Fast Inversion of Matrices Arising in Image Processing

Bernd Fischer Jan Modersitzki*

December-23, 1998

Abstract

In recent years, new nonlinear partial differential equation (PDE) based approaches have become popular for solving image processing problems. Although the outcome of these methods is often very promising, their actual realization is in general computationally intensive. Therefore accurate and efficient schemes are needed. The aim of this paper is two folded. First, we will show that the three dimensional alignment problem of a histological data set of the human brain may be phrased in terms of a nonlinear PDE. Second, we will devise a fast direct solution technique for the associated structured large systems of linear equations. In addition, the potential of the derived method is demonstrated on real-life data.

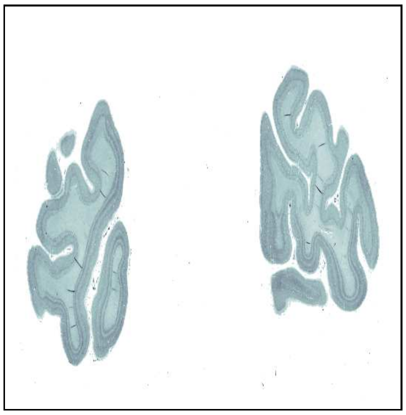
Keywords: Image registration, Elastic matching, FFT
AMS (MOS) classification: 68U10

1 Introduction

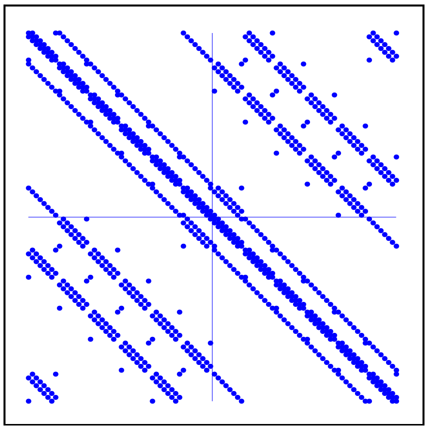
A typical image registration problem is the computation of a mapping that aligns two given images. To illustrate this problem, consider the images in Figure 1, which are pictures of two consecutive histological sections of a human brain. A three dimensional reconstruction of the histological data set (about 7000 sections per brain) will suffer from nonlinear distortions introduced by the preparation process, as it is apparent from the difference plot of theses images (see Figure 1).

Two different approaches for correcting this kind of distortions are common. One approach is based on the idea of representing the unknown correction in terms of the coefficients of a fixed basis, such as piecewise linear functions or higher order splines. Typically these coefficients are determined by a least squares condition for some user prescribed landmarks (see, e.g.,

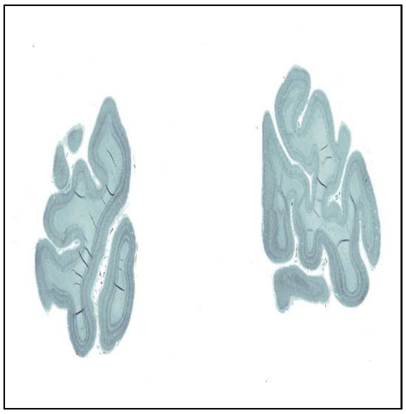
*Medical University of Lübeck, Wallstraße 40, 23560 Lübeck, Germany,
Email: {fischer,modersitzki}@math.mu-luebeck.de



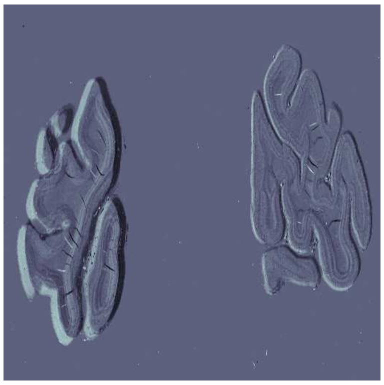
Reference R.



Nonzero Pattern of A for $M = 8$ and $N = 6$.



Template T.



Difference $T - R$, $\|T - R\|_2 \approx 10361$.

Figure 1: Sections from a human brain (512×512 pixel), difference plot, and spy of matrix A.

[3, 4]). The second approach is based on the formulation of the problem via a nonlinear PDE. To solve these equations no further information on the underlying images, e.g. landmarks, is needed (see, e.g. [1, 12]).

Here we follow the PDE approach. To model the behavior of a human tissue, it is assumed that the distortions result from elastic deformations of the tissue. For this reason, the resulting scheme is often called elastic matching (see, e.g., [2, 5, 6, 11]).

To derive the associated PDE, let $T, R : \Omega \to \mathbb{R}$ represent two images, e.g. $T(\mathbf{x})$, $R(\mathbf{x})$ are the grey or color values of the respective images at the point $\mathbf{x} = (x, y) \in \Omega$. Then the problem is to find a displacement field $\mathbf{u}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))$, such that the difference $T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x})$ is as small as possible.

An obvious method for computing the unknown correction $\mathbf{u}$ would be the minimization of the Euclidean distance between the template T and the reference R,

$$D(\mathbf{u}) = \int_\Omega \Big(T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x})\Big)^2 \, d\mathbf{x}.$$

However, the computed transformation $\mathbf{u}$ should be smooth such that for example connected regions of the tissue remain connected. Therefore we introduce a smoothing term which penalizes non smooth functions. Thus, we require $\mathbf{u}$ to minimize as well the strain energy $F(\mathbf{u})$ given by the elastic potential

$$F(\mathbf{u}) = F(u, v) = \int_\Omega \frac{\lambda}{2}(u_x + v_y)^2 + \mu\Big(u_x^2 + v_y^2 + \frac{1}{2}(u_y + v_x)^2\Big) \, d\mathbf{x},$$

where the so-called Lamé constants λ and μ are determined by the underlying material (see, e.g., [9]). Now, the resulting functional looks like

$$D(\mathbf{u}) + \alpha F(\mathbf{u}). \tag{1}$$

Finally, an application of the Euler-Lagrange calculus leads to a necessary condition for $\mathbf{u}$ being a minimizer of eq. (1):

$$\left.\begin{array}{rcl} \mathbf{f}(\mathbf{u}) & = & \Big(T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x})\Big)\nabla T(\mathbf{x} - \mathbf{u}(\mathbf{x})), \\ \mathbf{0} & = & \mathbf{f}(\mathbf{u}) + \mu\Delta\mathbf{u} + (\lambda + \mu)\nabla \operatorname{div} \mathbf{u}, \end{array}\right\} \tag{2}$$

where, for simplicity we used the notation λ, μ instead of $\alpha\lambda$ and $\mu\alpha$, respectively. The obtained nonlinear PDE is known as Navier-Lamé equation (NLE).

The main step in the numerical solution of eq. (2) is the repeated solution of large systems of linear equations. For example, a standard discretization of an image of 512×512 pixel results in systems with 524288 unknowns.

It should come as no surprise, that the associated matrices have a rich structure. It will turn out that, for the case of periodic boundary conditions, these matrices are roughly speaking block-circulant with circulant blocks. In Section 2 we show how to explicitly invert such matrices. In Section 3 we comment on implementation details and provide an illustrative real-life example for the derived method.

2 Fast Inversion of a Structured Matrix

A proper discretization of the NLE (2) yields a system of linear equations where the system matrix A has a rich structure (see Figure 1 for the nonzero pattern). As we show in Section 3, for periodic boundary conditions, the matrix A is essentially block-circulant.

In this section, we identify a class of matrices which may be diagonalized by FFT type techniques. As it turns out, the matrix A belongs to this class.

We start be defining this class. To this end let

$$C_n := \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 1 \\ 1 & 0 & \cdots & \cdots & 0 \end{pmatrix} \in \mathbb{C}^{n\times n}.$$

denote the basic circulant matrix. For ease of presentation we frequently make use of the Kronecker-product $\otimes$ of two matrices $B \in \mathbb{C}^{m\times n}$, $C \in \mathbb{C}^{p\times q}$, defined as

$$\begin{pmatrix} b_{1,1} \cdots & b_{1,n} \\ \vdots & \vdots \\ b_{m,1} \cdots & b_{m,n} \end{pmatrix} \otimes C := \begin{pmatrix} b_{1,1}C \cdots & b_{1,n}C \\ \vdots & \vdots \\ b_{m,1}C \cdots & b_{m,n}C \end{pmatrix} \in \mathbb{C}^{(mp)\times(nq)}.$$

Now we are in position to define the class of matrices in question. More precisely, let $A \in \mathbb{R}^{(2mn)\times(2mn)}$ be a 2-by-2 block matrix with block-circulant matrices

$$A = \begin{pmatrix} A^1 & A^2 \\ A^3 & A^4 \end{pmatrix} \in \mathbb{R}^{(2mn)\times(2mn)}, \tag{3}$$

where

$$A^p = (C_n)^{-1} \otimes A_3^p + (C_n)^0 \otimes A_2^p + (C_n)^1 \otimes A_1^p \tag{4}$$

$$= \begin{pmatrix} A_2^p & A_1^p & & A_3^p \\ A_3^p & \ddots & \ddots & \\ & \ddots & A_2^p & A_1^p \\ A_1^p & & A_3^p & A_2^p \end{pmatrix} \in \mathbb{R}^{(mn)\times(mn)},$$

and

$$A_q^p = S_{3,q}^p (C_m)^{-1} + S_{2,q}^p (C_m)^0 + S_{1,q}^p (C_m)^1 \tag{5}$$

$$= \begin{pmatrix} S_{2,q}^p & S_{1,q}^p & & S_{3,q}^p \\ S_{3,q}^p & \ddots & \ddots & \\ & \ddots & S_{2,q}^p & S_{1,q}^p \\ S_{1,q}^p & & S_{3,q}^p & S_{2,q}^p \end{pmatrix} \in \mathbb{R}^{m\times m}.$$

The matrices

$$S^p = \begin{pmatrix} S_{1,1}^p & S_{1,2}^p & S_{1,3}^p \\ S_{2,1}^p & S_{2,2}^p & S_{2,3}^p \\ S_{3,1}^p & S_{1,2}^p & S_{3,3}^p \end{pmatrix}, \quad p = 1,2,3,4,$$

are given and resemble in our application the constituent matrix stencils of the underlying discretization.

A main ingredient of our analysis is the fact that any circulant matrix may be diagonalized by a Fourier matrix, a proof of which may be found for example in [7].

Lemma 1 *Let $\omega_n := \exp(-2\pi i/n)$ be a root of unity and $F_n \in \mathbb{C}^{n\times n}$ be a Fourier matrix,*

$$F_n := \frac{1}{\sqrt{n}} (\omega_n^{(j-1)(k-1)})_{j,k=1,\ldots,n}, \quad \Omega_n := \operatorname{diag}\left(\omega_n^0, \ldots, \omega_n^{n-1}\right).$$

1. *The Fourier matrix is unitary, $F_n^{-1} = F_n^H$.*
2. *The circulant matrix C_n is diagonalized by F_n, $F_n^H C_n F_n = \Omega_n$.*
3. *Any circulant matrix $Z_n = \sum_{j=-n}^{n} \alpha_j (C_n)^j$ is diagonalized by F_n,*

$$F_n^H Z_n F_n = \sum_{j=-n}^{n} \alpha_j (\Omega_n)^j.$$

Next, we show how to factorize the block matrix A.

Lemma 2 *Let $p = 1,2,3,4,$ $q = 1,2,3,$ $j = 1,\ldots,m,$ and $k = 1,\ldots,n$.*

1. *The matrices $A_q^p \in \mathbb{R}^{m\times m}$ are diagonalized by F_m,*

$$F_m^H A_q^p F_m = \operatorname{diag}\left(\lambda_{q,j}^p,\ j = 1,\ldots,m\right),$$

where $\lambda_{q,j}^p = S_{2,q}^p + S_{3,q}^p \overline{\omega_m}^{j-1} + S_{1,q}^p \omega_m^{j-1}$.

2. *The matrices $A^p \in \mathbb{R}^{(mm)\times(mn)}$ are diagonalized by $F_n \otimes F_m$,*

$$\begin{aligned} & (F_n \otimes F_m)^{H} A^p (F_n \otimes F_m) \\ = & \operatorname{diag}\left(D^p_{k,j},\ j=1,\dots,m,\ k=1,\dots,n\right), \end{aligned}$$

where $D^p_{k,j} = \lambda^p_{2,j} + \overline{\omega_n}^{k-1}\lambda^p_{3,j} + \omega_n^{k-1}\lambda^p_{1,j}$.

3. *For the matrix $A \in \mathbb{R}^{(2mm)\times(2mn)}$ we have*

$$F^{H} A F = \begin{pmatrix} \operatorname{diag}\left(D^1\right) & \operatorname{diag}\left(D^2\right) \\ \operatorname{diag}\left(D^3\right) & \operatorname{diag}\left(D^4\right) \end{pmatrix},$$

where

$$F = \begin{pmatrix} F_n \otimes F_m & \mathbf{0} \\ \mathbf{0} & F_n \otimes F_m \end{pmatrix}$$

and $D^p = \operatorname{diag}\left(D^p_{j,k},\ j=1,\dots,m,\ k=1,\dots,n\right)$.

Proof

1. Follows directly from Lemma 1.

2. From

$$\begin{aligned} B^p &:= (E \otimes F_m)^{\mathrm{H}} A^p (E \otimes F_m) \\ &= (C_n)^{-1} \otimes (F_m^{\mathrm{H}} A_3^p F_m) \\ &\qquad + (C_n)^0 \otimes (F_m^{\mathrm{H}} A_2^p F_m) + (C_n)^1 \otimes (F_m^{\mathrm{H}} A_1^p F_m), \end{aligned}$$

we deduce that B^p is a block-circulant matrix with diagonal blocks. A suitable permutation P yields

$$P^{-1} B^p P = \operatorname{diag}\left(L_1^p, \dots, L_n^p\right),$$

where $L_j^p = (C_n)^{-1}\lambda^p_{3,j} + (C_n)^0\lambda^p_{2,j} + (C_n)^1\lambda^p_{1,j} \in \mathbb{R}^{n\times n}$. Finally, L_j^p may be diagonalized by F_n.

3. The matrices A^p are simultaneously diagonalizable.

□

Our goal is to actually invert $F^{\mathrm{H}}AF$. The trouble is that the matrix A might be singular (and for particular choices of S^p we are interested in, it is). Therefore we compute the so-called Moore-Penrose pseudo inverse [8].

We start by calculating the Moore-Penrose pseudo inverse $B^\dagger$ of a 2-by-2 matrix B.

Lemma 3 *For the symmetric matrix $B = \begin{pmatrix} b^1 & b^2 \\ b^2 & b^4 \end{pmatrix} \in \mathbb{R}^{2\times 2}$ we have*

$$B^\dagger = \begin{cases} B^{-1}, & \text{for } \det(B) \neq 0, \\ \frac{1}{(b^1+b^4)^2} \cdot B, & \text{for } \det(B) = 0,\ b^1 \neq 0 \vee b^4 \neq 0, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

Proof

1. If $\det(B) \neq 0$, then $B^\dagger = B^{-1}$.

2. Let $\det(B) = 0$ but $b^1 \neq 0 \vee b^4 \neq 0$. Then B has a singular value decomposition $B = Q \operatorname{diag}\left(b^1 + b^4, 0\right) Q^{\mathrm{T}}$, where

$$
\begin{aligned}
Q &= \frac{1}{\sqrt{(b^1)^2+(b^2)^2}} \begin{pmatrix} b^1 & b^2 \\ b^2 & -b^1 \end{pmatrix} \quad \text{for} \quad b^1 \neq 0, \\
Q &= \frac{1}{\sqrt{(b^4)^2+(b^2)^2}} \begin{pmatrix} b^2 & b^4 \\ b^4 & -b^2 \end{pmatrix} \quad \text{for} \quad b^1 = 0.
\end{aligned}
$$

   Hence, the pseudo inverse is given by $B^\dagger = \frac{1}{(b^1+b^4)^2} B$.

3. Let $\det(B) = 0$ and $b^1 = b^4 = 0$. Thus, $B = \mathbf{0}$ and $B^\dagger = \mathbf{0}$.

□

We are now ready to present our main result.

Theorem 1 *Let S^p be such that D^p is real and let $D^2 = D^3$. Then $A^\dagger = F D^\dagger F^H$, where*

$$
D^\dagger := \begin{pmatrix} \operatorname{diag}\left(D^{1,\dagger}\right) & \operatorname{diag}\left(D^{2,\dagger}\right) \\ \operatorname{diag}\left(D^{2,\dagger}\right) & \operatorname{diag}\left(D^{4,\dagger}\right) \end{pmatrix}
$$

and

$$
\begin{pmatrix} D^{1,\dagger}_{j,k} & D^{2,\dagger}_{j,k} \\ D^{2,\dagger}_{j,k} & D^{4,\dagger}_{j,k} \end{pmatrix} = \begin{pmatrix} D^1_{j,k} & D^2_{j,k} \\ D^2_{j,k} & D^4_{j,k} \end{pmatrix}^\dagger .
$$

Proof The starting point is Lemma 2.3). A suitable permutation matrix P yields

$$
P^{-1} \begin{pmatrix} \operatorname{diag}\left(D^1\right) & \operatorname{diag}\left(D^2\right) \\ \operatorname{diag}\left(D^2\right) & \operatorname{diag}\left(D^4\right) \end{pmatrix} P = \operatorname{diag}\left(B_{i,j}, i, \ldots, m,\ j = 1, \ldots, n\right),
$$

where $B_{i,j} = \begin{pmatrix} D^1_{i,j} & D^2_{i,j} \\ D^2_{i,j} & D^4_{i,j} \end{pmatrix} \in \mathbb{R}^{2\times 2}$. The pseudo inverse of B is given by Lemma 3.

□

3 Solving the Navier-Lamé Equation

In this section we deduce a discrete version of the NLE (2). Furthermore we show that the obtained coefficient matrix belongs to the class of matrices discussed in the previous section. Finally, we present a numerical example based on real life data.

To start with, we rewrite eq. (2) as follows

$$\begin{aligned} f &= \mu(u_{xx} + u_{yy}) &+& \quad (\lambda+\mu)(u_{xx} + v_{xy}), \\ g &= \mu(v_{xx} + v_{yy}) &+& \quad (\lambda+\mu)(u_{xy} + v_{yy}), \end{aligned}$$

where

$$\begin{aligned} f &= \Big(T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x})\Big) \cdot T_x(\mathbf{x} - \mathbf{u}(\mathbf{x})), \\ g &= \Big(T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x})\Big) \cdot T_y(\mathbf{x} - \mathbf{u}(\mathbf{x})). \end{aligned}$$

Note, that the lefthandside (f, g) may be interpreted as a force field and depends nonlinear on $\mathbf{u}$.

To discretize the PDE we employ finite differences. Actually, for the approximation of the derivatives we used standard stencils, e.g.

$$\begin{aligned} u_{i,j} &= u(i,j), \\ u_{xx}(i,j) &\approx u_{i+1,j} - 2u_{i,j} - u_{i-1,j}, \\ u_{xy}(i,j) &\approx (u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1})/4, \\ T_x(i,j) &\approx (T_{i+1,j} - T_{i-1,j})/2. \end{aligned}$$

We remark that the mesh size is chosen to be one, which is common in image processing applications.

With this particular approximations we obtain, for example, for the first part of the NLE

$$\begin{aligned} & \mu\Big(u_{xx}(i,j) + u_{yy}(i,j)\Big) + (\lambda+\mu)\Big(u_{xx}(i,j) + v_{xy}(i,j)\Big) \\ \approx\ & S^1 * u_{i,j} + S^2 * v_{i,j}. \end{aligned}$$

Here, the symbol $*$ denotes the convolution operator

$$S^p * u_{i,j} := \sum_{r,s=-1}^{1} S^p_{2+r,2+s} u_{i+r,j+s}.$$

Moreover, the matrix stencils look like

$$S^1 = (S^4)^{\mathrm{T}} = \begin{pmatrix} & 2\mu+\lambda & \\ \mu & -2(3\mu+\lambda) & \mu \\ & 2\mu+\lambda & \end{pmatrix}, \tag{6}$$

$$S^2 = S^3 = \frac{1}{4}\begin{pmatrix} \mu+\lambda & & -(\mu+\lambda) \\ & & \\ -(\mu+\lambda) & & \mu+\lambda \end{pmatrix}. \tag{7}$$

Next, we have to specify appropriate boundary conditions. In all our applications, the tissue of the human brain is embedded in paraffin wax. This results in images which have a wide boundary of homogeneous background color (cf. Figure 1). Therefore one may expect that the effect of different

Table 1: Algorithm for solving the NLE (2).

1. Choose $(u_{i,j})^{(0)}$, $(v_{i,j})^{(0)}$, e.g. $(u_{i,j})^{(0)} = (v_{i,j})^{(0)} = \mathbf{0}$.
2. For $k = 0, 1, 2, \ldots,$

   compute actual forces

$$\begin{aligned} (f_{i,j})^{(k)} &:= f((u_{i,j})^{(k)}, (v_{i,j})^{(k)}), \\ (g_{i,j})^{(k)} &:= g((u_{i,j})^{(k)}, (v_{i,j})^{(k)}), \end{aligned}$$

   solve the linear system

$$A((u_{i,j})^{(k+1)}, (v_{i,j})^{(k+1)})^{\mathrm{T}} = ((f_{i,j})^{(k)}, (g_{i,j})^{(k)})^{\mathrm{T}}.$$

boundary conditions on the overall process is neglectable. This expectation is supported by the example given at the end of this section (cf. Figure 2).

Here, for computational convenience we choose periodic boundary conditions

$$\begin{aligned} u_{1,j} &= u_{M-1,j}, & u_{M,j} &= u_{2,j}, \\ u_{i,1} &= u_{i,N-1}, & u_{i,N} &= u_{i,2}, \\ v_{1,j} &= v_{M-1,j}, & v_{M,j} &= v_{2,j}, \\ v_{i,1} &= v_{i,N-1}, & v_{i,N} &= v_{i,2}, \end{aligned}$$

for an image with $M \times N$ pixel.

It is straightforward to verify that this results in the following matrix version of the NLE

$$A \begin{pmatrix} u_{i,j} \\ v_{i,j} \end{pmatrix} := \begin{pmatrix} A^1 & A^2 \\ A^3 & A^4 \end{pmatrix} \begin{pmatrix} u_{i,j} \\ v_{i,j} \end{pmatrix} := \begin{pmatrix} S^1 * u_{i,j} + S^2 * v_{i,j} \\ S^3 * u_{i,j} + S^4 * v_{i,j} \end{pmatrix} = \begin{pmatrix} f_{i,j} \\ g_{i,j} \end{pmatrix},$$

where we used the notation $(u_{i,j})$ for vector s $(u_{2,2}, u_{3,2}, \ldots, u_{M-1,N-1}) \in \mathbb{R}^{(M-2)(N-2)}$.

The obtained linear system is the main ingredient of a fix point type algorithm for the solution of the NLE (2). A straightforward implementation is given in Table 1.

The main computational work in each iteration is the solution of the linear system. In view of Theorem 1 this may be done by explicitly computing the pseudo inverse of the block diagonal D and applying four FFT's. The implementation details for this process are listed in Table 2.

We note that for the particular choices of S^p, cf. eqn. (6) and (7), the computation of $D^\dagger$ simplifies considerably

$$\begin{aligned} D^1_{i,j} &= S^1_{2,2} + 2\cos(\frac{2\pi i}{M-1})S^1_{3,2} + 2\cos(\frac{2\pi j}{N-1})S^1_{2,3}, \\ D^4_{i,j} &= S^1_{2,2} + 2\cos(\frac{2\pi j}{N-1})S^1_{3,2} + 2\cos(\frac{2\pi i}{M-1})S^1_{2,3}, \end{aligned}$$

Table 2: Algorithm for solving the linear system (using a standard code for the 2D FFT).

Compute $(\widetilde{f}_{i,j}) := \texttt{fft2}(f_{i,j})$, $(\widetilde{g}_{i,j}) := \texttt{fft2}(g_{i,j})$.
For $j = 2, \ldots, N-1$, for $i = 2, \ldots, M-1$ do
$\widetilde{u}_{i,j} = D^{1,+}_{i,j}\widetilde{f}_{i,j} + D^{2,+}_{i,j}\widetilde{g}_{i,j}$,
$\widetilde{v}_{i,j} = D^{2,+}_{i,j}\widetilde{f}_{i,j} + D^{4,+}_{i,j}\widetilde{g}_{i,j}$.
End.
Compute $(u_{i,j}) := \texttt{fft2}^{-1}(\widetilde{u}_{i,j})$, $(v_{i,j}) := \texttt{fft2}^{-1}(\widetilde{v}_{i,j})$.

$$D^2_{i,j} = D^3_{i,j} \quad = \quad -4S^2_{1,1}\sin(\frac{2\pi i}{M-1})\sin(\frac{2\pi j}{N-1}).$$

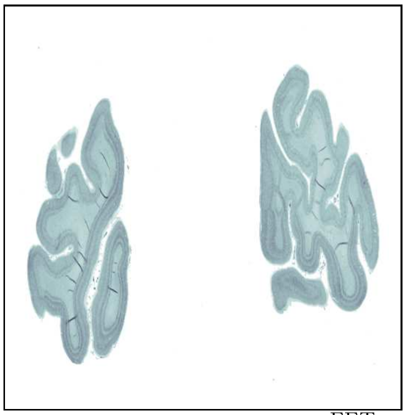
Finally, in Figure 2 we present a numerical example based on medical data. We display the result of the elastic matching of the two histological sections shown in Figure 1. The top part of Figure 2 shows the corrected template and the corresponding difference plot obtained after 500 fix point iterations by using the FFT-based solver with periodic boundary conditions. For comparison, we also solved the problem by a multigrid based solver with Dirichlet boundary conditions. Note, that the results of the two different methods are almost indistinguishable. Also, there is a considerable reduction in the Euclidean norm of the difference.

All computations were carried out in MATLAB [10].

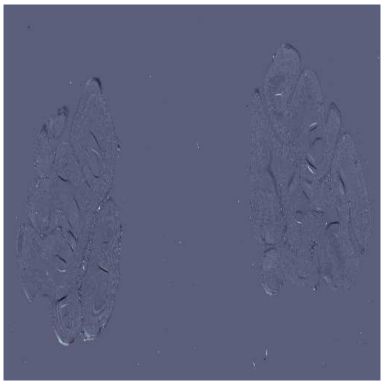
4 Conclusion

We present a direct method for computing the solution of the discrete Navier-Lamé-equation (2). The method is based on Fourier-techniques. This scheme enables one to match two images with say 512×512 pixel (leading to 524288 unknowns) on a standard desktop computer. It is worth noticing that the presented approach carries over straightforward to higher dimensional versions of the Navier-Lamé-equation. The only restrictions, whatsoever, are storage and computing time. We will comment on three-dimensional applications as well as on variations of the overall nonlinear process in a forthcoming paper.

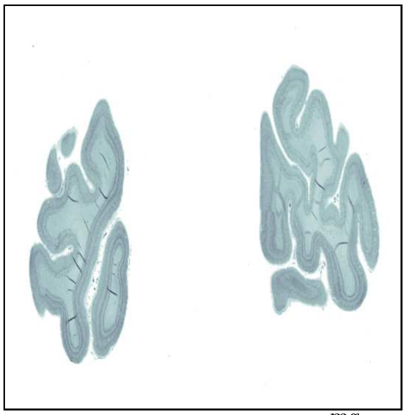
Acknowledgements We are indebted to Dr. Oliver Schmitt (Institute of Anatomy, Medical University of Lübeck) for providing the medical data.



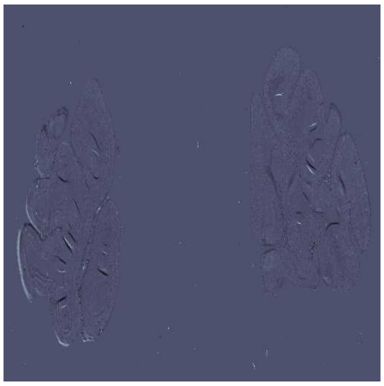
Results after 500 steps, T_{500}^{FFT} using the FFT-based solver for periodic boundary conditions.



Difference $T_{500}^{\mathrm{FFT}} - R$, $\frac{\|T_{500}^{\mathrm{FFT}} - R\|_2}{\|T - R\|_2} \approx 0.32$.



Results after 500 steps, T_{500}^{mg} using a multigrid solver for Dirichlet boundary conditions.



Difference $T_{500}^{\mathrm{mg}} - R$, $\frac{\|T_{500}^{\mathrm{mg}} - R\|_2}{\|T - R\|_2} \approx 0.35$.

Figure 2: Results after 500 outer steps using the FFT-based solver for periodic boundary conditions and a multigrid solver for Dirichlet boundary condition, respectively.

References

[1] Y. Amit, U. Grenander, and M. Piccioni, Structural Image Restoration Through Deformable Templates, Journal of the American Statistical Association, 86(414) (1991) 376–387.

[2] R. Bajcsy and S. Kovačič, Toward an Individualized Brain Atlas Elastic Matching, MS-CIS-86-71 Grasp Lap 76, Dept. of Computer and Information Science, Moore School, University of Philadelphia (1986).

[3] F.L. Bookstein, A statistical Method for Biological Shape Comparisions, J. theor. Biol. 107, (1984) 475–520.

[4] F.L. Bookstein, Size and shape spaces for landmark data in two dimensions, Stat. Sci. 1, (1986) 181–242.

[5] M. Bro-Nielsen, Medical Image Registration and Surgery Simulation, Ph.D. thesis, IMM, Technical University of Denmark (1996).

[6] G.E. Christensen, Deformable Shape Models for Anatomy, Ph.D. thesis, Sever Institute of Technology, Washington University (1994).

[7] P.J. Davis, *Circulant Matrices*, Chelsea Publishing, New York, 1979.

[8] G.H. Golub and C.F. van Loan, *Matrix Computations*, The John Hopkins University Press, Baltimore, Second edition (1989).

[9] M.E. Gurtin, ıAn Introduction to Continuum Mechanics, Academic Press, Orlando (1981).

[10] MATLAB, MATLAB User's Guide, MathWorks, Natick, Mass. (1992).

[11] T. Schormann, S. Henn, and K. Zilles, A New Approach to Fast Elastic Alignment with Applications to Human Brains, LNCS Vol. 1131 (1996) 337–342.

[12] J.-P. Thirion, Non-Rigid Matching Using Demons, *in IEEE, Correspondence on Computer Vision and Pattern Recognition*, 1996, 245–251.