive the input image u.

First, we test the well-posedness of our optimization problem for the particular setup and what is the impact of the choice of the free weight-generating parameters on it. Results are summarized in Table 2. We work with the proposed in Theorem 3.1 normalization $(1+\nu^{\mathrm{lab}})/(2+\nu^{\mathrm{lab}})$ of W. We consider three options for ν^{lab} in (7), namely $\nu^{\mathrm{lab}} = \{\frac{1}{12}, \frac{2}{9}, \frac{2}{3}\}$, which (see (16)) correspond to spectral bounds $\underline{\lambda} = \{0.04, 0.1, 0.25\}$ for Q, respectively. We measure

$$\text{error}_x = \left\| x - Q\left(U_k(\underline{\lambda}, Q)\left(U_k(\underline{\lambda}, Q)x\right)\right)\right\|_2, \qquad \min_i\{\min\{q_i', e_i'\}\}.$$

The degree k is fixed to 7. Several things are clearly observed. First of all, the computation of e' is numerically stable and less sensitive to the choice of approximating polynomial. This was expected, since by construction we forced e to be "almost"

Table 1
Numerical stability test for the $U_k(0.04, GS)a$ approximation of $GS^{-1/2}a$ for SPD matrices.

k	ε_k	ε_k − error$_{100}$	ε_k − error$_{200}$	ε_k − error$_{300}$	ε_k − error$_{400}$	ε_k−error$_{500}$
7	5.6e−2	4.3e−3	3.8e−3	3.8e−3	5.5e−3	5.8e−3
8	3.6e−2	3.3e−3	2.0e−3	2.0e−3	3.1e−3	3.1e−3
10	1.5e−2	9.2e−4	1.6e−3	7.3e−4	9.4e−4	1.1e−3
20	1.9e−4	1.4e−5	2.5e−5	9.8e−6	1.5e−5	1.9e−5
40	4.1e−8	3.4e−9	3.1e−9	2.9e−9	4.1e−9	4.2e−9
60	1e−11	5.8e−13	7.7e−13	9.7e−13	1.2e−12	−**4.7e−12**

Table 2
Stability and monotonicity test for $U_k(\lambda, Q)q$, $U_k(\lambda, Q)e$ and the operator $U_k(\lambda, Q)$.

$(\underline{\lambda}, k)$	ν^{lab}	$(\nu^{\text{pho}}, \alpha^{pho})$	error$_g$	error$_e$	$\mathbb{R}^{\mathbf{n}_1}_{\geq 0}$check
(0.04,7)	1/12	(1.0, 0.0)	38.7295	29.6586	−0.1046
		(1.0, 5.0)	6.4208	29.7463	−1.767e−2
		(5.0, 0.0)	21 664.2	30.2585	−4.4845
		(5.0, 5.0)	166.275	29.4665	−0.6919
	2/9	(1.0, 0.0)	54.1999	18.3875	−0.1382
		(1.0, 5.0)	8.5345	18.3966	−4.881e−2
		(5.0, 0.0)	105 158	19.0919	−8.3944
		(5.0, 5.0)	260.374	18.3222	−1.4228
	2/3	(1.0, 0.0)	33.7580	30.7764	−0.1132
		(1.0, 5.0)	7.7018	30.7747	−8.207e−2
		(5.0, 0.0)	40 764.1	30.8819	−6.1428
		(5.0, 5.0)	183.855	30.7751	−1.8869
(0.10,7)	1/12	(1.0, 0.0)	4.3222	311.142	−2.203e−2
		(1.0, 5.0)	4.2709	311.145	−1.533e−6
		(5.0, 0.0)	406.487	311.091	−0.6611
		(5.0, 5.0)	4.7062	311.091	−1.529e−2
	2/9	(1.0, 0.0)	4.0933	6.9761	−3.194e−2
		(1.0, 5.0)	4.0235	7.0257	−1.620e−5
		(5.0, 0.0)	554.330	6.8984	−0.4644
		(5.0, 5.0)	4.8503	6.8781	−3.782e−2
	2/3	(1.0, 0.0)	4.4470	9.2459	−7.171e−3
		(1.0, 5.0)	4.0309	9.2478	−4.228e−3
		(5.0, 0.0)	184.237	9.2412	−0.4607
		(5.0, 5.0)	4.2250	9.2370	−7.243e−2
(0.25,7)	1/12	(1.0, 0.0)	4.4422	632.804	0
		(1.0, 5.0)	4.4827	632.804	0
		(5.0, 0.0)	9.2458	632.776	−0.1328
		(5.0, 5.0)	4.5942	632.776	0
	2/9	(1.0, 0.0)	3.7977	88.9234	0
		(1.0, 5.0)	3.8175	88.9255	0
		(5.0, 0.0)	3.9491	88.9016	−3.261e−2
		(5.0, 5.0)	3.8694	88.9019	−9.449e−11
	2/3	(1.0, 0.0)	3.3629	6.1227	0
		(1.0, 5.0)	3.3629	6.1261	−6.549e−8
		(5.0, 0.0)	3.4173	6.1092	−1.3657e−2
		(5.0, 5.0)	3.3638	6.1097	−1.062e−7

an eigenvector of Q, thus any time a matrix–vector operation is executed within the $U_k(\underline{\lambda}, Q)e$ computational process it involves Q and a close-to-constant vector. On the other hand, error$_e$ is significant, when the corresponding eigenvector $\nu^{\text{lab}}/(2 + \nu^{\text{lab}})$ is far away from the interval of approximation $[\underline{\lambda}, 1]$ of U_k. The situation with q' is in a sense the opposite. The vector q is a data fidelity term, hence its entries vary in the whole range of $[0, 1]$ and the coefficient in front of e with respect to its representation in the Q-eigenvector basis is small. Therefore, inaccuracy in the estimation of q' rises not from the polynomial's error estimate ε, but from the possible numerical instability in the matrix–vector operations. The latter increases with $\underline{\lambda}$ decreasing, since $U_k(\underline{\lambda}, \cdot)$ oscillates more in order to simultaneously approximate $x^{-1/2}$ around 0 and around 1. The "positivity check" seems correlated with the numerical instability, so its quantity can be used for the optimization well-posedness. Using large ν^{pho} deepens the problem, while large α^{pho} alleviates it.

Finally, we compare the segmentation vectors of various segmentation processes together with their choice as the additional data fidelity term s in (14). We perform direct (unsupervised) segmentation on u with hard threshold 0.5, direct segmentation on g with hard threshold 0.5, direct volume preserving segmentation on u (i.e., taking the top N pixels with respect to their intensity value), the two segmentation algorithms from [25], and the volume and connectivity preserving algorithm from [14]. As parameters, we use $\nu^{\text{pho}} = 1$, $\alpha^{pho} = 5$, $\nu^{\text{lab}} = 2/9$ with $U_7(0.10, x)$ and $\nu^{\text{lab}} = 1$ with $U_7(0.25, x)$

Table 3
Segmentation comparisons.

	u_{direct}	g_{direct}	u_N	λ-QP [25]	ℓ_2-CP [25]	OST [14]
Pure	91.80%	94.01%	91.69%	94.13%	94.18%	92.05%
As s in (14)	94.18%	94.17%	94.18%	94.17%	94.18%	92.82%

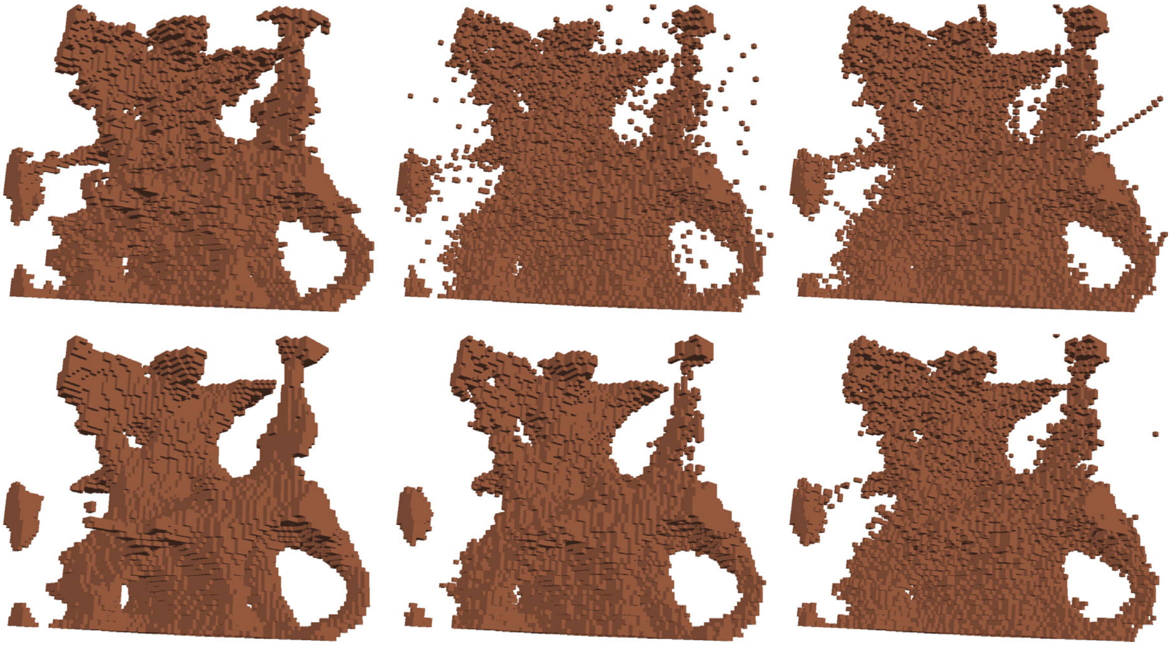

Fig. 1. Top (from left to right): True image; direct volume-constrained segmentation u_N; segmentation u_{OST} based on [14]. Bottom (from left to right): The output of Algorithm 1 for $s = 0$; $s = u_N$; $s = u_{OST}$.

for pure and "hybrid" segmentation, respectively. Quantitatively (see Table 3), with respect to their pure segmentation accuracies (correctly segmented vs. all voxels), we can split the methods into three groups. Non-supervised segmentations are the worst. Connectivity preservation is slightly better, while those, related to the weight matrix W are the best. Their qualitative comparison and interaction in (14) are illustrated in Fig. 1. The binary bone microstructure that we work with is a result of direct segmentation on noisy 3D data, followed by cutting out the secondary connected components. Therefore, as seen top, right, our ground-truth object is not smooth and lacks specific shape properties. The Gaussian blur we add smoothens the bone, and (to a certain extent) improves its quality. Hence, it is unrealistic to expect that we can further improve segmentation accuracy with regularization techniques like graph-Laplacian. When the additional data fidelity term s is non-structured, the smoothing effect of the regularity term prevails and for all s but the one, derived from [14], the "hybrid" outputs of our Algorithm are practically and visually the same. This is not the case when s itself, contains meaningful (different!) information about the object. In particular—when s captures the connectivity of the original. The bottom left image in Fig. 1 nicely illustrates the internal struggle for domination between s and F, regarding the properties of the minimizer v'_U. Most of the noise is removed, but not the whole of it, while the output object (apart from the remaining few noisy voxels) is almost connected and not oversmoothed.

This was exactly our main motivation: to allow for incorporating as much as possible and as diverse as possible information (in this case—physical properties) about the reconstructed object in the segmentation process.

For the second example we use synthetic data, that mimics foam structure. The generated 3D image is again 64 × 64 × 64 and consists of 4 identical vertical blocks, glued together. Each block has two levels of five connected discretized circles, where four of them have large radii and one has a small radius at the bottom, while at the top the situation is vice versa (see Fig. 2). Lines (edges) have thickness of 2 voxels, as illustrated in Fig. 4, up, left. We take the gray-scale intensity of the binary image to be 100 (i.e., we set background to 0, and foreground to 100) and pollute it with Gaussian noise of zero mean, and variance 10, projected on the interval $[-1/2, 1/2]$, then shifted to $[0, 1]$ and scaled 150 times. The input image $u \in [0, 250]^{2^{18}}$ is encoded in an 8-bit bitmap format. The true solid phase contains 46 080 voxels (porosity 82.4%). The image u has almost twice as many (namely 80 864) voxels of intensity 150 (see Fig. 4, top, right), so the optimal threshold value for direct segmentation is $\tau = 151$. Such direct segmentation captures the foam structure but loses almost half of its volume, as only 27 009 voxels have intensity within the range [151, 250].

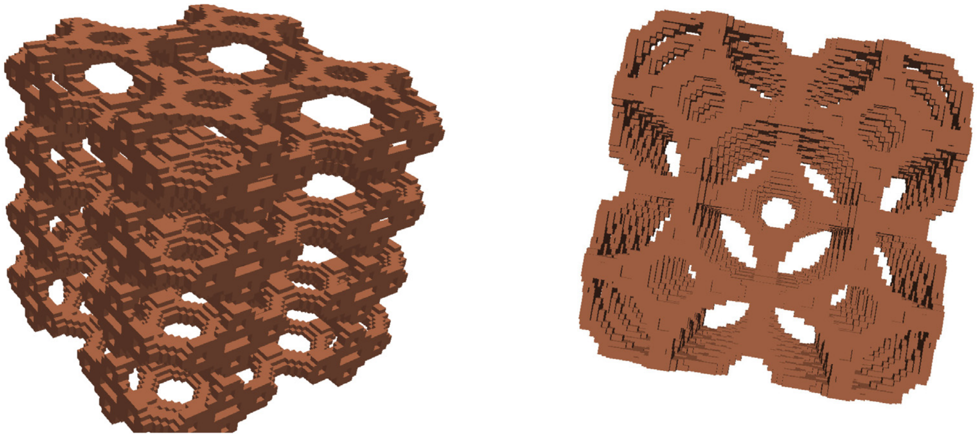

Fig. 2. The $64 \times 64 \times 64$ synthetic foam, viewed from two different angles.

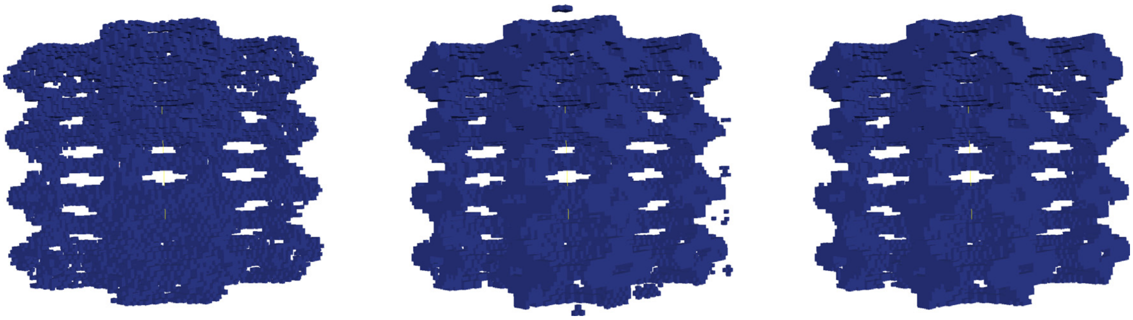

Fig. 3. From left to right: Direct segmentation u_{direct} with threshold $\tau = 151$; The output $u^{(1)}$ of Algorithm 1 for $s = 0$, $(\nu^{\text{pho}}, \nu^{\text{lab}}) = (1, 1/12)$; The output $u^{(2)}$ of Algorithm 1 for $s = u^{(1)}$, $(\nu^{\text{pho}}, \nu^{\text{lab}}) = (1/10, 1/12)$.

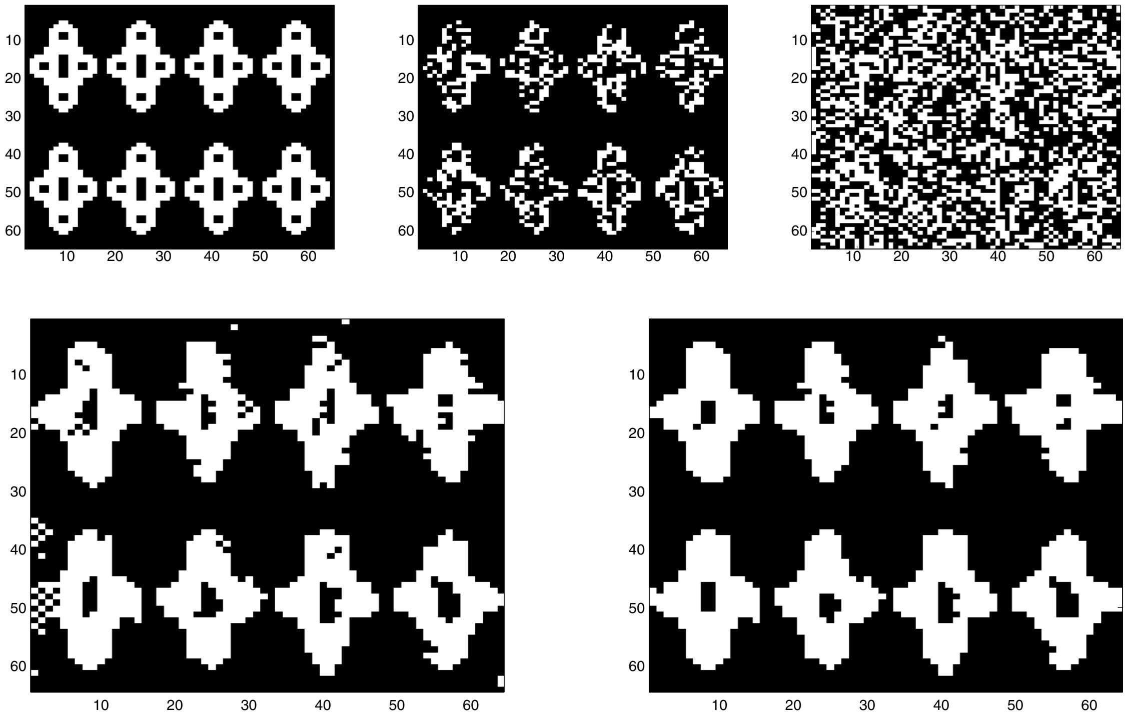

Fig. 4. A slice in the (x, y)-plane of: (top, from left to right) the true (binary) image, the input (noisy) image thresholded with $\tau = 151$, the input image thresholded with $\tau = 150$, and (bottom, from left to right) $u^{(1)}$, $u^{(2)}$ as in Fig. 3.

We first apply Algorithm 1 without additional data fidelity term s. The remaining parameters were chosen as follows: $\nu^{\text{pho}} = 1,\ \nu^{\text{lab}} = 1/12\ (U_7(0.04, x)),\ h = 1,\ \alpha^{pho} = 1,\ \ell, m = 3$. The output $u^{(1)}$ not only preserves the volume but also captures the structure, due to the dominant role of the weights w^{pho} in the matrix W. For such parameter choice, the impact of the spatial metric is small and the result is not connected. However, we can apply Algorithm 1 second time with $s = u^{(1)}$ and ν^{pho} changed to 0.1. The output $u^{(2)}$ is now connected.

Note that qualitatively $u^{(2)}$ is not better than $u^{(1)}$, since $u^{(1)}$ misdiagnosed 4924 of the material voxels, while $u^{(2)}$ failed on 4953. On the other hand, for this particular example we forced spatial connectivity on the solid phase at a very low cost. Considering the setup, the segmentation accuracy of both the outputs is very high (above 90% on the solid phase alone and above 96% overall), which is another attestation for the robustness of the weight class (6) against Gaussian noise.

6. Conclusions

Applying a convenient change of basis in the image domain, we decomposed the graph-Laplacian operator in (12) element-wise and came up with the fully split optimization problem (14). Carefully constructing the weight matrix W and using univariate polynomial algebra, we showed that the dense, large-scale transformation matrix $Q^{-1/2}$ need not be explicitly computed and its action on any vector s can be well approximated by the matrix polynomial $U_k(\underline{\lambda}, Q)$, where $U_k(\underline{\lambda}, \cdot)$ is the element of best uniform approximation of degree k to the function $x^{-1/2}$ over the interval $[\underline{\lambda}, 1]$ that covers the spectrum of Q. The approximation error is dimension-invariant, which together with the parallelizability of the proposed ADMM numerical algorithm for solving (14), allows us to attack large-scale problems. We also can include additional linear terms in the optimization function and keep control on all the constraints as long as their intersection is nonempty. The latter allows for theoretical and numerical investigation of “hybrid” segmentation models, where several segmentation vectors interact and correct each other in order to improve the quality of the reconstruction. Of particular interest is the combination of the volume constrained segmentation models [14,25], as they incorporate different physical properties of the reconstructed object (smooth surface vs. spatial connectivity). It remains unclear if such a collaboration can be mutually beneficial, or on the contrary, will neither denoise nor deblur the result. This will be a topic of a future research.

Acknowledgment

This work was partially supported by the “Young Scientists” Grant No. DFNP-92/04.05.2016, issued by the Bulgarian Academy of Sciences.

References

[1] D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, Comm. Pure Appl. Math. 42 (5) (1989) 577–685.

[2] T. Pock, D. Cremers, H. Bischof, A. Chambolle, An algorithm for minimizing the Mumford-Shah functional, in: Proc. IEEE 12th Conf. Comput. Vis, IEEE, 2009, pp. 1133–1140.

[3] X. Cai, R. Chan, T. Zeng, A two-stage image segmentation method using a convex variant of the Mumford–Shah model and thresholding, SIAM J. Imag. Sci. 6 (1) (2013) 368–390.

[4] T. Chan, L. Vese, Active contours without edges, IEEE Trans. Image Process. 10 (2) (2001) 266–277.

[5] L. Vese, T. Chan, A multiphase level set framework for image segmentation using the Mamford and Shah model, Int. J. Comput. Vis. 50 (3) (2002) 271–293.

[6] T. Chan, S. Esedolgu, M. Nikolova, Algorithms for finding global minimizers of image segmentation and denoising models, SIAM J. Appl. Math. 66 (5) (2006) 1632–1648.

[7] X. Bresson, S. Esedoglu, P. Vandergheynst, J. Thiran, S. Osher, Fast global minimization of the active contour/snake model, J. Math. Imaging Vision 28 (2) (2007) 151–167.

[8] Y. Zhang, B. Matuszewski, L. Shark, C. Moore, Medical image segmentation using new hybrid level-set method, in: 5th Int. Conf. BioMedical Visualization, MEDIVIS'08, IEEE, 2008, pp. 71–76.

[9] B. Dong, A. Chien, Z. Shen, Frame based segmentation for medical images, Commun. Math. Sci. 32 (2010) 1724–1739.

[10] X. Cai, G. Steidl, Multiclass segmentation by iterated ROF thresholding, in: Energy Minimization Methods in Computer Vision and Pattern Recognition, LNCS 8081, Springer, 2013, pp. 237–250.

[11] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1–4) (1992) 259–268.

[12] C.T. Zahn, Graph-theoretic methods for detecting and describing gestalt clusters, IEEE Trans. Comput. 20 (1971) 68–86.

[13] R. Urquhart, Graph theoretical clustering based on limited neighborhood sets, Pattern Recognit. 15 (3) (1982) 173–187.

[14] S. Harizanov, S. Margenov, L. Zikatanov, Fast constrained image segmentation using optimal spanning trees, in: 10th International Conference on Large-Scale Scientific Computations, LNCS 9374, Springer, 2015, pp. 15–29.

[15] Z. Wu, R. Leahy, An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 11 (1993) 1101–1113.

[16] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.

[17] C. Rother, V. Kolmogorov, A. Blake, “GrabCut”– interactive foreground extraction using iterated graph cuts, ACM Trans. Graph. 23 (3) (2004) 309–314.

[18] Y. Boykov, M.-P. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, in: International Conference on Computer Vision, vol. I, 2001, pp. 105–112.

[19] Y. Weiss, Segmentation using eigenvectors: A unifying view, in: Proc. IEEE 17th Conf. Comput. Vis, Vol. 2, IEEE, 1999, pp. 975–982.

[20] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient graph-based image segmentation, Int. J. Comput. Vis. 59 (2) (2004) 167–181.

[21] S.H. Kang, B. Shafei, G. Steidl, Supervised and transductive multi-class segmentation using *p*-Laplacians and RKHS methods, J. Vis. Commun. Image Represent. 25 (5) (2014) 1136–1148.

[22] G. Gilboa, S. Osher, Nonlocal linear image regularization and supervised segmentation, SIAM Multiscale Model. Simul. 6 (2) (2007) 595–630.

[23] U. von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.

[24] N.Y. Law, H.K. Lee, M.K. Ng, A.M. Yip, A semisupervised segmentation model for collections of images, IEEE Trans. Image Process. 21 (6) (2012) 2955–2968.

[25] I. Georgiev, S. Harizanov, Y. Vutov, Supervised 2-phase segmentation of porous media with known porosity, in: 10th International Conference on Large-Scale Scientific Computations, LNCS 9374, Springer, 2015, pp. 343–351.

[26] S. Amghibech, Eigenvalues of the discrete p-Laplacian for graphs, Ars Combin. 67 (2003) 283–302.

[27] T. Bühler, M. Hein, Spectral clustering based on the graph p-Laplacian, in: Proceedings of the 26th Ann. Intern. Conference on Machine Learning, 2009, pp. 81–88.

[28] J. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Comput. Geosci. 10 (2–3) (1984) 191–203.

[29] Y. He, B. Shafei, M.Y. Hussaini, J. Ma, G. Steidl, A new fuzzy c-means method with total variation regularization for segmentation of images with noisy and incomplete data, Pattern Recognit. 45 (2012) 3436–3471.

[30] A. Buades, B. Coll, J.-M. Morel, On image denoising methods, SIAM Multiscale Model. Simul. 4 (2) (2005) 490–530.

[31] G. Gilboa, S. Osher, Nonlocal operators with applications to image processing, SIAM Multiscale Model. Simul. 7 (3) (2008) 1005–1028.

[32] B.K. Natarajan, Sparse approximate solutions to linear systems, SIAM J. Comput. 24 (2) (1995) 227–234.

[33] B. Iannazzo, On the Newton method for the matrix pth root, SIAM J. Matrix Anal. Appl. 28 (2) (2006) 503–523.

[34] B. Iannazzo, A family of rational iterations and its application to the computation of the matrix pth root, SIAM J. Matrix Anal. Appl. 30 (4) (2008–09) 1445–1462.

[35] N.J. Higham, L. Lin, A Schur-Padé algorithm for fractional powers of a matrix, SIAM J. Matrix Anal. Appl. 32 (3) (2011) 1056–1078.

[36] N.J. Higham, L. Lin, An improved Schur-Padé algorithm for fractional powers of a matrix and their Fréchet derivatives, SIAM J. Matrix Anal. Appl. 34 (3) (2013) 1341–1360.

[37] P.N. Vabishchevich, Operator-difference scheme with a factorized operator, in: 10th International Conference on Large-Scale Scientific Computations, LNCS 9374, Springer, 2015, pp. 72–79.

[38] A. Bonito, J. Pasciak, Numerical approximation of fractional powers of elliptic operators, Math. Comp. 84 (295) (2015) 2083–2110.

[39] B. Sendov, V. Popov, Numerical Methods - Part I, Sofia University Publishing "St. Kliment Ohridski", 1996, (in Bulgarian).

[40] P. Marinov, A. Andreev, A modified Remez' algorithm for approximate determination of the rational function of the best approximation in Hausdorff metric, C. R. Acad. Bulgare Sci. 40 (3) (1987) 13–16.

[41] P. Marinov, Approximate determination of the generalized polynomial of the best one-sided Hausdorff approximation, Math. Balkanica, New Ser. vol. 3 (1) (1989) 97–105.

[42] P. Marinov, A. Andreev, P. Yalamov, Approximate determination of the rational function of the best restricted approximation in Hausdorff metric, Serdica 19 (1993) 59–65.

[43] G. Alefeld, N. Schneider, On square roots of M-matrices, Linear Algebra Appl. 42 (1982) 119–132.

[44] T. Teuber, G. Steidl, R.H. Chan, Minimization and parameter estimation for seminorm regularization models with I-divergence constraints, Inverse Problems 29 (3) (2013) 035007.

[45] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122.

[46] L. Dagum, R. Menon, OpenMP: an industry-standard API for shared-memory programming, IEEE Comput. Sci. Eng. 5 (1) (1998) 46–55.

[47] G. Beller, M. Burkhart, D. Felsenberg, W. Gowin, H.-C. Hege, B. Koller, S. Prohaska, P. Saparin, J. Thomsen, Vertebral Body Data Set ESA29-99-L3[link]. URL http://bone3d.zib.de/data/2005/ESA29-99-L3/.