In B. Andres, B. Rosenhahn (Eds.): Pattern Recognition. Lecture Notes in Computer Science, Vol. 9796, pp. 91–102, Springer, Cham, 2016.
GCPR 2016 Best Paper Award.

FSI Schemes: Fast Semi-Iterative Solvers for PDEs and Optimisation Methods

David Hafner[1], Peter Ochs[1], Joachim Weickert[1], Martin Reißel[2], and Sven Grewenig[1]

[1] Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science, Campus E1.7, Saarland University, 66041 Saarbrücken, Germany
{hafner,ochs,weickert,grewenig}@mia.uni-saarland.de
[2] Fachbereich Medizintechnik und Technomathematik, Standort Jülich, Fachhochschule Aachen, Heinrich-Mußmann-Straße 1, 52428, Jülich, Germany
reissel@fh-aachen.de

Abstract. Many tasks in image processing and computer vision are modelled by diffusion processes, variational formulations, or constrained optimisation problems. Basic iterative solvers such as explicit schemes, Richardson iterations, or projected gradient descent methods are simple to implement and well-suited for parallel computing. However, their efficiency suffers from severe step size restrictions. As a remedy we introduce a simple and highly efficient acceleration strategy, leading to so-called *Fast Semi-Iterative* (FSI) schemes that extrapolate the basic solver iteration with the previous iterate. To derive suitable extrapolation parameters, we establish a recursion relation that connects box filtering with an explicit scheme for 1D homogeneous diffusion. FSI schemes avoid the main drawbacks of recent Fast Explicit Diffusion (FED) and Fast Jacobi techniques, and they have an interesting connection to the heavy ball method in optimisation. Our experiments show their benefits for anisotropic diffusion inpainting, nonsmooth regularisation, and Nesterov's worst case problems for convex and strongly convex optimisation.

1 Introduction

In the present paper we propose efficient numerical solvers for three problem classes in image processing and computer vision: *(i)* diffusion evolutions, *(ii)* variational models leading to elliptic partial differential equations (PDEs), *(iii)* constrained convex optimisation problems.

Diffusion processes have applications e.g. as linear or nonlinear scale-spaces [8, 11, 21]. In the space-discrete case, they are given by dynamical systems of type

$$\partial_t \boldsymbol{u} \;=\; \boldsymbol{A}(\boldsymbol{u})\,\boldsymbol{u}\,, \tag{1}$$

where the vector $\boldsymbol{u} \in \mathbb{R}^N$ contains the grey values in N different pixel locations, $t \in (0, \infty)$ denotes the diffusion time, and $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is a symmetric negative semi-definite matrix that may depend in a nonlinear way on the evolving image $\boldsymbol{u}$. This abstract model holds in any dimension and includes isotropic as well as anisotropic diffusion models with differential operators of second or higher order.

Elliptic problems frequently arise as steady states of diffusion evolutions or as Euler-Lagrange equations of variational models [2]. Space-discrete formulations lead to systems of equations in the form of

$$\boldsymbol{B}(\boldsymbol{u})\,\boldsymbol{u} \;=\; \boldsymbol{d}(\boldsymbol{u})\,, \tag{2}$$

where $\boldsymbol{B} \in \mathbb{R}^{N \times N}$ is symmetric positive definite, and $\boldsymbol{d} \in \mathbb{R}^N$ is the known right hand side. In case of nonlinear evolutions or nonquadratic variational models, the system matrix $\boldsymbol{B}$ and the vector $\boldsymbol{d}$ may depend on the evolving image $\boldsymbol{u}$.

Constrained convex optimisation problems appear e.g. in dual formulations of certain nonsmooth minimisation tasks such as total variation (TV) regularisation [1, 14]. A general framework can be cast as

$$\min_{\boldsymbol{u} \in \mathcal{C}} F(\boldsymbol{u})\,, \tag{3}$$

where $F\colon \mathbb{R}^N \to \mathbb{R}$ is a smooth convex function, and $\mathcal{C}$ denotes a convex set that models the constraint. Such constrained optimisation methods are flexible modelling tools that have a broad range of applications.

For all three problem classes there exist basic iterative schemes, namely *(i)* explicit finite difference schemes, *(ii)* Richardson iterations, and *(iii)* projected gradient descent methods. These schemes are easy to implement and well-suited for parallel architectures such as GPUs. Unfortunately, severe restrictions of the time step sizes or the relaxation parameters render such algorithms rather inefficient. Hence, it would be highly desirable to find acceleration strategies that improve the efficiency of those basic schemes while preserving their advantages.

Our Contributions. We propose an acceleration strategy that consists of a semi-iterative approach in the sense of Varga [19]. It computes the new iterate $\boldsymbol{u}^{k+1}$ by applying the basic iterative scheme to $\boldsymbol{u}^k$ and extrapolating the result by means of $\boldsymbol{u}^{k-1}$. Here, the extrapolation step is responsible for a substantial acceleration. We call such techniques *Fast Semi-Iterative* (FSI) schemes. In contrast to classical semi-iterative approaches from the numerical literature, we obtain different extrapolation parameters that can be derived in an intuitive way from box filter recursions. Box filters are known to give a good compromise between efficiency and numerical stability [22]. On top of that, we uncover theoretical connections of our FSI schemes to well-performing iterative procedures such as Fast Explicit Diffusion (FED) [22] or Polyak's heavy ball method [12].

Paper Organisation. After a discussion of related work, we review 1D linear diffusion and expose its relation to box filtering in Sec. 2. Subsequently, we transfer this concept to iterative schemes and present our novel FSI techniques for diffusion evolutions (Sec. 3), elliptic problems (Sec. 4), and constrained optimisation (Sec. 5). Our experiments in Sec. 6 illustrate the benefits of our algorithms. Finally, we conclude our paper with a summary and outlook in Sec. 7.

Related Work. Our schemes are closely related to nonstationary iterative schemes, where the algorithmic parameters vary from iteration to iteration. In this context, already in 1911, Richardson discussed possible benefits of varying relaxation parameters in his iterative scheme [13]. Later, based on Chebyshev polynomials of the first kind, cyclic parameter choices were proposed that allow substantial speed-ups; see e.g. [24]. Inherently, the Richardson method in [13] is closely related to gradient descent schemes, and thus to the solution of parabolic PDEs. In this context, similar ideas have been proposed by Yaun'Chzhao-Din [25] and Saul'yev [16]. They are known under the name super time stepping [4]. Recently, motivated by box filter factorisations, Weickert et al. [22] propose cyclically varying parameters that substantially improve the damping properties of the resulting schemes. Additionally, the authors introduce a Jacobi-like scheme for elliptic problems. Setzer et al. [17] base on the work of [22] and provide an extension to projection methods with application to nonsmooth optimisation. Furthermore, we can relate such cyclic Richardson approaches to so-called semi-iterative procedures that rely on Chebyshev recursion formulas [5, 19]. Interestingly, these semi-iterative schemes additionally share similarities with Polyak's heavy ball method [12], where Ochs et al. [10] recently proposed an extension that includes proximal mappings. Similarly, our technique relates to the so-called momentum method that is frequently applied in machine learning approaches; see e.g. [15, 18].

2 How to Benefit from Box Filtering

2.1 Explicit Scheme for 1D Linear Diffusion

As starting point of our work we consider linear diffusion of a 1D signal $u(x,t)$:

$$\partial_t u \;=\; \partial_{xx} u\,. \tag{4}$$

With grid size h and time step size τ, an explicit scheme for (4) is given by

$$u_i^{k+1} \;=\; (I + \tau\,\Delta_h)\,u_i^k\,, \tag{5}$$

where I denotes the identity operator, $\Delta_h \coloneqq (1,-2,1)/h^2$ the discrete Laplacian, and u_i^k approximates u in pixel i at time level k. For stability reasons, all stencil weights should be nonnegative. This implies that the time step size must satisfy $\tau \leq h^2/2$. Obviously, this restriction makes such an explicit scheme inefficient: With n explicit diffusion steps, we can only reach a stopping time of $\mathcal{O}(n)$.

2.2 Box Filtering via Iterative Explicit Diffusion

In order to use explicit schemes more efficiently, let us make a didactic excursion to box filters. A box filter B_{2n+1} of length $(2n+1)h$ is given by

$$[B_{2n+1}\, u]_i \;=\; \frac{1}{2n+1} \sum_{j=-n}^{n} u_{i+j}. \tag{6}$$

It is well-known [6] that linear diffusion with stopping time T is equivalent to a convolution with a Gaussian of variance $\sigma^2 = 2T$. Moreover, the central limit theorem tells us that iterated box filtering approximates Gaussian convolution. Indeed, an m-fold iteration of a box filter B_{2n+1} with variance σ_n^2 approximates a Gaussian with variance $m\,\sigma_n^2$. The variance of B_{2n+1} is given by (cf. also [23])

$$\sigma_n^2 \;=\; \frac{1}{2n+1} \sum_{j=-n}^{n} (jh-0)^2 \;=\; \frac{2h^2}{2n+1} \frac{n(n+1)(2n+1)}{6} \;=\; \frac{n(n+1)}{3} h^2\,. \tag{7}$$

This implies that a single application of a box filter B_{2n+1} approximates linear diffusion with stopping time $T_n = \sigma_n^2/2 = \frac{n(n+1)}{3}\frac{h^2}{2}$. Note that this stopping time is $\mathcal{O}(n^2)$. Hence, if we were able to implement B_{2n+1} by means of n explicit linear diffusion steps, we could accelerate the explicit scheme from $\mathcal{O}(n)$ to $\mathcal{O}(n^2)$. To this end, we introduce the following theorem which can be proven by induction:

Theorem 1 (Connection of Box Filters and Explicit Diffusion).
A box filter B_{2n+1} of length $(2n+1)h$ can be constructed iteratively by n explicit linear diffusion steps:

$$B_{2k+3} \;=\; \alpha_k \cdot (I + \tau \Delta_h)\, B_{2k+1} \;+\; (1-\alpha_k) \cdot B_{2k-1} \qquad (k = 0, \ldots, n-1)$$

with $\tau := h^2/2$, $\alpha_k := (4k+2)/(2k+3)$, and $B_{-1} := I$.

Note that for $k = 0$ we have $B_3 = I + \frac{h^2}{3}\Delta_h$, which is a single diffusion step with time step size $\frac{2}{3}\tau$.

2.3 Accelerating the Explicit Scheme for 1D Linear Diffusion

To apply Theorem 1 for accelerating the explicit diffusion scheme (5), let us first rewrite it in matrix-vector notation:

$$\boldsymbol{u}^{k+1} \;=\; (\boldsymbol{I} + \tau\, \boldsymbol{L})\, \boldsymbol{u}^k\,, \tag{8}$$

where the vector $\boldsymbol{u} \in \mathbb{R}^N$ contains the discrete entries of u, $\boldsymbol{I} \in \mathbb{R}^{N \times N}$ is the identity matrix, and the symmetric negative semi-definite matrix $\boldsymbol{L} \in \mathbb{R}^{N \times N}$ implements the Laplacian. The box filter relation in Theorem 1 suggests the following scheme to accelerate the explicit diffusion scheme (8) such that $\boldsymbol{u}^k$ corresponds to an application of a box filter B_{2k+1}:

$$\boldsymbol{u}^{k+1} \;=\; \alpha_k \cdot (\boldsymbol{I} + \tau\, \boldsymbol{L})\, \boldsymbol{u}^k \;+\; (1-\alpha_k) \cdot \boldsymbol{u}^{k-1} \tag{9}$$

with $\tau = h^2/2$, $\alpha_k = (4k+2)/(2k+3)$, and $\boldsymbol{u}^{-1} := \boldsymbol{u}^0$. As we have seen, n iterations of this scheme implement a box filter B_{2n+1} of length $(2n+1)h$. However, a single box filter might be a poor approximation for the actual linear diffusion process that is equivalent to Gaussian convolution. To improve the approximation quality, we should iterate the box filter. Hence, we propose a *cyclic* application of (9), where the m-th cycle with cycle length n is given by

$$\boxed{\begin{array}{c} \boldsymbol{u}^{m,k+1} \;=\; \alpha_k \cdot \big(\boldsymbol{I} + \tau\, \boldsymbol{L}\big)\, \boldsymbol{u}^{m,k} \;+\; (1-\alpha_k) \cdot \boldsymbol{u}^{m,k-1} \\ \text{with } \; \boldsymbol{u}^{m,-1} := \boldsymbol{u}^{m,0} \; \text{ and } \; \alpha_k = (4k+2)/(2k+3) \; \text{ for } \; k = 0, \ldots, n-1. \end{array}} \tag{10}$$

Here, the number of cycles is responsible for the accuracy, while the cycle length n accounts for the $\mathcal{O}(n^2)$ efficiency. For the next cycle, we set $\boldsymbol{u}^{m+1,0} := \boldsymbol{u}^{m,n}$.

3 FSI Schemes for Diffusion Evolutions

Our discussion so far was for didactic reasons only, since linear diffusion can be implemented directly as an efficient box filter without explicit iterations. However, it suggests how we could generalise these ideas to arbitrary isotropic or anisotropic nonlinear diffusion processes (1) that have explicit schemes of type

$$\boldsymbol{u}^{k+1} \;=\; \big(\boldsymbol{I} + \tau\, \boldsymbol{A}(\boldsymbol{u}^k)\big)\, \boldsymbol{u}^k \,. \tag{11}$$

Since $\boldsymbol{A}$ is negative semi-definite, stability in the Euclidean norm requires the time step size restriction $0 < \tau < 2/\rho(\boldsymbol{A}(\boldsymbol{u}^k))$, where ρ denotes the spectral radius. Obviously, there is a strong similarity of (11) to the linear diffusion scheme in (8). Hence, it appears to be natural to formulate, in analogy to (10), the following *Fast Semi-Iterative* (FSI) scheme:

$$\boxed{\begin{array}{c} \boldsymbol{u}^{m,k+1} \;=\; \alpha_k \cdot \big(\boldsymbol{I} + \tau\, \boldsymbol{A}(\boldsymbol{u}^{m,k})\big)\, \boldsymbol{u}^{m,k} \;+\; (1-\alpha_k) \cdot \boldsymbol{u}^{m,k-1} \\ \text{with } \; \boldsymbol{u}^{m,-1} := \boldsymbol{u}^{m,0} \; \text{ and } \; \alpha_k = (4k+2)/(2k+3) \; \text{ for } \; k = 0, \ldots, n-1. \end{array}} \tag{12}$$

This scheme describes the m-th cycle with length n of our FSI algorithm. We repeat it several times to reach a specific stopping time. Similar to [7], one can conduct an analysis of the internal stability of this iterative scheme. This way, one can show that for every iteration our scheme is stable in a suitable norm. This stability analysis requires the symmetry of $\boldsymbol{A}$.

Connection to Fast Explicit Diffusion (FED). As discussed in Sec. 2, we base on the connection between box filtering and explicit schemes. This was inspired by the cyclic FED approach of Weickert et al. [22] which demonstrates the benefits of a reliance on box filters. More specifically, they exploit a factorisation of a box filter into several explicit diffusion steps to construct their algorithms. In case of linear problems, one can even show that FED and FSI schemes yield identical results after each cycle. However, in case of nonlinear problems, the

discussed semi-iterative structure of our FSI scheme is highly beneficial since it allows to perform nonlinear updates within one cycle. We illustrate this by means of our experiments in Sec. 6. Moreover, FED schemes are highly sensitive to numerical inaccuracies. Thus, sophisticated rearrangements of possibly unstable time steps are required to avoid the explosion of rounding errors; see e.g. [22] and references therein. We eliminate this drawback by our FSI technique.

4 FSI Schemes for Elliptic Problems

So far, we have considered diffusion-like processes that correspond to parabolic PDEs. Next we explain how to transfer this concept to discretised elliptic PDEs, or more generally to the solution of equation systems in the form of (2). In the linear case, formulating (2) as $\boldsymbol{u} = \boldsymbol{u} - \omega(\boldsymbol{B}\boldsymbol{u} - \boldsymbol{d})$ gives rise to the Richardson scheme [13]

$$\boldsymbol{u}^{k+1} \;=\; (\boldsymbol{I} - \omega\,\boldsymbol{B})\,\boldsymbol{u}^k + \omega\,\boldsymbol{d}\,. \tag{13}$$

Choosing $0 < \omega < 2/\rho(\boldsymbol{B})$ guarantees stability in the Euclidean norm, since the eigenvalues of $\boldsymbol{I} - \omega\,\boldsymbol{B}$ lie in $(-1, 1]$. Considering the error vector $\boldsymbol{e}^k = \boldsymbol{u}^k - \boldsymbol{u}^*$ between the current estimate $\boldsymbol{u}^k$ and the unknown exact solution $\boldsymbol{u}^*$ yields

$$\boldsymbol{e}^{k+1} \;=\; (\boldsymbol{I} - \omega\,\boldsymbol{B})\,\boldsymbol{e}^k\,. \tag{14}$$

We observe a strong similarity of (14) to the explicit diffusion scheme in (8). Hence, we propose the following FSI scheme for elliptic problems:

$$\boxed{\begin{array}{c} \boldsymbol{u}^{m,k+1} \;=\; \alpha_k \cdot \big((\boldsymbol{I} - \omega\,\boldsymbol{B})\,\boldsymbol{u}^{m,k} + \omega\,\boldsymbol{d}\big) \;+\; (1 - \alpha_k) \cdot \boldsymbol{u}^{m,k-1} \\ \text{with } \; \boldsymbol{u}^{m,-1} \coloneqq \boldsymbol{u}^{m,0} \;\text{ and }\; \alpha_k = (4k+2)/(2k+3) \;\text{ for }\; k = 0, \ldots, n-1. \end{array}} \tag{15}$$

To extend (15) to nonlinear systems of equations in the form of (2), we replace $\boldsymbol{B}$ and $\boldsymbol{d}$ by their nonlinear counterparts $\boldsymbol{B}(\boldsymbol{u}^{m,k})$ and $\boldsymbol{d}(\boldsymbol{u}^{m,k})$, and choose $0 < \omega < 2/L$, where $L > 0$ is the Lipschitz constant of $\boldsymbol{B}(\boldsymbol{u})\boldsymbol{u} - \boldsymbol{d}(\boldsymbol{u})$.

Preconditioning. The discussed FSI method yields fast convergence for problems where the coefficients of the equation system have a similar value of magnitude. However, in case of strongly differing coefficients a preconditioning or, in other words, a different splitting of the system matrix $\boldsymbol{B}$, is highly beneficial. As an example, we consider the Jacobi overrelaxation splitting $\boldsymbol{B} = \frac{1}{\omega}\boldsymbol{D} + (\boldsymbol{B} - \frac{1}{\omega}\boldsymbol{D})$, where $\boldsymbol{D}$ denotes a positive definite diagonal matrix. This leads to

$$\boldsymbol{u}^{m,k+1} \;=\; \alpha_k \cdot \big((\boldsymbol{I} - \omega\,\boldsymbol{D}^{-1}\boldsymbol{B})\,\boldsymbol{u}^{m,k} + \omega\,\boldsymbol{D}^{-1}\boldsymbol{d}\big) \;+\; (1 - \alpha_k) \cdot \boldsymbol{u}^{m,k-1}\,. \tag{16}$$

Assuming a symmetric positive definite matrix $\boldsymbol{B}$, this process is stable with a suitable ω such that the eigenvalues of $\boldsymbol{I} - \omega\,\boldsymbol{D}^{-1}\boldsymbol{B}$ lie in $(-1, 1]$.

Connection to Fast Jacobi. Similar to the connection to FED in the parabolic case, we can show a relation of our FSI scheme to the recent Fast Jacobi solver of Weickert et al. [22] for elliptic problems. As before, we eliminate the drawback of rearranging the relaxation parameters and provide an internal stability that allows for intermediate nonlinear updates.

5 FSI Schemes for Constrained Optimisation

Often, the elliptic problem from the previous section can be interpreted as the minimality condition of a suitable optimisation problem. In fact, the gradient descent scheme to compute a minimiser of the optimisation problem (3) without side-constraints is given by

$$\boldsymbol{u}^{k+1} \;=\; \boldsymbol{u}^k - \omega\,\boldsymbol{\nabla} F(\boldsymbol{u}^k)\,. \tag{17}$$

This scheme is stable for $0<\omega<2/L$, where $L>0$ is the Lipschitz constant of $\boldsymbol{\nabla} F$. Again, the structural similarity to (8) suggests the following FSI iteration:

$$\boldsymbol{u}^{m,k+1} \;=\; \alpha_k \cdot \big(\boldsymbol{u}^k - \omega\,\boldsymbol{\nabla} F(\boldsymbol{u}^{m,k})\big) \;+\; (1-\alpha_k)\cdot \boldsymbol{u}^{m,k-1}\,. \tag{18}$$

Adaptation to Constrained Problems. As it turns out, the provided internal stability of our algorithm enables us to perform projections onto convex sets in every iteration step. Thus, our technique is additionally well-suited for constrained optimisation problems in the form of (3), where the solution $\boldsymbol{u}$ is constrained to some convex set $\mathcal{C}$. With the corresponding orthogonal projection operator $P_{\mathcal{C}}$, our FSI scheme for constrained optimisation is given by

$$\boxed{\begin{array}{c} \boldsymbol{u}^{m,k+1} \;=\; P_{\mathcal{C}}\Big(\alpha_k \cdot \big(\boldsymbol{u}^k - \omega\,\boldsymbol{\nabla} F(\boldsymbol{u}^{m,k})\big) + (1-\alpha_k)\cdot \boldsymbol{u}^{m,k-1}\Big) \\ \text{with } \boldsymbol{u}^{m,-1} \coloneqq \boldsymbol{u}^{m,0} \text{ and } \alpha_k = (4k+2)/(2k+3) \text{ for } k=0,\dots,n-1. \end{array}} \tag{19}$$

Adaptation to Strongly Convex Problems. So far, we have considered the case where the Lipschitz constant L is assumed to be known. However, strongly convex problems additionally provide information about the strong convexity parameter ℓ. To make use of this additional knowledge, we propose the following recursive parameter choice to accelerate our iterative scheme:

$$\alpha_k \;=\; \frac{1}{1-\frac{\alpha_{k-1}}{4}\cdot\left(\frac{L-\ell}{L+\ell}\right)^2} \qquad (k=1,\dots,n-1)\,, \tag{20}$$

where $\alpha_0 = 2(L+\ell)/(3L+\ell)$ and $\omega = 2/(L+\ell)$. It can be derived using classical Chebyshev reasonings. With $\ell = 0$, these parameters come down to (18). We present a full description of our FSI method in Algorithm 1. Also in the elliptic case (Sec. 4) we can apply such a parameter choice. Here, ℓ and L correspond to the smallest and largest eigenvalues of the positive definite system matrix $\boldsymbol{B}$.

Algorithm 1: FSI scheme with projection operator $P_{\mathcal{C}}$ and $0 \leq \ell < L$.

input : ∇F, $P_{\mathcal{C}}$, L, ℓ, n, $\boldsymbol{u}^0$

$\omega = \frac{2}{L+\ell}$, $\quad \alpha_0 = 2(L+\ell)/(3L+\ell)$, $\quad \alpha_k = 1 \Big/ \left(1 - \frac{\alpha_{k-1}}{4} \cdot \left(\frac{L-\ell}{L+\ell}\right)^2\right)$

$\boldsymbol{u}^{0,-1} = \boldsymbol{u}^{0,0} = \boldsymbol{u}^0$

for $m = 0, 1, \ldots$ **do**

  for $k = 0, 1, \ldots, n-1$ **do**

    $\boldsymbol{u}^{m,k+1} = P_{\mathcal{C}}\Big(\alpha_k \left(\boldsymbol{u}^k - \omega \, \nabla F(\boldsymbol{u}^{m,k})\right) + (1-\alpha_k)\, \boldsymbol{u}^{m,k-1}\Big)$

  $\boldsymbol{u}^{m+1,-1} = \boldsymbol{u}^{m+1,0} = \boldsymbol{u}^{m,n}$

Connection to Heavy Ball Method. Let us now derive a close relation of the proposed FSI scheme to Polyak's heavy ball method [12]

$$\boldsymbol{u}^{k+1} \;=\; \boldsymbol{u}^k \;-\; \tilde{\alpha} \cdot \nabla F(\boldsymbol{u}^k) \;+\; \tilde{\beta} \cdot (\boldsymbol{u}^k - \boldsymbol{u}^{k-1})\,. \tag{21}$$

The first part of this iterative scheme can be seen as a gradient descent step, while the second part represents an inertial term. It relates the current iterate $\boldsymbol{u}^k$ to the old time step $\boldsymbol{u}^{k-1}$. This allows significant speed-ups. Interestingly, we can connect (21) to our FSI approach by applying *cyclically varying* parameters $\tilde{\alpha}_k = \omega\alpha_k$ and $\tilde{\beta}_k = \alpha_k - 1$ with α_k given by (18), combined with a restart $\boldsymbol{u}^{m,-1} \coloneqq \boldsymbol{u}^{m,0}$ after each cycle. We illustrate benefits of our approach in Sec. 6.

6 Experiments

6.1 Inpainting with Edge-Enhancing Anisotropic Diffusion

In our first experiment, we consider image inpainting by means of edge-enhancing anisotropic diffusion [20]:

$$\partial_t u \;=\; (1 - c(\boldsymbol{x})) \cdot \operatorname{div}\left(\boldsymbol{D}(\nabla u_\sigma)\, \nabla u\right) \;-\; c(\boldsymbol{x}) \cdot (u - f)\,, \tag{22}$$

where $c\colon \Omega \to [0,1]$ is a binary mask that indicates known data, and $\Omega \subset \mathbb{R}^2$ denotes the rectangular image domain. Furthermore, $\boldsymbol{D}(\nabla u_\sigma) \coloneqq \mu_1 \cdot \boldsymbol{v}_1 \boldsymbol{v}_1^\top + \mu_2 \cdot \boldsymbol{v}_2 \boldsymbol{v}_2^\top$ is the so-called *diffusion tensor*, where the first eigenvector $\boldsymbol{v}_1 \parallel \nabla u_\sigma$ points across image edges, and the second one $\boldsymbol{v}_2 \perp \nabla u_\sigma$ along them. Here, u_σ denotes convolution of u with a Gaussian of standard deviation σ. Since we want to perform full diffusion along edges but reduced smoothing across them, we set $\mu_2 = 1$ and determine μ_1 by means of the Charbonnier diffusivity with a contrast parameter $\lambda > 0$, i.e. $\mu_1 = 1/\sqrt{1+|\nabla u_\sigma|^2/\lambda^2}$.

Fig. 1 depicts the sparse input data as well as our inpainting results. First, the provided speed-up by our FSI technique compared to the baseline explicit diffusion (ED) scheme is obvious. Moreover, we perform nonlinear updates after *each* iteration step. As discussed in [22], FED schemes inherently do not allow for such intermediate updates within one cycle. Hence, we offer more flexibility in this regard which further allows for a better performance.

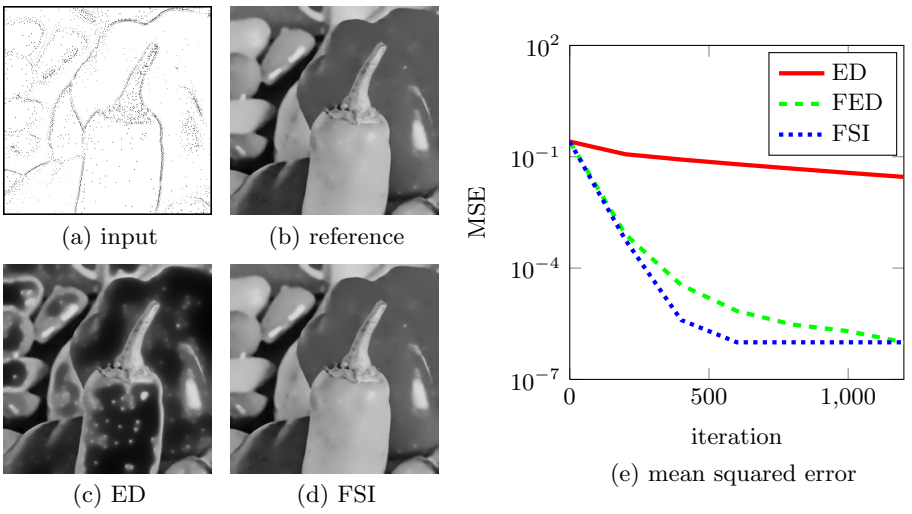


Fig. 1. Inpainting with edge-enhancing anisotropic diffusion, initialised with a black image. (a) Input with inverted colours for visibility reasons. (b) Converged reference result of a standard explicit diffusion (ED) scheme. (c) Result of ED after 600 iterations. (d) Result of FSI after three cycles with length $n = 200$. (e) Mean squared error (MSE) between current estimates and reference solution. The grey value range is $[0, 1]$.

6.2 Total Variation Regularisation

In this experiment, we consider image regularisation by means of the following energy functional [1, 14]:

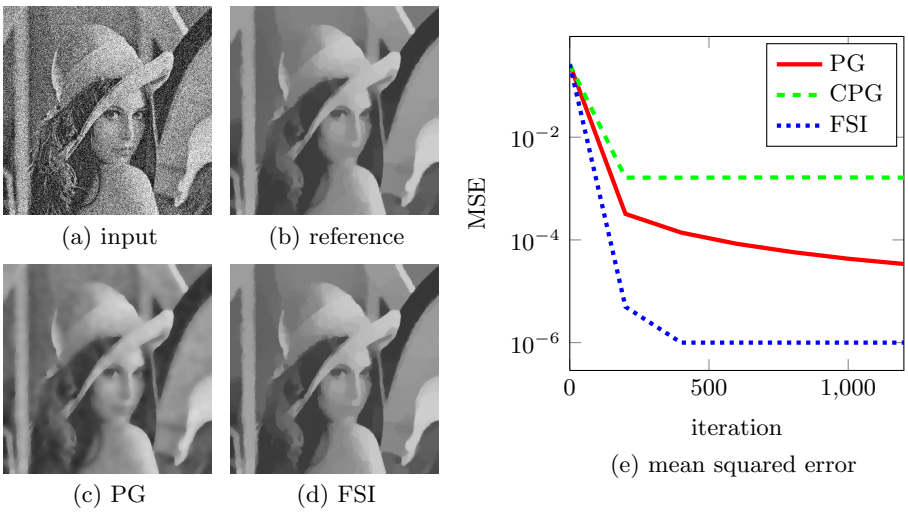
$$E(u) \;=\; \frac{1}{2}\int_\Omega (u-f)^2 \,\mathrm{d}\boldsymbol{x} \;+\; \gamma\,\mathrm{TV}(u)\,, \tag{23}$$

where $f\colon \Omega \to \mathbb{R}$ represents a noisy input image, and the parameter $\gamma > 0$ steers the amount of smoothness. The regulariser $\mathrm{TV}(u) := \sup_{\|\boldsymbol{p}\|_\infty \le 1} \int_\Omega u \operatorname{div} \boldsymbol{p}\,\mathrm{d}\boldsymbol{x}$ penalises the total variation of u, and $\boldsymbol{p}\colon \Omega \to \mathbb{R}^2$ is a smooth vector field with compact support on Ω. To compute the minimiser of (23), we solve the dual problem [3]

$$\min_{\|\boldsymbol{p}\|_\infty \le \gamma} \frac{1}{2}\int_\Omega (f - \operatorname{div}\boldsymbol{p})^2 \,\mathrm{d}\boldsymbol{x}\,. \tag{24}$$

We discretise this constrained optimisation problem and solve it with the proposed FSI scheme in Algorithm 1. In this example, the projection operator is given point-wise by $P_{\mathcal{C}}(\boldsymbol{p}) = \boldsymbol{p}/\max\{1, |\boldsymbol{p}|/\gamma\}$.

Fig. 2 shows the noisy input image and our smoothed results. Also here, the provided acceleration of the baseline projected gradient (PG) scheme by our FSI approach is obvious. The comparison to the cyclic FED-like projection method by Setzer et al. [17] (CPG) demonstrates that our FSI scheme is stable under



(a) input (b) reference (c) PG (d) FSI (e) mean squared error

Fig. 2. Total variation (TV) regularisation. (a) Input with white Gaussian noise. (b) Converged reference result of standard projected gradient (PG) approach. (c) Result of PG after 200 iterations. (d) Result of FSI after one cycle with length $n = 200$. (e) Mean squared error (MSE) between current estimates and reference solution.

the applied projections within each iteration, while the standard CPG approach without backtracking does not converge properly for such large cycle lengths.

6.3 Performance on Nesterov's Worst Case Problems

In our last experiment, we evaluate the performance of our FSI techniques w.r.t. related iterative schemes on Nesterov's worst case problems for smooth convex and strongly convex optimisation [9]. These are quadratic minimisation problems which are difficult for any algorithm that can solve all instances of the respective class of problems. In [9, Thm. 2.1.13], Nesterov provides a lower bound for all iterations and predicts a linear convergence rate on the class of *smooth strongly convex* problems. The lower bound for *smooth convex* problems derived in [9, Thm. 2.1.7] is only valid for one specific iteration count $k \in \mathbb{N}$ (here: $k = 50$). Nothing is said about the error before and after the k-th iteration.

Fig. 3 plots for both problems the resulting error curves of different solvers. While our FSI scheme offers state-of-the-art performance for the strongly convex problem, it even outperforms competing methods in case of convex optimisation. In particular, this illustrates the benefits of our non-stationary cyclic parameter choice compared to Polyak's heavy ball method.

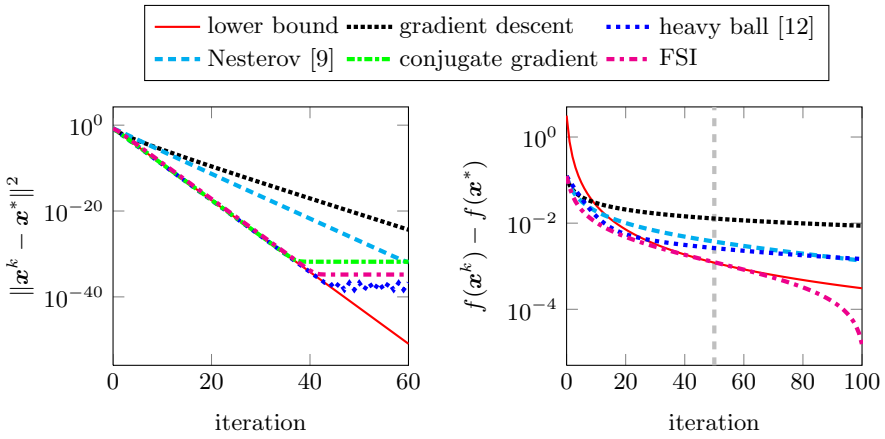


Fig. 3. Numerical comparisons by means of worst case functions of Nesterov [9]. *Left*: Strongly convex optimisation. *Right*: Convex optimisation. Here, the lower bound only holds for $k = 50$ (grey line). In both cases, the problem dimension is $N = 10^5$.

7 Conclusions and Future Work

We have presented *Fast Semi-Iterative* (FSI) schemes that offer efficient solutions for diffusion evolutions, elliptic problems, and constrained optimisation. The proposed schemes are simple to implement and well-suited for parallel implementations. Hence, they are applicable for a wide range of image processing and computer vision tasks. More specifically, we investigated the relation between box filtering and explicit schemes. It is fascinating to see how this simple concept generalises to flexible and highly efficient algorithms. In contrast to FED-like approaches, the provided internal stability of our techniques allows for performing nonlinear updates and projections within each iteration. Our experiments demonstrate these benefits w.r.t. related iterative procedures. In future work, we plan to conduct a deeper theoretical analysis of the presented approaches which may allow us to state precise convergence rates. Furthermore, we plan to extend and generalise our schemes, e.g. to general non-expansive proximal operators and nonsymmetric system matrices.

Acknowledgements. Our research has been partially funded by the *Deutsche Forschungsgemeinschaft* (DFG) through a Gottfried Wilhelm Leibniz Prize for Joachim Weickert and by the Graduate School of Computer Science at Saarland University. This is gratefully acknowledged.

References

1. Acar, R., Vogel, C.R.: Analysis of bounded variation penalty methods for ill-posed problems. Inverse Problems 10, 1217–1229 (1994)

2. Bertero, M., Poggio, T.A., Torre, V.: Ill-posed problems in early vision. Proceedings of the IEEE 76(8), 869–889 (Aug 1988)
3. Chambolle, A.: An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision 20(1), 89–97 (Jan 2004)
4. Gentzsch, W., Schlüter, A.: Über ein Einschrittverfahren mit zyklischer Schrittweitenänderung zur Lösung parabolischer Differentialgleichungen. Zeitschrift für Angewandte Mathematik und Mechanik 58, T415–T416 (1978), in German
5. Golub, G.H., Varga, R.S.: Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods, Part I. Numerische Mathematik 3(1), 147–156 (Dec 1961)
6. Hellwig, G.: Partial Differential Equations. Teubner, Stuttgart (1977)
7. van der Houwen, P.J., Sommeijer, B.P.: On the internal stability of explicit, m-stage Runge-Kutta methods for large m-values. Zeitschrift für Angewandte Mathematik und Mechanik 60(10), 479–485 (1980)
8. Iijima, T.: Basic theory on normalization of pattern (in case of typical one-dimensional pattern). Bulletin of the Electrotechnical Laboratory 26, 368–388 (Jan 1962), in Japanese
9. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course, Applied Optimization, vol. 87. Kluwer, Boston (Dec 2004)
10. Ochs, P., Brox, T., Pock, T.: iPiasco: Inertial proximal algorithm for strongly convex optimization. Journal of Mathematical Imaging and Vision 53(2), 171–181 (Oct 2015)
11. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–639 (Jul 1990)
12. Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics 4(5), 1–17 (1964)
13. Richardson, L.F.: The approximate arithmetical solution by finite differences of physical problems involving differential equation, with an application to the stresses in a masonry dam. Philosophical Transactions of the Royal Society A 210, 307–357 (Jan 1911)
14. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)
15. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, chap. 8, pp. 318–362. MIT Press, Cambridge, MA (1986)
16. Saul'yev, V.K.: Integration of Equations of Parabolic Type by the Method of Nets. Pergamon, Oxford (1964)
17. Setzer, S., Steidl, G., Morgenthaler, J.: A cyclic projected gradient method. Computational Optimization and Applications 54(2), 417–440 (Mar 2013)
18. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: Proc. 30th International Conference on Machine Learning. pp. 1139–1147. Atlanta, GA (Jun 2013)
19. Varga, R.S.: A comparison of the successive overrelaxation method and semi-iterative methods using Chebyshev polynomials. Journal of the Society for Industrial and Applied Mathematics 5(2), 39–46 (Jun 1957)
20. Weickert, J.: Theoretical foundations of anisotropic diffusion in image processing. Computing Supplement 11, 221–236 (1996)
21. Weickert, J.: Anisotropic Diffusion in Image Processing. Teubner, Stuttgart (1998)

22. Weickert, J., Grewenig, S., Schroers, C., Bruhn, A.: Cyclic schemes for PDE-based image analysis. International Journal of Computer Vision 118(3), 275–299 (Jul 2016)
23. Wells III, W.M.: Efficient synthesis of Gaussian filters by cascaded uniform filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(2), 234–239 (Mar 1986)
24. Young, D.M.: On Richardson's method for solving linear systems with positive definite matrices. Journal of Mathematics and Physics 32(1), 243–255 (Apr 1954)
25. Yuan'Chzhao-Din: Some difference schemes for the solution of the first boundary value problem for linear differential equations with partial derivatives. Ph.D. thesis, Moscow State University (1958), in Russian