Though it is a perfect tool for this application, it is not a general purpose tool. The regularizer is very restricted, even in the variant described in Section 9.7. For very dissimilar objects, elastic registration is not the method of choice. The distance measure may not be reduced sufficiently; see, e.g., the • to C registration in Section 9.8. Note that the method is designed for linear elastic deformations only and implicitly assumes only small changes of the displacement. Relaxing the Lamé constants by too much may violate this assumption and one might end up with a non-diffeomorphic transformation.

10

FLUID REGISTRATION

10.1 Variational motivation of fluid registration

The essential difference between elastic and fluid registration is, roughly speaking, that for elastic registration the regularization is based on $\mathcal{S}^{\text{elas}}[u] = \mathcal{P}[u]$, whereas in the fluid approach it is based on $\mathcal{S}^{\text{fluid}}[u] = \mathcal{P}[\partial_t u]$. Thus, if u converges to a steady-state solution, the regularizer becomes less important. In this context, the fluid approach may also be viewed as a sequence of Tychonov regularizations for the distance measure,

$$\mathcal{D}[u] + \alpha^k \mathcal{S}[u] \xrightarrow{u} \min, \quad k \geq 0,$$

where α^k goes to zero.

From Theorem 9.4 we obtain

$$f(\,\cdot\,, u(\,\cdot\,,\cdot\,)) = \mu\Delta v + (\lambda + \mu)\nabla \operatorname{div} v, \quad v = \partial_t u,$$

as a characterizing equation for fluid registration.

In contrast to the elastic model the strain in a fluid model depends on the rate of change. Elastic models are characterized by a spatial smoothing of the displacement field. In contrast, fluid models are characterized by a spatial smoothing of the velocity field. Thus, in principle, any displacement can be obtained given enough time: the internal stresses disappear eventually.

10.2 Physical motivation of fluid registration

In this section we present a coherent description of the elastic and fluid continuum models of the deformation of a body, see also Christensen (1994) or Bro-Nielsen (1996). Note that the fluid model is also called *visco-elastic*; cf., e.g., Wollny & Kruggel (2000).

The transformation $\varphi : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$ (cf., Section 9.1.1) is now time dependent and smooth. For any time t, $\varphi(\cdot, t)$ has to be bijective and to preserve the orientation of the tissue. With the displacement $u : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ (cf., eqn (8.1)) the location of a particle P can be described by

$$\tilde{x} = \varphi(x, t) = x + u(x, t),$$

where $x := \varphi(P, 0)$ is the initial position of the particle and $\tilde{x}$ its current position at a fixed time t. With $\varphi(\cdot, t)$ being bijective, we can make use of its inverse

$\tilde{\varphi}(\cdot, t) := \varphi^{-1}(\cdot, t)$ and obtain

$$x = \tilde{\varphi}(\tilde{x}, t) = \tilde{x} - \tilde{u}(\tilde{x}, t),$$

where $u(\cdot, t)$ and $\tilde{u}(\cdot, t)$ are related via $u(x, t) = \tilde{u}(\tilde{x}, t)$.

Using the initial coordinates x of the particles P as the reference coordinate system is also called the Lagrange frame, whereas tracking the particles with respect to their actual position $\tilde{x}$ is called the Euler frame; see also Section 3.3.2 and Remark 9.2. Since the displacement and the velocity of a particle are physical properties and thus independent of the reference frame, we have

$$u(x, t) = \tilde{u}(\tilde{x}, t) \quad \text{and} \quad v(x, t) = \tilde{v}(\tilde{x}, t),$$

where the velocity of the particle P is defined as the partial time derivative of the transformation,

$$v(x, t) := \partial_t \varphi(x, t), \quad x = \varphi(P, 0).$$

Note that

$\partial_t \varphi(P,t) \underset{\text{Remark 9.2}}{=} v(P,t) = \tilde{v}(\tilde{x},t)$

$$\tilde{v}(\tilde{x}, t) = \frac{d}{dt}\tilde{u}(\tilde{x}, t) = \nabla\tilde{u}(\tilde{x}, t)\partial_t\tilde{x} + \partial_t\tilde{u}(\tilde{x}, t) = \nabla\tilde{u}(\tilde{x}, t)\tilde{v}(\tilde{x}, t) + \partial_t\tilde{u}(\tilde{x}, t).$$

As already introduced in Chapter 9, *strain* is measured using the deformation gradient $\nabla\varphi = I_d + \nabla u$; cf., Section 9.1.2.

10.2.1 *The basics of fluid mechanics*

The continuous model satisfies the basic physical axioms such as balance of mass and linear and angular momentum, which are stated below. An important relation between Lagrange and Euler coordinates is given by the Reynolds' transport theorem; cf., e.g., Gurtin (1981, III.10 (5)).

Theorem 10.1 (Reynolds' transport theorem) *Let $\varphi : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ be a smooth spatial field. Then for any set Ω and time t,*

$$\begin{aligned}\frac{d}{dt}\int_{\varphi(\Omega,t)} g(\tilde{x}, t)\, d\tilde{x} &\overset{(*)}{=} \int_{\varphi(\Omega,t)} \frac{d}{dt}g(\tilde{x}, t) + g(\tilde{x}, t)\operatorname{div}\tilde{v}(\tilde{x}, t)\, d\tilde{x} \\ &\overset{(**)}{=} \int_{\varphi(\Omega,t)} \partial_t g(\tilde{x}, t) + \operatorname{div}[g(\tilde{x}, t)\tilde{v}(\tilde{x}, t)]\, d\tilde{x}.\end{aligned}$$

Conservation of mass Let ρ denote the density of the body Ω and let $\tilde{\rho}$ be related to ρ via $\tilde{\rho}(\tilde{x}, t) := \rho(x, t)$. The total mass of the body is constant, i.e.,

$$\frac{d}{dt}\int_{\varphi(K,t)} \tilde{\rho}(\tilde{x}, t)\, d\tilde{x} = 0,$$

for any $K \subset \Omega$. Applying the transport theorem 10.1 and using the fact that the set K can be chosen arbitrarily, we obtain the *continuity equation*

$$\partial_t\tilde{\rho} + \operatorname{div}(\tilde{\rho}\tilde{v}) = 0. \tag{10.1}$$

If in particular the body is an incompressible continuum, the density $\tilde{\rho}$ is constant and the continuity equation (10.1) results in $\operatorname{div}\tilde{v} = 0$.

Conservation of linear momentum The axiom of conservation of linear momentum balances the momentum and the external and internal forces of any subbody K,

$$\frac{d}{dt}\int_{\varphi(K,t)} \tilde{\rho}\tilde{v}\, d\tilde{x} = \int_{\varphi(K,t)} \tilde{f}\, d\tilde{x} - \int_{\partial[\varphi(K,t)]} \tilde{\gamma}(\tilde{x}, \tilde{n})\, d\tilde{x},$$

where γ denotes the *stress* (cf., Section 9.1.3), $\partial[\varphi(K, t)]$ denotes the boundary of the set $\varphi(K, t)$, and $\tilde{n}$ denotes the outer normal. Here, we use the Euler coordinates.

From the transport theorem 10.1 and the continuity equation (10.1) we have

$$\frac{d}{dt}\int_{\varphi(K,t)} \tilde{\rho}\tilde{v}\, d\tilde{x} \overset{(*)}{=} \int_{\varphi(K,t)} \tilde{\rho}\frac{d}{dt}\tilde{v} + \tilde{v}\underbrace{\left(\frac{d}{dt}\tilde{\rho} + \tilde{\rho}\operatorname{div}\tilde{v}\right)}_{\overset{(**)}{=}\ldots\overset{(10.1)}{=} 0} d\tilde{x} = \int_{\varphi(K,t)} \tilde{\rho}\frac{d}{dt}\tilde{v}\, d\tilde{x}.$$

With the stress tensor Σ (cf., Section 9.1.3), we thus have

$$\int_{\varphi(K,t)} \left(\tilde{\rho}\frac{d}{dt}\tilde{v} - \tilde{f} + \operatorname{div}\tilde{\Sigma}\right) d\tilde{x} = 0,$$

or, varying over all subbodies,

$$\tilde{\rho}\frac{d}{dt}\tilde{v} = \tilde{f} - \operatorname{div}\tilde{\Sigma}. \tag{10.2}$$

Remark 10.1 For *static elastic deformations* all time derivatives are zero, since there is no temporal change. Thus, if the strain–stress relation discussed in the previous chapter is used, eqn (10.2) becomes

$$\tilde{f} = \operatorname{div}\tilde{\Sigma} = \mu\Delta\tilde{u} + (\lambda + \mu)\nabla\operatorname{div}\tilde{u}$$

(see Section 9.1.4) and we rediscover the Navier–Lamé equations.

10.2.2 *Stokes fluids*

The Stokes hypothesis is that the stress tensor can be decomposed into parts related to velocity $\tilde{v}$ and the pressure $\tilde{p}$,

$$\tilde{\Sigma} = [\mu(\nabla\tilde{v} + \nabla\tilde{v}^\top) + \lambda \operatorname{div} \tilde{v} I_d] + \tilde{p} I_d. \tag{10.3}$$

Assuming $\tilde{\Sigma}$ to be as in eqn (10.3) it follows that

$$\operatorname{div} \tilde{\Sigma} = \mu\Delta\tilde{v} + (\lambda+\mu)\nabla \operatorname{div} \tilde{v} + \nabla\tilde{p}.$$

Moreover, if we assume very slow motion flows (small Reynolds' number), we may neglect the time derivative of the velocity and the spatial derivative of the pressure. Hence, from eqn (10.2) we get

$$\tilde{f} = \operatorname{div} \tilde{\Sigma} = \mu\Delta\tilde{v} + (\lambda+\mu)\nabla \operatorname{div} \tilde{v},$$

which is the underlying partial differential equation for *fluid registration.* Note that severe simplifications of the general fluid model have been applied. In particular, one assumes that the momentum vanishes and the fluid fulfills the Stokes hypothesis with spatially constant pressure.

10.3 Solving the Navier–Lamé equations for the velocity

Now we present a numerical scheme for the solution of the partial differential equation

$$\mu\Delta v + (\lambda+\mu)\nabla \operatorname{div} v = f(\,\cdot\,, u(\,\cdot\,,\,\cdot\,)), \quad v = \partial_t u + \nabla u\, v; \tag{10.4}$$

see Section 10.2. Since we are only interested in the Euler coordinates, from now on we omit the additional $\tilde{\ }$-notation of the variables.

The basic idea here is to use a fixed-point iteration, i.e., to compute the force field f for a given displacement u, to solve the linear partial differential equation for a new velocity v, and to compute the new displacement from this velocity using an Euler scheme. For the numerical computations we use a finite difference approximation for the derivatives as introduced in Chapter 8. For each step k, the computation of the force field $\vec{F}^{(k)}$ and the solution of the linear system of equations $A\vec{V}^{(k+1)} = \vec{F}^{(k)}$ are the same as in our implementation of elastic registration. The only difference with respect to the implementation of elastic registration is the computation of $\vec{U}^{(k+1)}$ from $\vec{V}^{(k+1)}$, which requires an additional Euler step.

For each grid point $x_i \in \mathbb{R}^d$ with $i = (i_1,\dots,i_d) \in \mathbb{N}^d$ we also have $u^{(k)}(x_i) \in \mathbb{R}^d$. Thus, we need an $n_1 \times \cdots \times n_d \times d$ array for the storage of all components of $u^{(k)}(X)$. This array is denoted by

$$\vec{U}^{(k)} = [U^{(k,1)},\dots,U^{(k,d)}] \in \mathbb{R}^{n_1\times\cdots\times n_d\times d}.$$

For convenience we introduce this data structure also for the identity on the grid and denote the corresponding array by $\vec{X} = [X^1,\dots,X^d]$. Hence, the coordinates of the i^{th} grid point are $x_i = (X_i^1,\dots,X_i^d)^\top$.

Example 10.1 Suppose $d=2$, $n_1 = n_2 = 3$, and our grid matrix is given, for example, by

$$X = (x_{i_1,i_2})_{i_1,i_2=1,2,3} \in \mathbb{R}^{n_1\times n_2} \quad \text{with} \quad x_{i_1,i_2} = (i_1,i_2)^\top \in \mathbb{R}^d.$$

With

$$X^1 = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix}, \quad X^2 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \in \mathbb{R}^{3\times 3}$$

we have $\vec{X} = [X^1, X^2]$ and, for example, $x_{2,3} = [X^1_{2,3}, X^2_{2,3}]^\top = (2,3)^\top$.

For a fixed index $i = (i_1,\dots,i_d)$, we set

$$\vec{U}_i^{(k)} := \left(U_i^{(k,1)},\dots,U_i^{(k,d)}\right)^\top \in \mathbb{R}^d$$

and

$$\vec{V}_i^{(k)} := \left(V_i^{(k,1)},\dots,V_i^{(k,d)}\right)^\top \in \mathbb{R}^d,$$

respectively. Using a centered finite difference approximation for $\nabla u(x_i, k\tau)$,

$$\hat{J}_i^{(k)} := \left(\hat{J}^{(k)}_{i;\ell,q}\right)_{\ell,q=1,\dots,d}, \quad \text{where } \hat{J}^{(k)}_{i;\ell,q} := \tfrac{1}{2}\left(U^{(k,\ell)}_{i+e_q} - U^{(k,\ell)}_{i-e_q}\right), \tag{10.5}$$

and a forward finite difference approximation for the partial time derivative $\partial_t u(x_i, k\tau)$, we obtain a discretized version of the Euler step,

$$\frac{\vec{U}_i^{(k+1)} - \vec{U}_i^{(k)}}{\tau} = \left(I_d - \hat{J}_i^{(k)}\right)\vec{V}_i^{(k)} \quad \text{for all } i.$$

Our algorithm is summarized in Algorithm 10.2.

Example 10.2 Suppose $d=2$, $n_1 = n_2 = 3$, $\vec{U} = [U^{(k,1)}, U^{(k,2)}]$. For a fixed k and all $i = (i_1,i_2)$ we have to compute the d^2 values

$$\hat{J}^{(k)}_{i;1,1} = \tfrac{1}{2}\left(U^{(k,1)}_{i_1+1,i_2} - U^{(k,1)}_{i_1-1,i_2}\right), \quad \hat{J}^{(k)}_{i;1,2} = \tfrac{1}{2}\left(U^{(k,1)}_{i_1,i_2+1} - U^{(k,1)}_{i_1,i_2-1}\right),$$
$$\hat{J}^{(k)}_{i;2,1} = \tfrac{1}{2}\left(U^{(k,2)}_{i_1+1,i_2} - U^{(k,2)}_{i_1-1,i_2}\right), \quad \hat{J}^{(k)}_{i;2,2} = \tfrac{1}{2}\left(U^{(k,2)}_{i_1,i_2+1} - U^{(k,2)}_{i_1,i_2-1}\right).$$

The matrix

$$\hat{J}_i^{(k)} = \begin{pmatrix} \hat{J}^{(k)}_{i;1,1} & \hat{J}^{(k)}_{i;1,2} \\ \hat{J}^{(k)}_{i;2,1} & \hat{J}^{(k)}_{i;2,2} \end{pmatrix} \in \mathbb{R}^{d\times d}$$

is the finite difference approximation to $\nabla u(x_i, k\tau)$.

Algorithm 10.2 *MATLAB pseudo-code for fluid registration.*

```
[U⃗, T_out] = FluidRegistration(R, T);
  k = 0;  q = regrid_counter = 0;  [n_1, ..., n_d] = size(R);
  X⃗ = [X_1, ..., X_d] = ndgrid(1 : n_1, ..., 1 : n_d);
  U⃗^(k) = zeros(size(X⃗));  Y⃗^(q) = zeros(size(X⃗));
  setup_NLE_matrix;

  while 1
    W⃗^(k) = interpolate(Y⃗^(regrid_counter), X⃗ − U⃗^(k));
    T^(k) = interpolate(T, X⃗ − W⃗^(k) − U⃗^(k));
    if k ≥ maxiter, STOP; end;
    if ||R − T^(k−1)||_F − ||R − T^(k)||_F ≤ ||R − T^(k)||_F tol_dist,
  STOP; end;

    F⃗^(k) = compute_forces(R, T^(k));
    if ||F⃗^(k)||_V ≤ tol_for, STOP; end;
    V⃗^(k) = solve_NLE_matrix(F⃗^(k));

    for j, ℓ = 1, ..., d,  J_{j,ℓ}^(k) = compute_Jacobian(U⃗_j^(k), ℓ);
  end,

    min_Jacobian = min(det([J_{j,ℓ}^(k)]));
    if (min_Jacobian < tol_Jac),
      regrid_counter = regrid_counter + 1;
      Y⃗^(regrid_counter) = W⃗^(k) + U⃗^(k);
      U⃗^(k+1) = zeros(size(X⃗));
    else
      δU⃗^(k) = [J_{j,ℓ}^(k)] V⃗^(k);
      δu_max^(k) = ||δU⃗^(k)||_V;
      δt^(k) = min{1, tol_u / δu_max^(k)};
      U⃗^(k+1) = U⃗^(k) + δt^(k) δU⃗^(k);
    end,
    k = k + 1;
  end;
  U⃗ = W⃗^(k) + U⃗^(k);
  T_out = interpolate(T, X⃗ − U⃗).
```


Remark 10.2 It is worthwhile noticing that the Jacobian matrix

$$J(x) := \nabla[x - u(x)]$$

arising in the Euler step

$$\partial_t u = (\mathcal{I} - \nabla u)v \tag{10.6}$$

is nothing but the derivative of the transformation with respect to the Euler coordinates. Since the entries of this matrix are available during the computation, it is to easy to verify whether this matrix is regular or not. If the matrix is not regular, the transformation is not bijective and the step is not admissible. To circumvent this undesirable situation, Christensen (1994) suggests a so-called *regridding*, which means to clear the memory of the deformation.

Christensen suggests a regridding if $\det J(x) < \text{tol}$ for a grid point x. A mathematically preferable criterion is based on the smallest singular value of $J(x)$. Although of the same complexity as the computation of the determinants, the computation of the singular values is slightly more involved.

10.4 Implementation details

A few remarks concerning the implementation of our fluid registration scheme are in order. To this end we provide a MATLAB pseudo-code of our implementation, see Algorithm 10.2.

In an initialization phase, we set the iteration counter $k = 0$ and a counter for the regridding steps to zero. The d-dimensional images to be registered are $R, T \in \mathbb{R}^{n_1 \times \cdots \times n_d}$.

The initial displacement as well as the variable for the regridding step is set to zero, $\vec{U}^{(0)} = \vec{Y}^{(0)} = 0$. Finally, we initialize the matrices $D^{j,k}$, $j, k = 1, \ldots, d$; cf., Lemma 9.6.

In the following `while`-loop we find four essential blocks: the computation of the deformed template $T^{(k)}$, the computation of the forces $\vec{F}^{(k)}$, the computation of the velocities $\vec{V}$, and the Euler step. The most striking part is probably the regridding step.

10.4.1 *Euler step and regridding*

The displacement and the velocity are connected via the material derivative, see Section 10.3 and in particular eqn (10.6).

An approximation to the Jacobian of the transformation at the grid point x_i is given by

$$J_i^{(k)} = I_d - \hat{J}_i^{(k)}$$

where $\hat{J}_i^{(k)}$ is an approximation to the Jacobian $\nabla u(x_i, k\tau)$; cf., eqn (10.5).

The transformation is regarded as admissible if the determinants of the Jacobian on the grid exceed a certain user-supplied threshold tol_{Jac}. In our implementation we used $\text{tol}_{\text{Jac}} = 0.025$.

If the transformation is admissible, the Euler step $\vec{U}_i^{(k+1)} = \vec{U}_i^{(k)} + \delta t^{(k)} \delta\vec{U}_i^{(k)}$ is performed for all i, with

$$\delta\vec{U}_i^{(k)} = (I_d - \hat{J}_i^{(k)})\vec{V}_i^{(k)}.$$

The time-step $\delta t^{(k)}$ is chosen such that $||\delta t^{(k)}\delta\vec{U}^{(k)}||_V \le \mathrm{tol_u}$, where $\mathrm{tol_u}$ is a user-supplied tolerance, e.g., $\mathrm{tol_u} = 1$. Here,

$$q(i) = [q_1(i), \dots, q_d(i)] \in \mathbb{R}^d,$$
$$||\vec{Q}||_V = \max\{||q(i)||_{\mathbb{R}^d}, \quad i_\ell = 1, \dots, n_\ell, \quad \ell = 1, \dots, d\}. \tag{10.7}$$

If, in contrast, the transformation is not admissible, we perform a regridding and the regridding counter is increased. The displacements of the previous regridding steps with respect to the actual deformation $\vec{W}^{(k)}$ are updated by $\vec{U}^{(k)}$ and denoted by $\vec{Y}^{(\mathrm{regrid\_counter})}$ and then they are set to zero. Note that setting the displacements to zero removes the inner strain of the body with respect to the grid $\vec{X} - \vec{Y}^{(\mathrm{regrid\_counter})}$.

10.4.2 *Interpolation and regridding*

In the interpolation step, the displacement with respect to the last regridding step $\vec{w}^{(k)}$ on the deformed grid $\vec{x} - \vec{u}^{(k)}$ is computed using an interpolation scheme, e.g., d-linear interpolation; cf., Section 3.1.3. Thus, $\vec{x} - \vec{w}^{(k)}$ are the Lagrange coordinates of the strain-free body. The template is interpolated on the deformed grid $\vec{x} - \vec{w}^{(k)} - \vec{u}^{(k)}$.

Figure 10.1 illustrates an original template T_0, the template T_1 on the grid with respect to the latest regridding step, and the template T_2 on the actual grid. For a fixed particle, let x_j, $j = 0, 1, 2$, denote the Euler coordinates of the particle with respect to the three images. Thus, $T_0(x_0) = T_1(x_1) = T_2(x_2)$. The displacement of the particle is phrased in Euler coordinates by $y(x_1) := x_1 - x_0$.

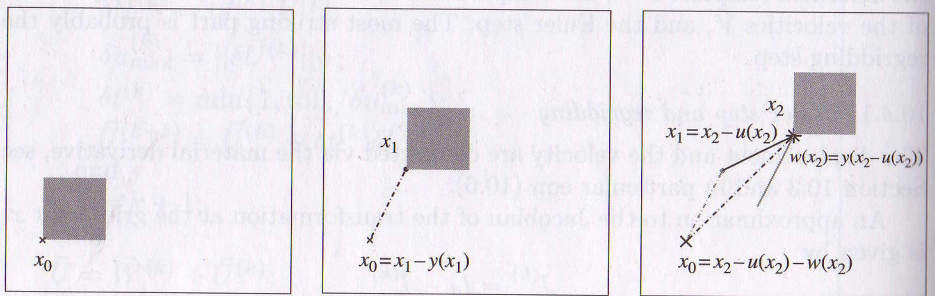

FIG. 10.1 Fluid registration and regridding. LEFT: template T_0 and position of a particle; MIDDLE: template on deformed grid T_1, initial and final position of the particle; RIGHT: template on actual grid T_2, initial, intermediate, and actual position of the particle.

The strain in the body is related to the displacement of x_1 and x_2, which is also phrased in Euler coordinates, $u(x_2) = x_2 - x_1$. The intensity at x_2 is given by

$$\begin{aligned} T_2(x_2) &= T_1(x_2 - u(x_2)) \\ &= T_0(x_1 - y(x_1)) = T_0(x_2 - u(x_2) - y(x_1)) \\ &= T_0(x_2 - u(x_2) - w(x_2)), \end{aligned}$$

where $w(x_2) = y(x_1) = y(x_2 - u(x_2))$. By introducing w, the intermediate T_1 becomes redundant.

10.4.3 *Computing forces and velocities*

For the computation of the forces, we proceed as explained in Section 8.3. Note that in the computation of the finite difference approximations of the gradients the particular boundary conditions enter into play. For the computation of the velocities, we exploit the scheme developed in Section 9.5.

10.4.4 *Stopping criteria*

Different conditions are used as a stopping criterion.

1. As a safeguard, we provide a maximum number of iteration steps.
2. If the relative change of the distance measure is brought below a user-supplied tolerance, e.g., 10^{-5}, the iteration is stopped,

$$\frac{||R - T^{(k-1)}||_F - ||R - T^{(k)}||_F}{||R - T^{(k)}||_F} \le \mathrm{tol_{dist}}.$$

   Note that in our implementation a possible division by zero is avoided.
3. The iteration might be stopped if the forces are too small, $||\vec{f}^{(k)}||_V \le \mathrm{tol_{for}}$. We do not use this criterion in our implementation.
4. Finally, the iteration might be stopped if the change of the displacement is too small. Though the implementation of this criterion is straightforward, a description is lengthy if regridding is taken into account. For this reason, this criterion is not contained in the implementation illustrated in Algorithm 10.2.

10.5 Fluid registration: ● to C

We continue the registration of a ● to a C, see Section 9.8. In Fig. 10.2, we show intermediate results of the fluid registration for $k = 0(40)200$. In order to visualize the deformation, the template has been shaded. Note that the computations have been performed on the original images. In this example, the Lamé constants are $\lambda = 0$ and $\mu = 5000$. The regridding tolerance is 0.025.

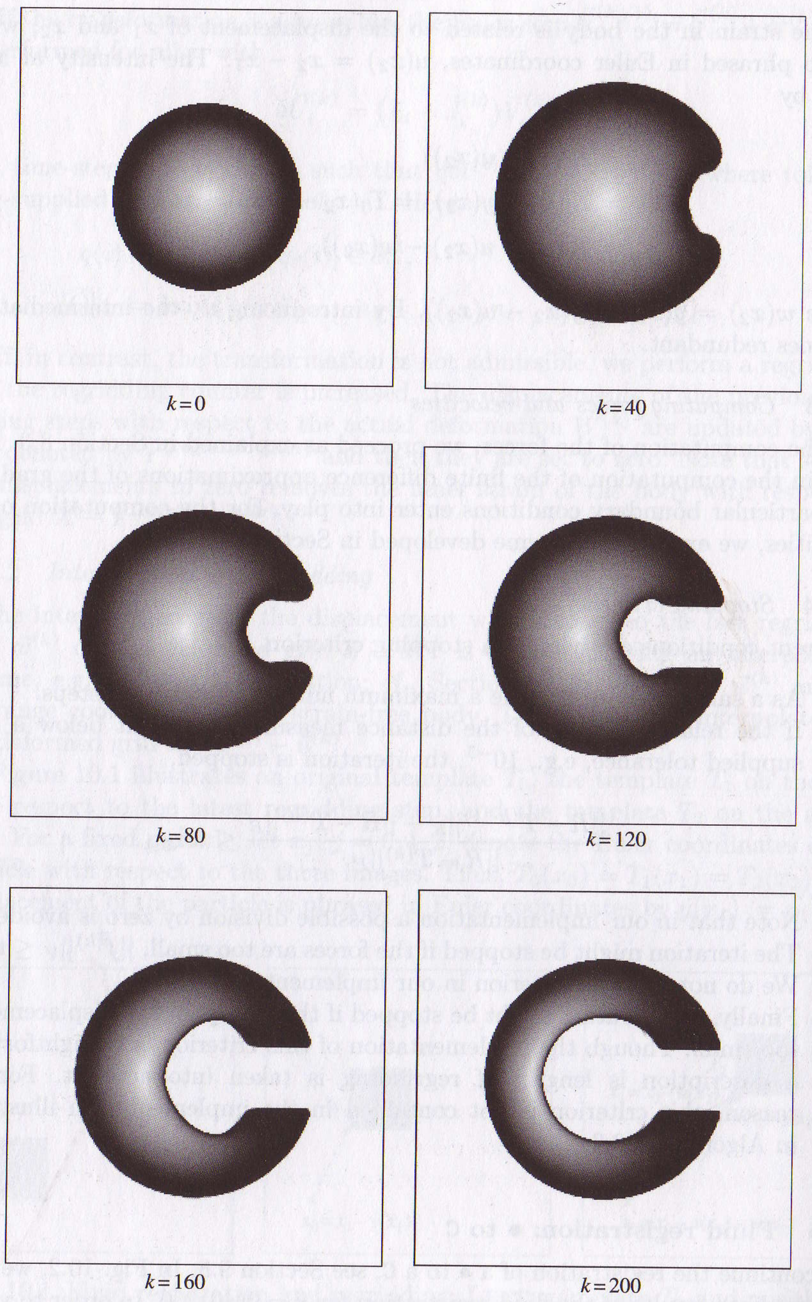

FIG. 10.2 Fluid registration of • to C, $\lambda = 0$, $\mu = 5000$; intermediate results $k = 0(40)200$.

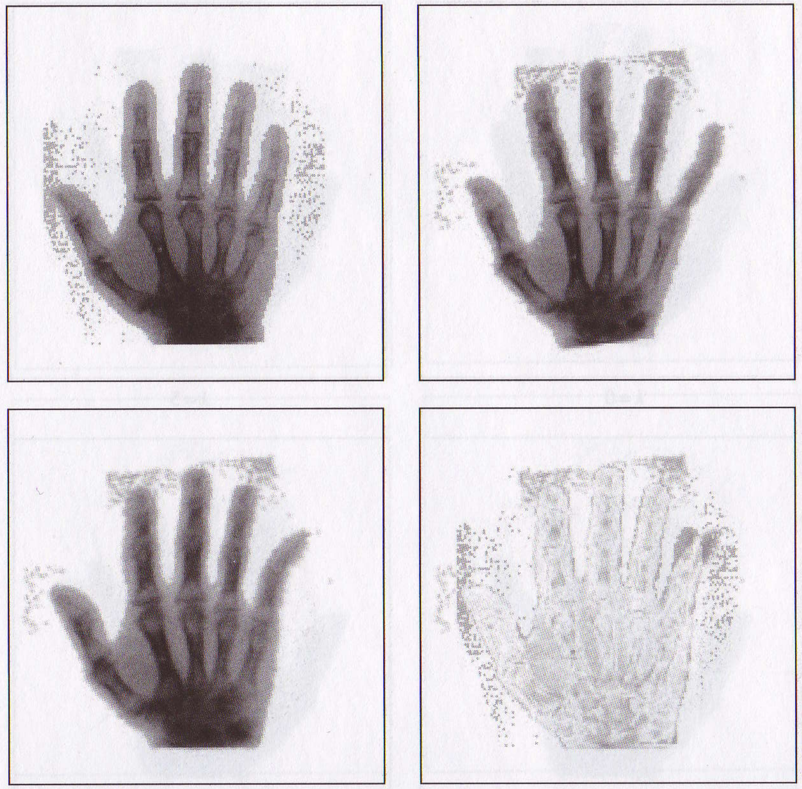

FIG. 10.3 Fluid registration of hands ($\mu = 500$ and $\lambda = 0$), affine linear preregistered; TOP LEFT: reference image R, TOP RIGHT: template image T, BOTTOM LEFT: fluid registered template T^{fluid} with deformation, and BOTTOM RIGHT: difference $|R - T^{\text{fluid}}|$.

10.6 Fluid registration: hands

Figure 10.3 shows the two modified X-ray images of human hands and the fluid registered hand with an illustration of the deformation, see also Section 9.9. As is apparent from these figures, the fluid registration gives optically pleasing results; see also Fig. 10.4 for intermediate results.

Note that fluid as well as elastic registrations are not able to perform a rigid registration. In the modeling of the strain tensor, we assume that the deformation has no rigid components. We illustrate the importance of this assumption with the registration of the images shown in Fig. 10.5, see also Fig. 10.6. Here, the initial template image has been rotated by 20 degrees.

Though we can see a rotation of the grid by about 20 degrees as well, parts of the image like, for example, the finger tips are deformed in a physically unpleasing fashion.

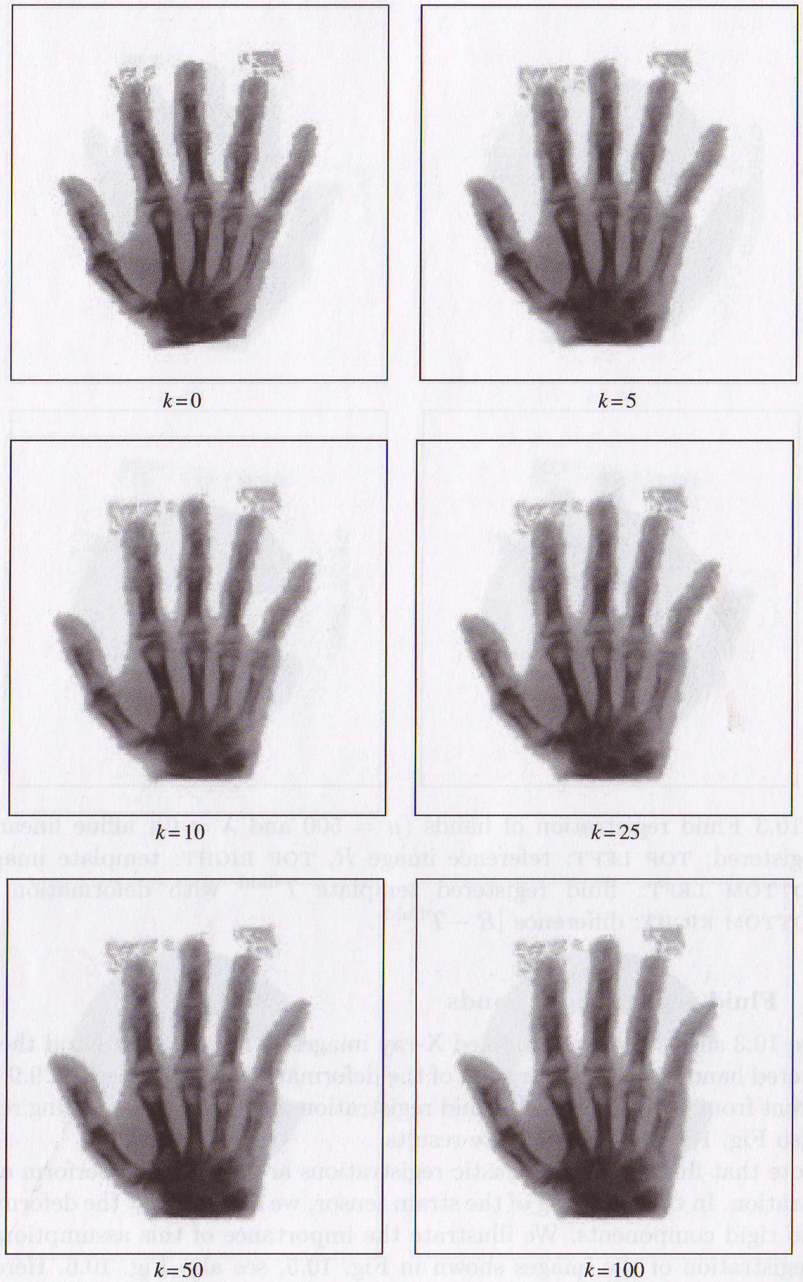

FIG. 10.4 Fluid registration of hands ($\mu = 500$ and $\lambda = 0$), intermediate results for $k = 0, 5, 10, 25, 50, 100$.

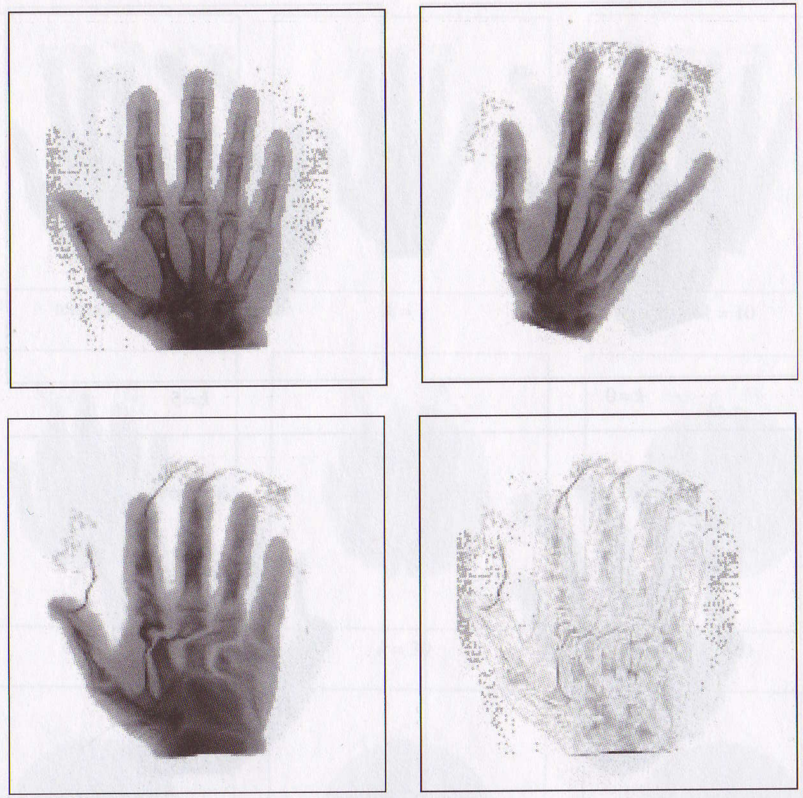

FIG. 10.5 Fluid registration of hands ($\mu = 100$ and $\lambda = 0$), no pre-registration; TOP LEFT: reference image R, TOP RIGHT: template image T, BOTTOM LEFT: fluid registered template T^{fluid} with deformation, and BOTTOM RIGHT: difference $|R - T^{\text{fluid}}|$.

10.7 Fluid registration: hand to disk

In this example, we register the template, a modified version of a hand image from Amit (1994), to the reference image depicted in Fig. 10.7. The images are manipulated such that they share the same gray value range. Besides interpolation artifacts, a complete registration of the template to the reference should be possible. As this example shows, this registration is in fact almost possible. Here, the registration process stopped, because the forces became too small due to interpolation and discretization artifacts.

10.8 Discussion of fluid registration

The fluid registration technique provides a powerful tool for image registration. In principle, it is possible to deform any template image to any reference image,

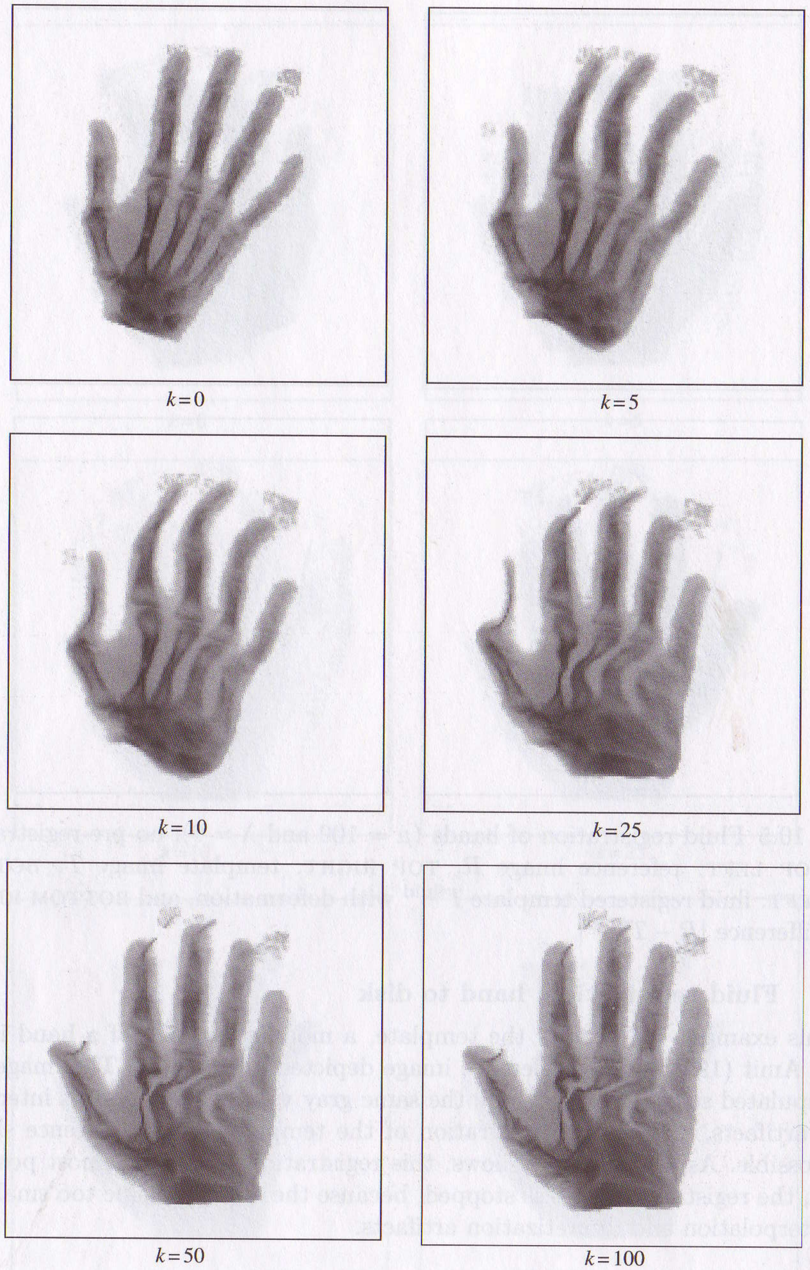

FIG. 10.6 Fluid registration of hands without pre-registration ($\mu = 100$ and $\lambda = 0$); intermediate results for $k = 0, 5, 10, 25, 50, 100$.

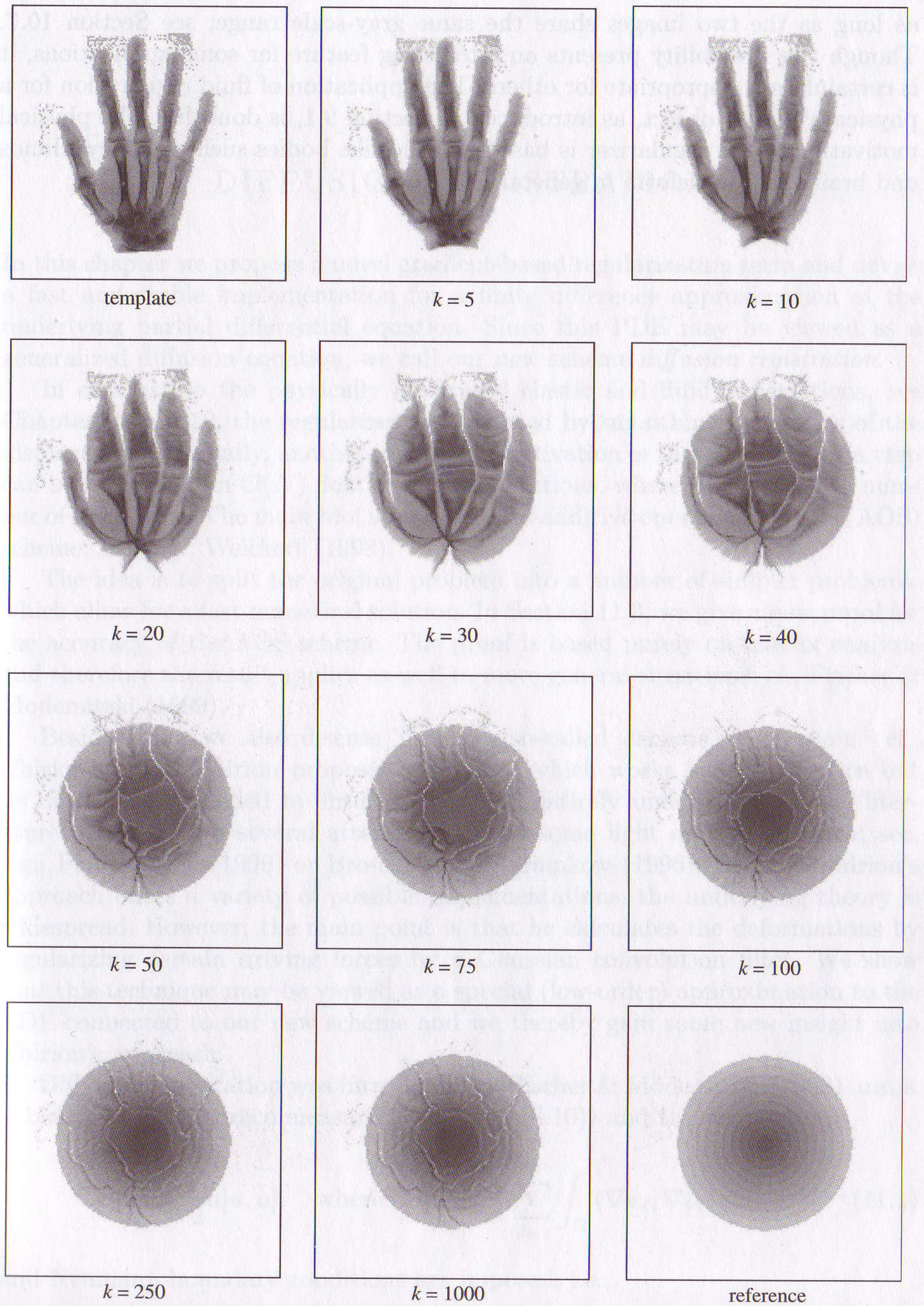

FIG. 10.7 Fluid registration of a hand to a disk, template, intermediate, and reference images.

as long as the two images share the same gray-scale range; see Section 10.7. Though this possibility presents an interesting feature for some applications, it is certainly not appropriate for others. The application of fluid registration for a physically elastic object, as introduced in Section 9.1, is doubtful. The physical motivation of the regularizer is based on fluid-like bodies such as honey. Hands and brains do not deform in general like honey.

11

DIFFUSION REGISTRATION

In this chapter we propose a novel gradient-based regularization term and devise a fast and stable implementation for a finite difference approximation of the underlying partial differential equation. Since this PDE may be viewed as a generalized diffusion equation, we call our new scheme *diffusion registration.*

In contrast to the physically motivated elastic and fluid registrations, see Chapters 9 and 10, the regularizer is motivated by smoothing properties of the displacement. Actually, another important motivation is that a registration step can be performed in $\mathcal{O}(N)$ floating point operations, where N denotes the number of unknowns. The main tool is the so-called additive operator splitting (AOS) scheme; see, e.g., Weickert (1998).

The idea is to split the original problem into a number of simpler problems, which allow for a fast numerical solution. In Section 11.3, we give a new proof for the accuracy of the AOS scheme. The proof is based purely on matrix analysis and therefore the result applies as well to more general situations; cf., Fischer & Modersitzki (1999).

Besides this, we also discuss Thirion's so-called *demons registration*; cf., Thirion (1998). Thirion proposed a method which works well in practice but its derivation is guided by intuition and not entirely understood. In the literature one may find several attempts to shed some light on his approach; see, e.g., Pennec et al (1999) or Bro-Nielsen & Gramkow (1996). Because Thirion's approach offers a variety of possible implementations, the underlying theory is widespread. However, the main point is that he calculates the deformations by regularizing certain driving forces by a Gaussian convolution filter. We show that this technique may be viewed as a special (low-order) approximation to the PDE connected to our new scheme and we thereby gain some new insight into Thirion's approach.

Diffusion registration was introduced by Fischer & Modersitzki (1999) and it is based on the distance measure $\mathcal{D}$ (cf., eqn (8.10)) and the regularizer

$$S[u] := \frac{1}{2}a[u,u], \quad \text{where} \quad a[u,v] = \sum_{\ell=1}^{d} \int_{\Omega} \langle \nabla u_\ell, \nabla v_\ell \rangle \, dx \qquad (11.1)$$

and Neumann boundary conditions are imposed, i.e.,

$$\langle \nabla u_\ell(x), \vec{n}(x) \rangle_{\mathbb{R}^d} = \langle \nabla v_\ell(x), \vec{n}(x) \rangle_{\mathbb{R}^d} = 0 \quad \text{for } x \in \partial\Omega$$

and $\ell = 1, \ldots, d$. Here, $\vec{n}$ denotes the outer normal unit vector of $\partial\Omega$, $\Omega :=]0,1[^d$.