A fast full multigrid solver for applications in image processing

M. Stürmer, H. Köstler, U. Rüde
Lehrstuhl für Systemsimulation
Friedrich-Alexander Universität Erlangen-Nürnberg,Germany

March 19th 2007

Motivation

Challanges of Medical Image Processing

- large volume data sets (512^3 voxels and more)

or / and

- real time performance

Variational Approaches

typically elliptic PDE

- e. g. image denoising
- e. g. non-rigid image registration

- optimized MG solver for time-dependent Poisson equation
- compare with optimized FFT-based implementations

Contents

Linear Heat Equation

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) - \Delta u(\mathbf{x}, t) = f(\mathbf{x}), \quad u(\mathbf{x}, 0) = u_0(\mathbf{x}) \tag{1}$$

with time $t \in \mathbb{R}^+$, $u, f : \Omega \subset \mathbb{R}^3 \longrightarrow \mathbb{R}, \mathbf{x} \in \Omega$,
initial solution $u_0 \; : \; \Omega \subset \mathbb{R}^3 \longrightarrow \mathbb{R}$ and
homogeneous Neumann boundary conditions.

(1) is discretized using finite differences

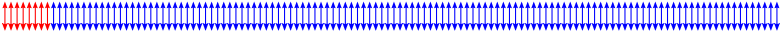
$$\frac{u_h(\mathbf{x}, \tau) - u_0(\mathbf{x})}{\tau} - \Delta_h u_h(\mathbf{x}, \tau) = f_h(\mathbf{x}), \tag{2}$$

on a regular grid Ω_h with mesh size h and time step τ.

Cell-based multigrid solver using averaging restriction, ωRBGS smoother and constant interpolation.

Single Instruction Multiple Data

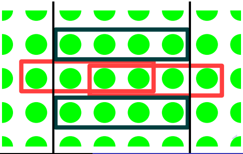
- Buses become wider and wider but ineffective for scalar operations.



- SIMD units exploit the wider buses if
  - the same operation is done on neighboring values

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \begin{matrix} * \\ * \\ * \\ * \end{matrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

  - and data is naturally aligned in memory.
- Example: SIMDized 2D Poisson Jacobi



SIMDized ωRBGS

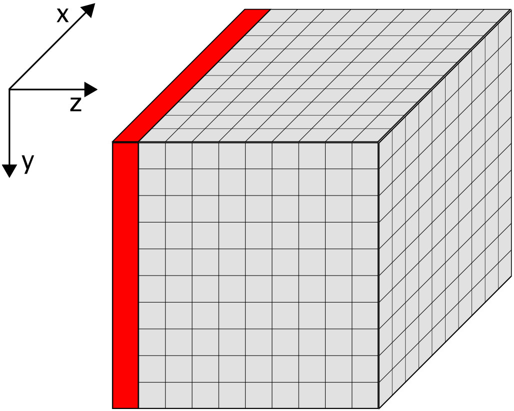
Using two – per line – constant SIMD registers, we can treat vector members differently:

$$\begin{bmatrix} u_a^{\text{new}} \\ u_b^{\text{new}} \\ u_c^{\text{new}} \\ u_d^{\text{new}} \end{bmatrix} \begin{matrix} = \\ = \\ = \\ = \end{matrix} \begin{bmatrix} 1-\omega \\ 1 \\ 1-\omega \\ 1 \end{bmatrix} \begin{matrix} * \\ * \\ * \\ * \end{matrix} \begin{bmatrix} u_a^{\text{old}} \\ u_b^{\text{old}} \\ u_c^{\text{old}} \\ u_d^{\text{old}} \end{bmatrix} \begin{matrix} + \\ + \\ + \\ + \end{matrix} \begin{bmatrix} \omega \\ 0 \\ \omega \\ 0 \end{bmatrix} \begin{matrix} * \\ * \\ * \\ * \end{matrix} \begin{bmatrix} u_a^{\text{jac}} \\ u_b^{\text{jac}} \\ u_c^{\text{jac}} \\ u_d^{\text{jac}} \end{bmatrix} .$$

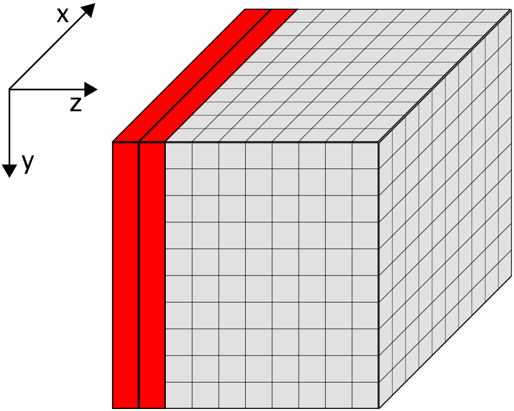
A red or black ωRBGS sweep is computationally about as expensive as a damped Jacobi.

Although we effectively invalidate half FLOPs done, the better internal and external bandwidth of SIMD over the scalar unit leads to a real performance gain.

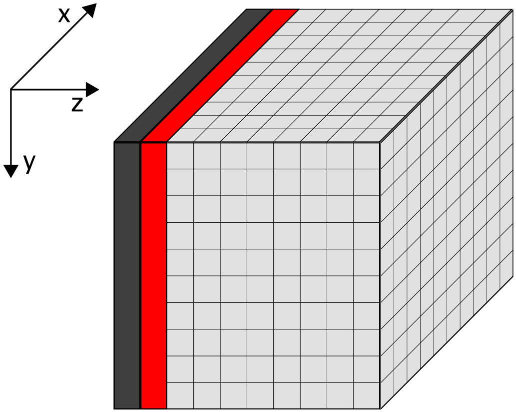
Simple Plane Blocking



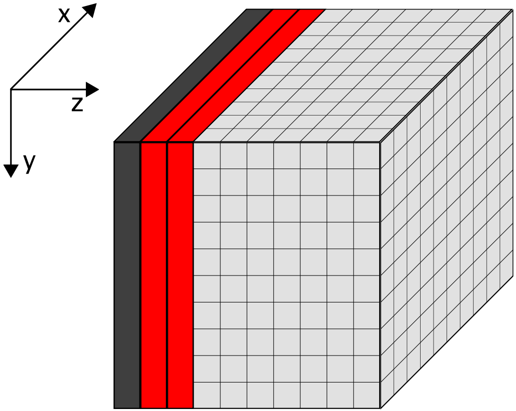
Simple Plane Blocking



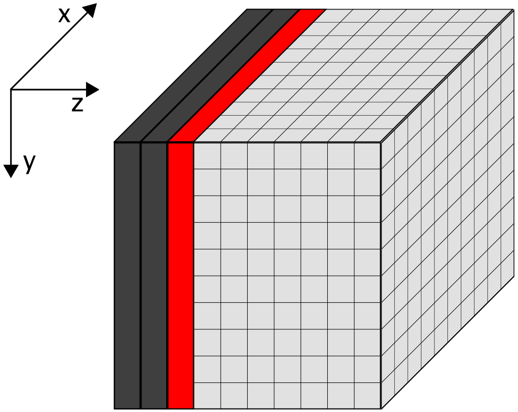
Simple Plane Blocking



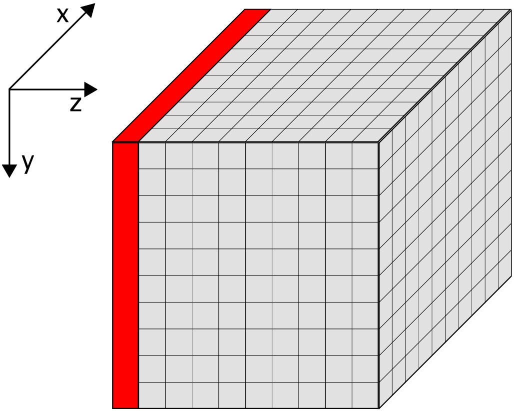
Simple Plane Blocking



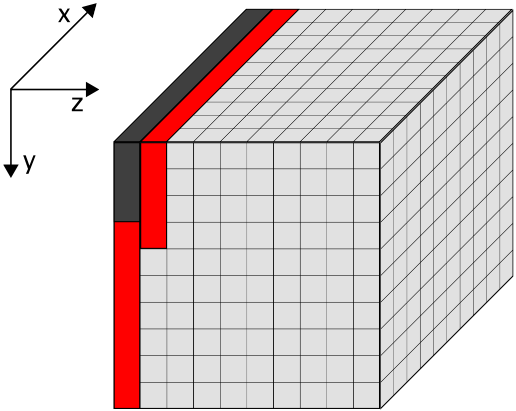
Simple Plane Blocking



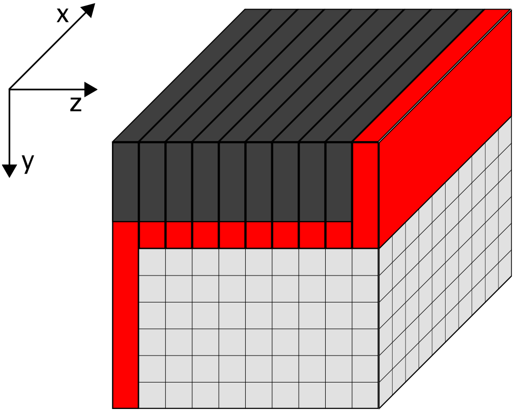
Sub-Blocking



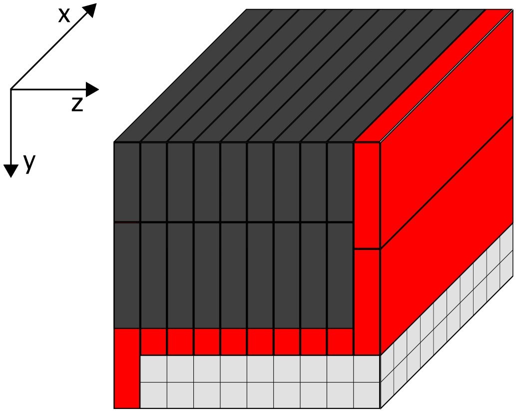
Sub-Blocking



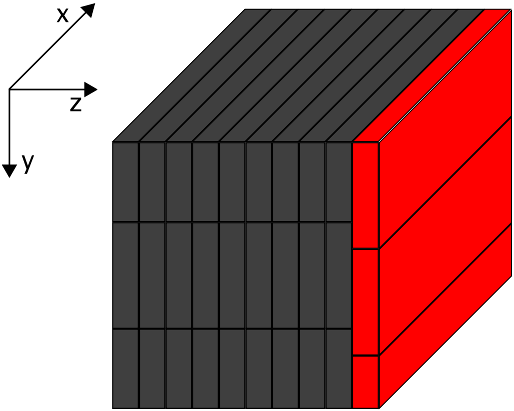
Sub-Blocking



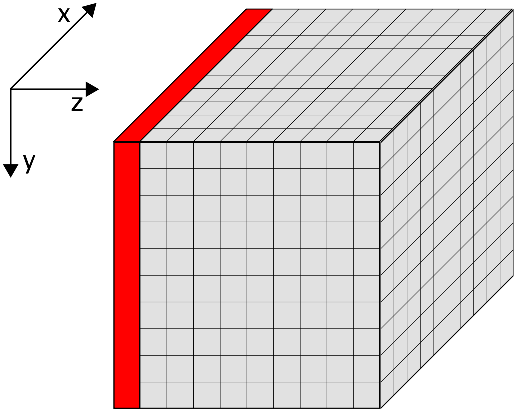
Sub-Blocking



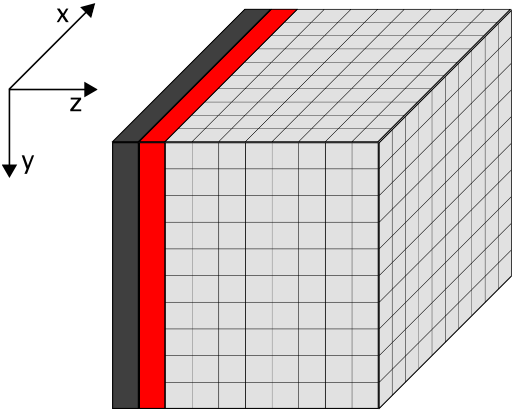
Sub-Blocking



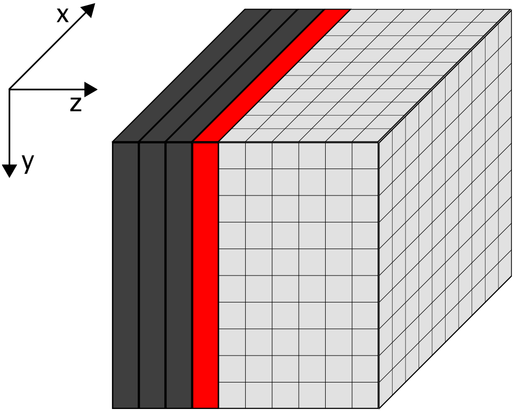
Fusion with Calculation of Residual and Restriction



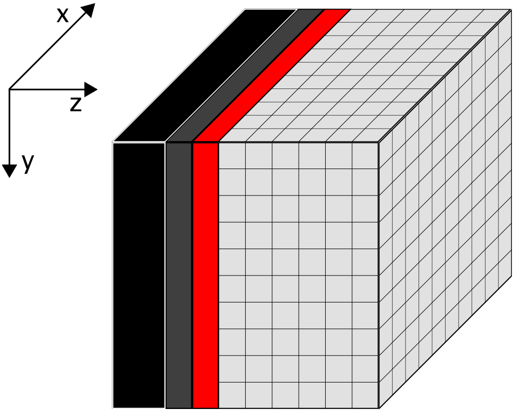
Fusion with Calculation of Residual and Restriction



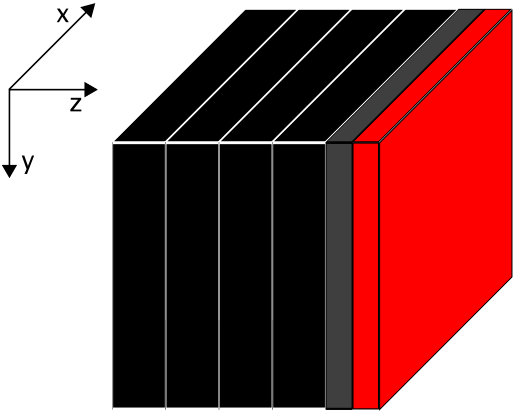
Fusion with Calculation of Residual and Restriction



Fusion with Calculation of Residual and Restriction



Fusion with Calculation of Residual and Restriction



MG vs. FFT

size	V (1,1)	FMG V(1,1)	FMG V(2,2)	FFT (fftw)	DCT (fftw)	FFT (mkl)
32	0.43	0.55	0.93	0.40	1.43	0.71

Table: Wallclock times in ms for FFT (real type, out of place, forward and backward) and the optimized multigrid on an Intel Core2 Duo 2.4 GHz (Conroe) workstation.

MG vs. FFT

size	V (1,1)	FMG V(1,1)	FMG V(2,2)	FFT (fftw)	DCT (fftw)	FFT (mkl)
32	0.43	0.55	0.93	0.40	1.43	0.71
64	3.33	4.29	7.12	3.73	12.2	5.27

Table: Wallclock times in ms for FFT (real type, out of place, forward and backward) and the optimized multigrid on an Intel Core2 Duo 2.4 GHz (Conroe) workstation.

MG vs. FFT

size	V (1,1)	FMG V(1,1)	FMG V(2,2)	FFT (fftw)	DCT (fftw)	FFT (mkl)
32	0.43	0.55	0.93	0.40	1.43	0.71
64	3.33	4.29	7.12	3.73	12.2	5.27
128	31.6	44.1	68.3	50.4	123	45.8

Table: Wallclock times in ms for FFT (real type, out of place, forward and backward) and the optimized multigrid on an Intel Core2 Duo 2.4 GHz (Conroe) workstation.

MG vs. FFT

size	V (1,1)	FMG V(1,1)	FMG V(2,2)	FFT (fftw)	DCT (fftw)	FFT (mkl)
32	0.43	0.55	0.93	0.40	1.43	0.71
64	3.33	4.29	7.12	3.73	12.2	5.27
128	31.6	44.1	68.3	50.4	123	45.8
256	264	370	574	473	1246	401

Table: Wallclock times in ms for FFT (real type, out of place, forward and backward) and the optimized multigrid on an Intel Core2 Duo 2.4 GHz (Conroe) workstation.

MG vs. FFT

size	V (1,1)	FMG V(1,1)	FMG V(2,2)	FFT (fftw)	DCT (fftw)	FFT (mkl)
32	0.43	0.55	0.93	0.40	1.43	0.71
64	3.33	4.29	7.12	3.73	12.2	5.27
128	31.6	44.1	68.3	50.4	123	45.8
256	264	370	574	473	1246	401
512	2168	3026	4699	4174	11067	3510

Table: Wallclock times in ms for FFT (real type, out of place, forward and backward) and the optimized multigrid on an Intel Core2 Duo 2.4 GHz (Conroe) workstation.

Convergence Rates

size	τ	V(1,1)	V(2,2)
64^3	10^4	0.26	0.07
128^3	10^4	0.28	0.07
256^3	10^4	0.29	0.07
512^3	10^4	0.32	0.07
64^3	10^{30}	0.27	0.07
128^3	10^{30}	0.29	0.07
256^3	10^{30}	0.32	0.07
512^3	10^{30}	0.34	0.07

Table: Convergence rates measured experimentally for $h = 1.0$ on the finest grid and 1 grid point on the coarsest level using 100 vector iterations (power method).

Image Denoising: Model

Simple variational based on Tikhonov regularization, minimizing the functional

$$E_1(u) = \int_\Omega |u_0 - u|^2 + \alpha |\nabla u|^2 d\mathbf{x} \quad (3)$$

with $\mathbf{x} \in \mathbb{R}^d$ and $\alpha \in \mathbb{R}^+$ over image domain $\Omega \subset \mathbb{R}^d$.

Necessary condition for a minimizer u (the denoised image) is characterized by the Euler-Lagrange equation

$$u - u_0 - \alpha \Delta u = 0 \quad (4)$$

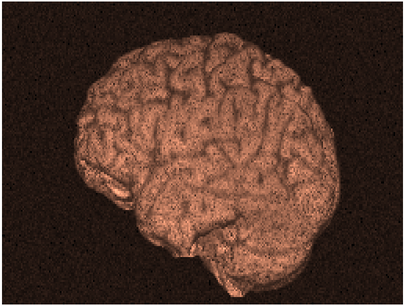
with homogeneous Neumann boundary conditions.

Image Denoising: Computation

1. Convolution of the image with a discrete version of the Gaussian kernel (stencil),
2. multiplication in Fourier space (5) or

$$F[G_\sigma * u_0](\mathbf{w}) = e^{-|\mathbf{x}|^2/(2/\sigma^2)} F[u_0](\mathbf{w}) \quad (5)$$

3. application of a multigrid method to (2).



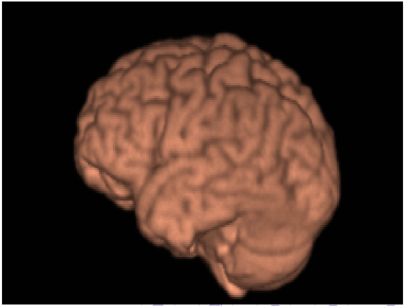


Image Denoising: Runtimes

method	runtime
filtering with mask of size $5 \times 5 \times 5$	1200 ms
filtering with mask of size $3 \times 3 \times 3$	681 ms
$V(1,1)$-cycle	390 ms
FFTW-package	1140 ms

Table: Runtime for image denoising using a 3D MRI image (size $256 \times 256 \times 160$) of a human head with added Gaussian noise, measured on the AMD Opteron platform.

Image Registration: Model I

Variational approach to minimize the energy functional

$$E_2(u) = \int_\Omega (T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x}))^2 + \alpha \sum_{l=1}^{d} \|\nabla u_l\|^2 \, d\mathbf{x} \ . \qquad (6)$$

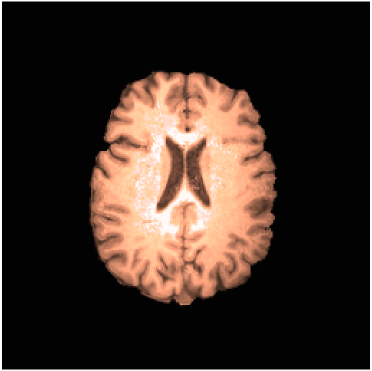
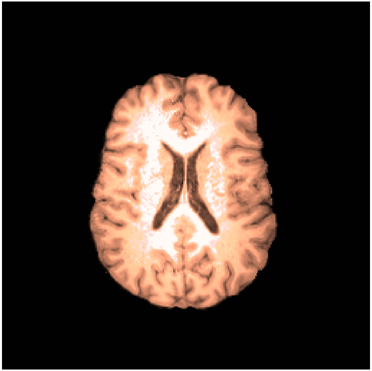



Image Registration: Model II

The optimization of the energy functional results in nonlinear Euler Lagrange equations. To treat the nonlinearity, an artificial time is introduced and discretized by an semi-implicit scheme:

$$\frac{(\mathbf{u}_h^{k+1} - \mathbf{u}_h^k)}{\tau} - \alpha \Delta_h \mathbf{u}_h^{k+1} = \nabla_h T(\mathbf{x} - \mathbf{u}_h^k)\left(T(\mathbf{x} - \mathbf{u}_h^k) - R(\mathbf{x})\right) \qquad (7)$$

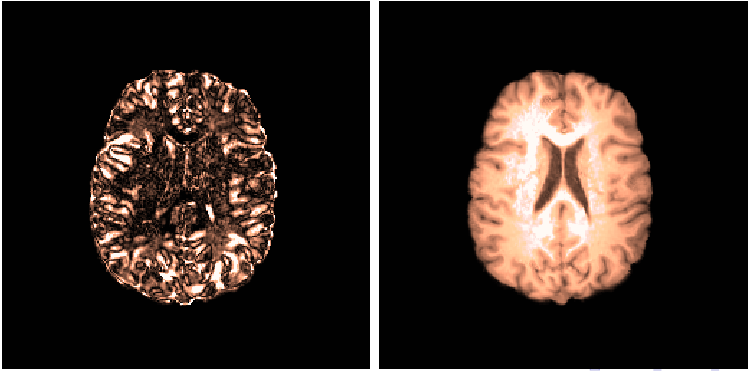


Image Registration: Runtimes

method	runtime
FMG-V(2,2)	608 ms
FMG-V(2,1)	499 ms
FMG-V(1,1)	390 ms
DCT	2107 ms
AOS	1971 ms

Table: Runtime for one linear solve in one time step in the image registration algorithm for an image of size $256 \times 256 \times 160$.

Future Work

- Extend multigrid solver to (an)isotropic diffusion.
- Implement on CBEA.

Convergence Rates According to LFA

interpolation	ω	V(1,1)			V(2,2)		
		ρ	$\rho(M_{3L})$	c_{exp}	ρ	$\rho(M_{3L})$	c_{exp}
constant	1.0	0.20	0.47		0.04	0.07	
constant	1.15	0.08	0.20	0.27	0.04	0.06	0.07
trilinear	1.15	0.08	0.10		0.04	0.06	

Table: Smoothing factor (ρ) and three-grid asymptotic convergence factor ($\rho(M_{3L})$) for size 64^3, $\tau = 10^{30}$ obtained by LFA (using the lfa package from R. Wienands).

Image Denoising: Gaussian Kernel

In an infinite domain an explicit solution is given by

$$u(\mathbf{x}, t) = \int_{\mathbb{R}^d} G_{\sqrt{2t}}(\mathbf{x} - \mathbf{y}) u_0(\mathbf{y}) d\mathbf{y} = (G_{\sqrt{2t}} * u_0)(\mathbf{x}) \;, \qquad (8)$$

where the operator $*$ denotes the convolution of the grid function u_0 and the Gaussian kernel

$$G_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2} e^{-|\mathbf{x}|^2/(2\sigma^2)} \;, \qquad (9)$$

with standard deviation $\sigma \in \mathbb{R}^+$. This is equivalent to applying a low-pass filter and can be transformed into Fourier space, where a convolution corresponds to a multiplication of the transformed signals.

Image Denoising: FFT

If we denote by $F[f]$ the Fourier transform of a signal $f : \mathbb{R}^d \to \mathbb{R}$ and use

$$F[G_\sigma](\mathbf{w}) = e^{-|\mathbf{x}|^2/(2/\sigma^2)}, \mathbf{w} \in \mathbb{R}^d$$

it follows that

$$F[G_\sigma * u_0](\mathbf{w}) = e^{-|\mathbf{x}|^2/(2/\sigma^2)} F[u_0](\mathbf{w}) \quad . \tag{10}$$

Image Registration: Functional

The optimization of the energy functional results in nonlinear Euler Lagrange equations

$$\nabla T(\mathbf{x}-\mathbf{u}(\mathbf{x}))\,(T(\mathbf{x}-\mathbf{u}(\mathbf{x}))-R(\mathbf{x}))+\alpha\Delta\mathbf{u}=0 \tag{11}$$

with homogeneous Neumann boundary conditions that can be discretized by finite differences on a regular grid Ω_h with mesh size h. To treat the nonlinearity often an artificial time is introduced

$$\partial_t\mathbf{u}(\mathbf{x},t)-\alpha\Delta\mathbf{u}(\mathbf{x},t)=\nabla T(\mathbf{x}-\mathbf{u}(\mathbf{x},t))\,(T(\mathbf{x}-\mathbf{u}(\mathbf{x},t))-R(\mathbf{x})) \tag{12}$$

that is discretized by a semi-implicit scheme with a discrete time step τ, where the nonlinear term is evaluated at the old time level

$$\frac{(\mathbf{u}_h^{k+1}-\mathbf{u}_h^k)}{\tau}-\alpha\Delta_h\mathbf{u}_h^{k+1}=\nabla_h T(\mathbf{x}-\mathbf{u}_h^k)\left(T(\mathbf{x}-\mathbf{u}_h^k)-R(\mathbf{x})\right). \tag{13}$$