In T. Brox, A. Bruhn, M. Fritz (Eds.): Pattern Recognition. Lecture Notes in Computer Science, Vol. 11269, pp. 669-681, Springer, Cham, 2019.
The final publication is available at link.springer.com.

AFSI: Adaptive Restart for Fast Semi-Iterative Schemes for Convex Optimisation

Jón Arnar Tómasson[1], Peter Ochs[2], and Joachim Weickert[1]

[1] Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science, Campus E1.7, Saarland University, 66041 Saarbrücken, Germany
{tomasson,weickert}@mia.uni-saarland.de
[2] Mathematical Optimization Group, Faculty of Mathematics and Computer Science, Campus E1.7, Saarland University, 66041 Saarbrücken, Germany
ochs@math.uni-sb.de

Abstract. Smooth optimisation problems arise in many fields including image processing, and having fast methods for solving them has clear benefits. Widely and successfully used strategies to solve them are accelerated gradient methods. They accelerate standard gradient-based schemes by means of extrapolation. Unfortunately, most acceleration strategies are generic, in the sense, that they ignore specific information about the objective function. In this paper, we implement an adaptive restarting into a recently proposed efficient acceleration strategy that was coined Fast Semi-Iterative (FSI) scheme. Our analysis shows clear advantages of the adaptive restarting in terms of a theoretical convergence rate guarantee and state-of-the-art performance on a challenging image processing task.

1 Introduction

The high dimensionality of many variational problems in image processing or computer vision dictates the usage of first-order optimisation algorithms. These are iterative schemes that combine gradient information to construct a sequence of improving approximations to a solution of the variational problem. The simplest instance is the well-known Steepest Descent Method. While the complexity of each iteration is usually very cheap, their efficiency highly depends on the curvature of the objective function to be minimised. In flat regions, short gradient vectors must be compensated by large step sizes while in steep regions the opposite configuration appears.

Essentially, this information is captured by the second derivative (Hessian) of the objective function, which is not available for first-order methods. A successful strategy to deal with this problem is provided by accumulating momentum in

steep regions, which is used in flat regions to preserve the speed. This is the underlying idea of *accelerated gradient schemes*, which are widely used to solve the aforementioned problems. The accelerated gradient scheme that we consider is the Fast Semi-Iterative Scheme (FSI) [6]. The efficiency of this method comes from a clever extrapolation step with cyclically varying parameters.

However, this is an idealised picture. Around a steep local minimum or along a long ramp, accumulating too much momentum leads to oscillatory behaviour [1, 15]. These are just two different (simplified) pictures of optimisation scenarios, which indicate that optimisation algorithms should *adapt* to the objective function. Of course, the situation is significantly more complex for high dimensional problems.

In this paper, we introduce an adaptive restart strategy for FSI. Our adaptation rule comes with several significant advantages: (i) The algorithm *automatically selects* the cycle length of FSI, an otherwise problem-sensitive *parameter*; (ii) it efficiently solves a wide variety of problems, and shows state-of-the-art performance on a difficult image processing problem; and (iii) we prove a worst-case convergence rate, which is not available for the FSI method.

Paper Organisation. Section 2 introduces the underlying accelerated gradient method, FSI, which directly leads to the difficult question of selecting its cycle length parameter. After discussing related work in Section 3, we introduce our adaptive restarting that automatically selects a good cycle length for FSI in Section 4. The convergence of this adaptive FSI (AFSI) scheme is studied in Section 5, and Section 6 demonstrated the high quality of AFSI in image processing. Section 7 concludes the paper and provides a brief outlook.

2 Fast Semi-Iterative Schemes

FSI schemes have been introduced recently by Hafner et al. [6]. They are versatile strategies that accelerate the simplest solvers for four problem classes: the explicit scheme for parabolic partial differential equations, Richardson's iteration for linear systems of equations, the gradient descent method for convex optimisation problems, and the projected gradient descent method for constrained convex optimisation problems. This acceleration is achieved by an extrapolation step in the direction from the previous to the current iterate. While FSI schemes have been introduced in the context of image analysis applications, they have also been used successfully in other fields [2].
In the present paper we are interested in the FSI schemes that solve smooth convex optimisation problems of type

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{\nabla} F$ is Lipschitz continuous with constant L. The classical gradient descent method for this problem reads

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \omega\, \boldsymbol{\nabla} F(\boldsymbol{x}^k)\,, \tag{2}$$

where the upper index denotes the iteration number, and the step size parameter ω must satisfy $\omega \in (0, 2/L)$ for stability reasons.
FSI accelerates gradient descent by considering the cyclic iteration

$$\boldsymbol{x}^{m,\,k+1} = \boldsymbol{x}^{m,\,k} - \alpha_k\,\omega\,\boldsymbol{\nabla} F(\boldsymbol{x}^{m,\,k}) + (\alpha_k - 1)(\boldsymbol{x}^{m,\,k} - \boldsymbol{x}^{m,\,k-1}) \tag{3}$$

with $\boldsymbol{x}^{m,-1} := \boldsymbol{x}^{m,0}$ and extrapolation parameter $\alpha_k = \frac{4k+2}{2k+3}$. The inner iteration index k ranges from 0 to $K-1$, where K denotes the cycle length. The outer index m counts the cycles. In [6] it has been argued that the cycle length K is responsible for the efficiency of the method, while the number of cycles influences the accuracy. In practice there is a natural tradeoff between both parameters, and it requires hand tuning to obtain the highest convergence speed for a given number of iterations. This may be burdensome. Therefore, in this paper, we suggest an automatic adaptive selection of this parameter.

For strongly convex problems with convexity parameter μ (i.e. $F(\boldsymbol{x}) - \frac{\mu}{2}||\boldsymbol{x}||^2$ is convex), Hafner et al. [6] suggest to consider only a single cycle with step size $\omega = \frac{2}{L+\mu}$ and modified extrapolation parameters

$$\alpha_0 = \frac{2(L+\mu)}{3L+\mu}, \quad \alpha_k = \frac{1}{1 - \frac{\alpha_{k-1}}{4}\left(\frac{L-\mu}{L+\mu}\right)^2}\,. \tag{4}$$

While this removes the need to select the cycle length it requires us to know the strong convexity parameter μ.

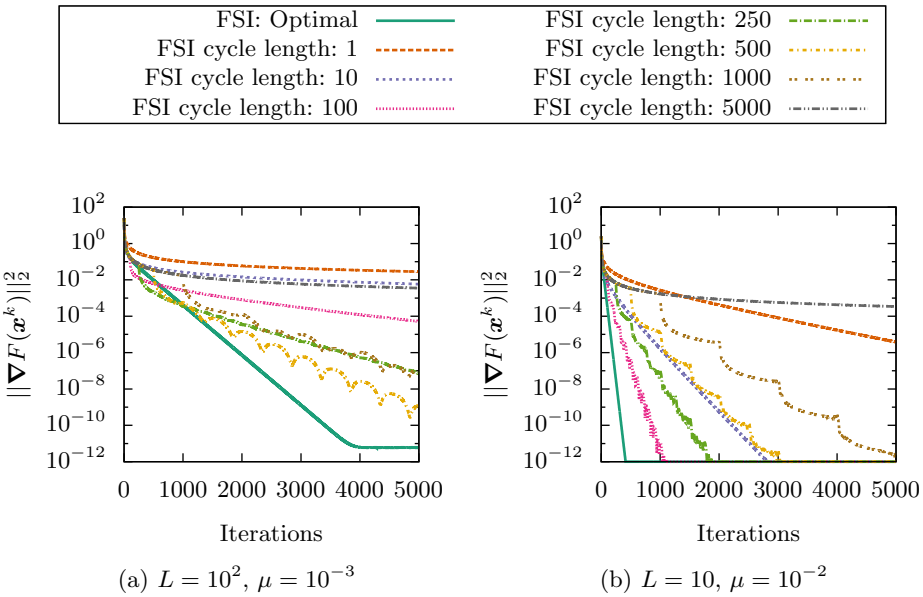
Fig. 1 demonstrates the difficulty of selecting the cycle length of FSI. It shows the performance of different FSI cycle lengths compared to FSI that has been adapted to the strong convexity. The function being minimised is Nesterov's worst case strongly convex function. This is a quadratic function that is designed in such a way that it is difficult for all methods to minimise it. For details on its construction, see Section 6.1.

The minimisation was performed for two sets of parameters for the function. We observe that while a good choice of a cycle length can get close to FSI that is using the exact value of μ, the problem is that for different parameters of the function, different cycle lengths are optimal. While we can attempt to derive an optimal cycle length from the condition number it would not be exciting since if we know both the condition number and the Lipschitz constant we can simply use FSI that has been adapted to the strong convexity and a single cycle. The goal of this paper is to determine a good cycle length in an automatic way without any additional information about the function.

3 Related Work

A classic gradient-based method is Polyak's heavy ball method [13] which has the following updating rule:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha \boldsymbol{\nabla} F(\boldsymbol{x}^k) + \beta(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})\,, \tag{5}$$



(a) $L = 10^2$, $\mu = 10^{-3}$ (b) $L = 10$, $\mu = 10^{-2}$

Fig. 1. Convergence plots of FSI with different cycle lengths for solving Nesterov's worst case problem. The problem dimension is $N = 10^5$ and we explore two difference parameter choices for L and μ. The convergence is measured in terms of the squared Euclidean norm of the gradient of the objective. FSI that has been optimally adapted to the strong convexity is used as a baseline. The optimal cycle length for FSI can change significantly depending on the problem parameters.

where the inertial parameter $\beta \in [0,\, 1)$ controls the momentum we gain and $\alpha \in (0,\, \frac{2(1+\beta)}{L})$ is the step size. As shown in [6], FSI schemes for convex optimisation can be viewed as a variant of Polyak's heavy ball method by allowing cyclically varying parameters; see (3).

A closely related method is Nesterov's accelerated gradient descent [8]. The accelerated gradient descent has the following updating rule:

$$\begin{aligned} \boldsymbol{y}^k &= \boldsymbol{x}^k + \beta_k(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})\,, \\ \boldsymbol{x}^{k+1} &= \boldsymbol{y}^k - \frac{1}{L}\nabla F(\boldsymbol{y}^k)\,, \\ \beta_k &= \theta_k(\theta_{k-1}^{-1} - 1)\,, \end{aligned} \tag{6}$$

where $\theta_0 := \theta_{-1} := 1$ and $\frac{1-\theta_{k+1}}{\theta_{k+1}^2} \leq \frac{1}{\theta_k^2}$. Here the extrapolated point is also used in the evaluation of the gradient. This allows the accelerated gradient descent to respond to an increase in the function values earlier, reducing its oscillatory behaviour.

A common choice for the inertial parameter is $\beta_k = \frac{k-1}{k+2}$. Comparing it to the inertial parameter of FSI, $(\alpha_k - 1) = (k - \frac{1}{2})/(k + \frac{3}{2})$, we observe that β_k

converges to 1 at a slower rate than $(\alpha_k - 1)$. If we fix the cycle length of FSI to some K then $(\alpha_k - 1)$ will be bounded from above by the constant $(\alpha_K - 1) < 1$ and for a large enough k, β_k will overtake it.

Like FSI, the accelerated gradient descent can be adapted to the strong convexity of a function with the parameter choice [9, Section 2.2.1]:

$$\beta_k = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}\,. \tag{7}$$

A closer relative to FSI is the restarted accelerated gradient descent introduced by O'Donoghue and Candès [12]. Just like FSI the restarted accelerated gradient descent also cyclically varies its parameters and resets its momentum. O'Donoghue and Candès consider both a fixed restart interval, and two different adaptive schemes:

– *Function scheme*: Restart whenever

$$F(\boldsymbol{x}^k) > F(\boldsymbol{x}^{k-1})\,. \tag{8}$$

– *Gradient scheme*: Restart whenever

$$\left\langle \boldsymbol{\nabla} F(\boldsymbol{y}^{k-1}), \boldsymbol{x}^k - \boldsymbol{x}^{k-1} \right\rangle > 0\,. \tag{9}$$

While the function scheme offers monotonicity it can be numerically unstable. The gradient scheme works better in practice and is often cheaper to compute.

Another restarting scheme for the accelerated gradient descent was recently proposed by Su, Boyd and Candès [15]. The scheme restarts when it detects decreasing speed. This can be detected with the following restart criterion:

– *Speed based scheme*: Restart whenever

$$||\boldsymbol{x}^k - \boldsymbol{x}^{k-1}|| < ||\boldsymbol{x}^{k-1} - \boldsymbol{x}^{k-2}||\,. \tag{10}$$

Although this scheme often performs worse than the gradient scheme it comes with the advantage of a linear worst case convergence rate.

While the heavy ball method, the restarted accelerated gradient descent and obviously FSI are the closest to our work, there exist many extensions of the heavy ball method and the accelerated gradient descent. For example, FISTA [3] extends the accelerated gradient descent to include non-smooth functions, and iPiano [11] extends the heavy ball method to include non-smooth and non-convex functions. These methods are all closely related to the momentum method that is frequently applied in machine learning [14, 16].

FED [5] uses another acceleration strategy instead of accumulating momentum. FED uses step sizes that on their own are unstable but can be combined into a stable cycle. In the linear setting this is equivalent to FSI [6].

4 Adaptive Restarting

In Section 2 we saw that in general selecting a good cycle length for the FSI solver is difficult. For specific tasks we might be able to find cycle lengths that work well in practice but this requires some extra work from the user. Instead of the user needing to adapt FSI to the problem, FSI should adapt in an automatic way.

We seek a simple criterion that is both cheap to compute, and yields a performance that is comparable to an expert selecting the cycle length manually for a specific task.

One way to achieve this is using adaptive restarts [12]. The idea comes from the observation that the function values of momentum based methods start to oscillate if the inertia parameter is set higher than the optimal value.

To visualise this we consider the heavy ball method in 2D. The heavy ball method can be thought of as a ball rolling down some landscape. If our landscape is a bowl then it is easy to see that if the friction between the bowl and and ball is low the ball will roll past the minimum and up the other side of the bowl. If we want to get to the minimum fast it is a natural idea to simply stop the ball whenever it starts going upwards.

Therefore, the idea of adaptive restarting is to discard the momentum whenever the function values start increasing. Since the gradient is always pointing upwards in the function, we can go into a new cycle when the following condition is met:

$$\left\langle \boldsymbol{\nabla} F\left(\boldsymbol{x}^k\right), \boldsymbol{x}^k - \boldsymbol{x}^{k-1} \right\rangle > 0\,. \tag{11}$$

This leads to *Adaptive FSI* (*AFSI*) schemes for strongly convex optimisation.

Since we discard the entire momentum once it is pointing towards higher function values, the advantage of the accelerated gradient descent discussed in Section 3 becomes less obvious. With the oscillatory behaviour taken care of explicitly, the faster growing inertial parameter and larger step sizes of FSI become more attractive.

Algorithm 1 shows the general idea of AFSI. In each iteration we need to compute one additional inner product. When compared to the rest of the iteration this is not expensive.

Let us compare our restart strategy from (11) to the function scheme from (8) and the gradient scheme from (9). While we designed it to prevent an increase in function values like the function scheme, it shares the structure with the gradient scheme. The difference is where we evaluate the gradient. When deciding whether the momentum at $\boldsymbol{x}^k$ should be reset or used for the extrapolation step from (6), the gradient scheme uses the gradient information from the previous intermediary point $\boldsymbol{y}^{k-1}$. Our scheme uses the gradient information from the current iterate $\boldsymbol{x}^k$, this is illustrated in Fig. 2. While our scheme requires an extra gradient evaluation when applied to Nesterov's accelerated gradient descent, FSI requires the gradient at $\boldsymbol{x}^k$ anyway. Therefore no extra gradient evaluations are required when implementing AFSI. In the next section we will see that evaluating the restarting condition at the current iterate gives us monotonicity of the

Algorithm 1 AFSI

$\omega \in \left(0, \frac{2}{L}\right)$, $\boldsymbol{x}^{-1} := \boldsymbol{x}^0$, $k := 0$
while stopping criterion is not met **do**
  if $\langle \nabla f(\boldsymbol{x}^k), \boldsymbol{x}^k - \boldsymbol{x}^{k-1} \rangle > 0$ {Check the reset condition} **then**
    $\boldsymbol{x}^{-1}, \boldsymbol{x}^0 \leftarrow \boldsymbol{x}^{k-1}$ { Reset the momentum}
    $k \leftarrow 0$
  end if
  $\alpha_k \leftarrow \frac{4k+2}{2k+3}$
  $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k - \alpha_k \omega \nabla f(\boldsymbol{x}^k) + (\alpha_k - 1)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$
  $k \leftarrow k + 1$
end while

function values. This combines the theoretical properties of the function scheme with the numerical stability and cheap computation of the gradient scheme.

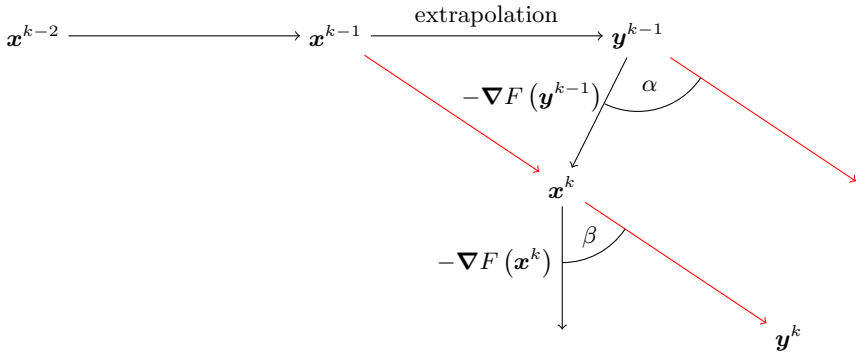


Fig. 2. This figure illustrates the difference between the gradient scheme from (9) and our scheme. When deciding whether the momentum at $\boldsymbol{x}^k$ (in red) should be reset or used to compute $\boldsymbol{y}^k$, the schemes use gradient information from different locations. The gradient scheme resets when the angle α is obtuse and our scheme resets when β is obtuse.

5 Theoretical Insights

While resetting the momentum term and going into a new cycle when it is pointing towards higher function values sounds intuitive, it can also be motivated directly by the convergence analysis of FSI. If we know that the algorithm goes into a new cycle whenever the inequality from (11) is satisfied we have the additional information that all previous iterates satisfy

$$\left\langle \boldsymbol{\nabla} F\left(\boldsymbol{x}^k\right), \boldsymbol{x}^{k-1} - \boldsymbol{x}^k \right\rangle \geq 0\,. \tag{12}$$

This term appears in a lot of useful inequalities from convex analysis and knowing its sign proves to be very useful. The fact that it allows us to ignore a lot of terms in the convergence analysis of FSI can be considered a motivation for the restart condition from (11). We can use the subgradient inequality (see for example [9, Section 2.1.1])

$$F\left(\boldsymbol{y}\right) \geq F\left(\boldsymbol{x}\right) + \left\langle \boldsymbol{\nabla} F\left(\boldsymbol{x}\right), \boldsymbol{y} - \boldsymbol{x} \right\rangle \tag{13}$$

to conclude that

$$F\left(\boldsymbol{x}^k\right) \leq F\left(\boldsymbol{x}^{k-1}\right) . \tag{14}$$

This tells us that AFSI is a descent method. Note that we are not able to conclude this with the gradient scheme from (9).

By applying (12) repeatedly in the convergence analysis we can derive the following linear convergence rate for AFSI. The proof can be found in the preprint version of this paper.

Theorem 1. *Let $F : \mathbb{R}^N \to \mathbb{R}$ be a smooth strongly convex function with convexity parameter μ and a L-Lipschitz continuous gradient with $L > \mu$. Furthermore, let F^* be the unique minimum of F. Let $(\boldsymbol{x}^k)_{k\in\mathbb{N}}$ be generated by a single cycle of Algorithm 1 with initialisation $\boldsymbol{x}^0 \in \mathbb{R}^N$, and step size $\omega \in \left(0, \frac{1}{L}\right)$. Then AFSI has the following convergence rate:*

$$F(\boldsymbol{x}^k) - F^* \leq q^k C\left(\frac{L}{\mu}\right)\left(F(\boldsymbol{x}^0) - F^*\right) , \tag{15}$$

where the constant $C\left(\frac{L}{\mu}\right)$ depends on the condition number L/μ and

$$q := \sqrt{\frac{\frac{L}{\mu}}{2\mu\omega\left(1 - L\omega\right) + \frac{L}{\mu}}} . \tag{16}$$

The convergence rate from Theorem 1 is optimised for $\omega = \frac{1}{2L}$, where we have

$$q = \sqrt{\frac{1}{\frac{1}{2}\left(\frac{\mu}{L}\right)^2 + 1}} . \tag{17}$$

Fig. 3 shows how the convergence rate compares to the convergence rate of gradient descent, and the optimal convergence rate in the sense of Nemirovski and Yudin [7, 9]. While the convergence rate provided by theorem 1 is worse than the convergence rate of gradient descent, it is linear. This is an improvement over FSI. Section 6 shows that in practice we observe much faster convergence. The constant C is in general not very large and for $L/\mu \geq 25$ we have $C(L/\mu) = 1$.

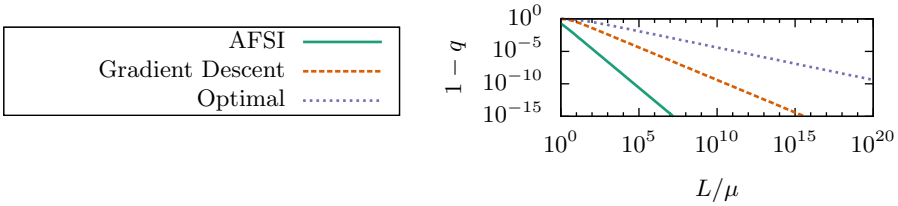


Fig. 3. Plot of 1 minus the convergence rate as a function of the condition number L/μ for AFSI, gradient descent, and the optimal convergence rate for the class of strongly convex smooth functions. While the worst case convergence rate of AFSI is worse than gradient descent, it is linear.

6 Experiments

6.1 Nesterov's Worst Case Functions

Nesterov's worst case functions are a family of functions that are designed to be difficult to minimise for all methods. They are defined by [9, Section 2.1.4]:

$$F_{\mu,L}(\boldsymbol{x}) = \frac{L-\mu}{8}\left(x_1^2 + \sum_{i=1}^{+\infty}(x_i - x_{i+1})^2 - 2x_1\right) + \frac{\mu}{2}||\boldsymbol{x}||_2^2 \tag{18}$$

for strongly convex functions, and [9, Section 2.1.2]

$$F_{k,L}(\boldsymbol{x}) = \frac{L}{4}\left(\frac{1}{2}\left(x_1^2 + \sum_{i=1}^{2k}(x_i - x_{i+1})^2 + x_{2k+2}^2\right) - x_1\right) \tag{19}$$

for convex functions. The difficulty arises when we initialise with $\boldsymbol{x}^0 := \mathbf{0}$, then at iteration k, $\boldsymbol{x}^k$ will at most have k nonzero entries. The nonzero entries of the minimum $\boldsymbol{x}^*$ then provide a bound on the convergence rate. For the class of strongly convex functions the worst case function is defined for $\mathbb{R}^{+\infty} \to \mathbb{R}$. Therefore, when approximating it the problem dimension should be large compared to the number of iterations taken.

For the convex problem the function has a parameter $k \leq (N-1)/2$. This parameter governs at which iteration the following lower bound holds:

$$F_{k,L}(\boldsymbol{x}^k) - F^* \geq \frac{3L||\boldsymbol{x}^0 - \boldsymbol{x}^*||^2}{32(k+1)^2}. \tag{20}$$

While this proves that for all iterations k there exists a function such that the bound holds, it is important to note that for a specific instance of the function it is only guaranteed to hold at a single iteration k. Since the function is not designed to be difficult to minimise at any other iteration we can argue that the performance after iteration k is not interesting.

Since we are using adaptive restarting the obvious method to compare AFSI to is Nesterov's accelerated gradient descent with adaptive restarting that was introduced by O'Donoghue and Candes [12]. We consider 2 restarting schemes, the gradient scheme from (9), and the speed based scheme of Su et al. from (10).

In Fig. 4 we observe that AFSI can get close to FSI adapted to the strong convexity with the optimal parameters and can even beat the accelerated gradient descent adapted to the convexity. For this problem the gradient scheme for the accelerated gradient descent has not yet reset its momentum. Therefore, it is still identical to the accelerated gradient descent from (6). The speed based scheme performs better but it cannot achieve the performance of AFSI.

While our analysis was only conducted for strongly convex functions we also evaluate the performance on Nesterov's worst case convex function with $k = 50$. We observe that we keep the state-of-the-art performance of FSI in the interesting part of the function. Once we get past it, AFSI becomes superior. Here the gradient scheme for the accelerated gradient descent works as intended by restarting once the function values start to increase. At this point AFSI is already in its third cycle.

We observe that the speed based scheme performs worse than the gradient scheme due to restarting too early. Since the convergence rate of the speed based scheme is linear it will overtake the accelerated gradient descent without restarting. This happens after around 1000 iterations.

6.2 Non-quadratic Minimisation

For this experiment we consider the following functional:

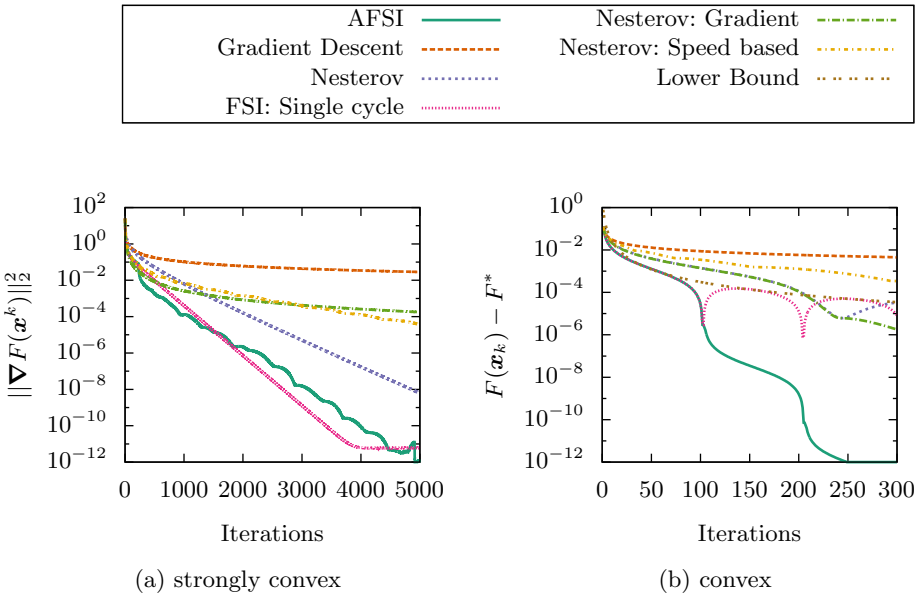
$$E(u) = \frac{1}{2} \int_{\Omega} \left(\Psi((u-f)^2) + \alpha \Psi(|\boldsymbol{\nabla} u|^2) \right) dxdy \tag{21}$$

and its discretised counterpart $F : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}$. Here we aim to remove noise from an image $f : \Omega \to \mathbb{R}$ by finding an image $u : \Omega \to \mathbb{R}$ that is both similar to f and is smooth. To achieve this we use the Charbonnier penaliser [4]:

$$\Psi(s^2) = 2\lambda^2 \sqrt{1 + \frac{s^2}{\lambda^2}} - 2\lambda^2 \,. \tag{22}$$

It is worth noting that as λ goes to 0 our functional approaches TV-L1 regularisation [10].

What makes this problem interesting is that both the similarity term and the smoothness term are subquadratic. Therefore, we have no quadratic lower bound on F, and consequently it is not globally strongly convex. While we cannot find a μ that works globally, we can always find a $\mu > 0$ if we restrict ourselves to a level set of F. In practice getting a better bound than $\mu > 0$ for a given level set is difficult. Since in (14) we showed that AFSI is a descent method we will not leave the initial level set. Therefore, AFSI sees F as being effectively strongly convex even if globally it is not. In other words, AFSI can adapt to the local structure of F while FSI uses the global information.



(a) strongly convex (b) convex

Fig. 4. Convergence plots of AFSI, FSI that has been optimally adapted to the strong convexity, Nesterov's accelerated gradient descent that has been optimally adapted to the strong convexity, and Nesterov's accelerated gradient descent with the gradient based restarting scheme from (9) and the speed based restarting scheme from (10). The methods are solving Nesterov's worst case problem. The strongly convex problem in (a) has the dimension 10^5, and parameters $L = 10^2$ and $\mu = 10^{-3}$. The convergence is measured in terms of the squared Euclidean norm of the gradient. The convex problem in (b) has the dimension 10^3, and parameters $k = 50$ and $L = 1$. The residual of the objective is used to evaluate the convergence. AFSI achieves state-of-the-art performance and even outperforms Nesterov's optimal method for both functions.

In Fig. 5 we observe that this does indeed give AFSI and the restarted accelerated gradient descent an advantage over their counterparts without restarts. AFSI even performs better than FSI that has the cycle length tuned by hand to 425 for the fastest convergence. While this cycle length results in a comparable convergence rate once we are close to the minimum it is not well suited at the beginning of the process. In contrast the adaptive cycle length of AFSI performs well at all stages of the optimisation.

Comparing the gradient scheme and the speed based scheme for restarting the accelerated gradient descent, we again observe that the speed based scheme is restarting too frequently. While the gradient scheme converges faster than both the speed based scheme and the accelerated gradient descent without any restarting, it converges slower than AFSI.

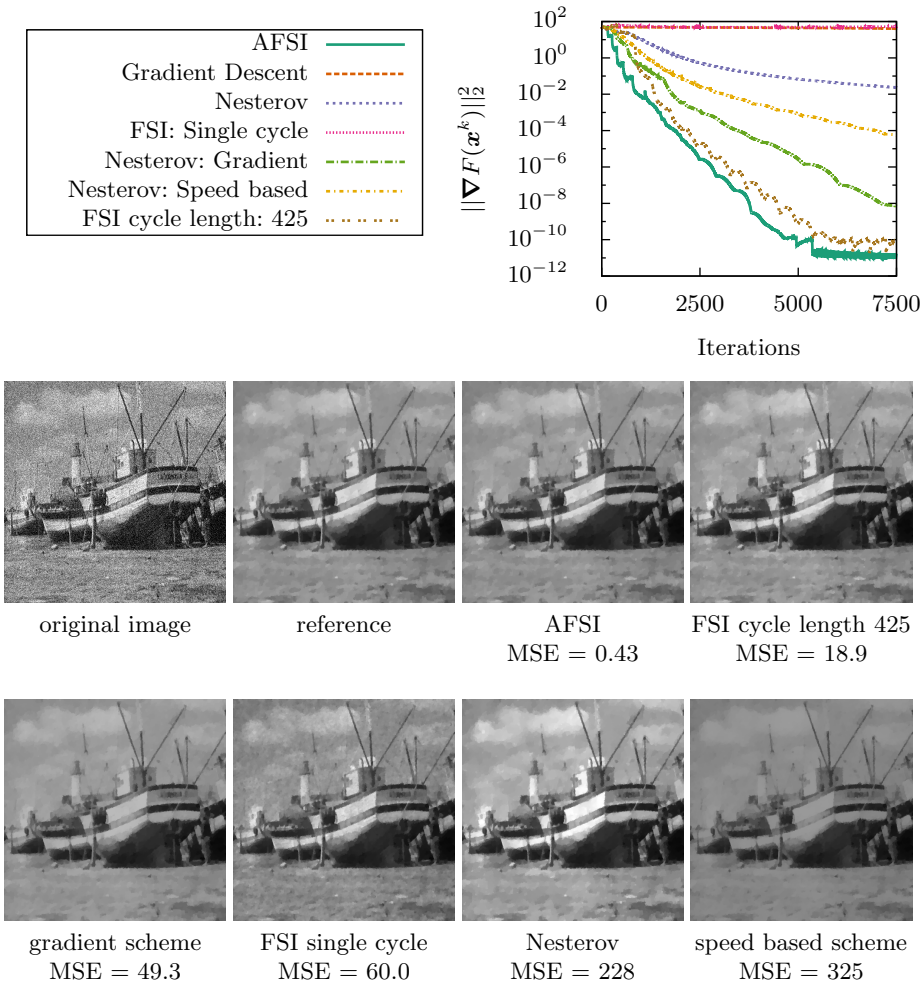


Fig. 5. Denoising with $\alpha = 1$, $\lambda = 0.1$, and a black initialisation. Comparison of AFSI, FSI, and Nesterov's accelerated gradient descent with and without adaptive restarting. The solutions and mean square error (MSE) of the methods after 500 iterations are shown above. The solution of the gradient descent method (not shown) is still the black initialisation after 500 iterations. The reference solution is obtained with 20000 iterations of Nesterov's method restarted with the gradient scheme. The cycle length of 425 was hand-tuned for the best performance. AFSI reaches a low MSE faster than the other methods, and converges faster than FSI for all fixed cycle lengths.

We observe that while a single FSI cycle converges slowly it achieves a good approximation quickly. This is exactly what allows FSI to perform so well once we apply adaptive restarting. For a restarted method we only care about how fast it can reach the minimum and get into the next cycle, what happens after that is not important.

7 Conclusions and Future Work

We have introduced *Adaptive FSI* (AFSI) schemes for strongly convex optimisation problems. They provide an automatic way of selecting the cycle length for FSI schemes. Since we no longer have an extra parameter, we have a method that is both simple to implement and simple to use. We can show that AFSI offers additional stability guarantees over FSI where the cycle length is a free parameter.

Our experiments demonstrate that when the strong convexity parameter of the function is known we can get close to the performance of the optimal methods. Additionally, when no useful bound on the strong convexity parameter is available AFSI can outperform them.

While we have only considered FSI schemes for unconstrained optimisation, FSI schemes can also be used for solving parabolic and elliptic partial differential equations, and for constrained optimisation [6]. In our ongoing work, we are studying how to extend the results for AFSI to these other types of FSI schemes.

Acknowledgements. Our research has been partially funded by the Cluster of Excellence on Multimodal Computing and Interaction within the Excellence Initiative of the German Research Foundation (DFG) and by the ERC Advanced Grant INCOVID. This is gratefully acknowledged.

References

1. Attouch, H., Peypouquet, J., Redont, P.: Fast convergence of an inertial gradient-like system with vanishing viscosity (2016)
2. Bähr, M., Dachsel, R., Breuß, M.: Fast solvers for solving shape matching by time integration. Annual Workshop of the AAPR (42), 65–72 (May 2018)
3. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences **2**(1), 183–202 (2009)
4. Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Deterministic edge-preserving regularization in computed imaging. IEEE Transactions on Image Processing **6**(2), 298–311 (Feb 1997)
5. Grewenig, S., Weickert, J., Bruhn, A.: From box filtering to fast explicit diffusion. In: Goesele, M., Roth, S., Kuijper, A., Schiele, B., Schindler, K. (eds.) Pattern Recognition, DAGM 2010, LNCS, vol. 6376, pp. 533–542. Springer, Berlin, Heidelberg (2010)
6. Hafner, D., Ochs, P., Weickert, J., Reißel, M., Grewenig, S.: FSI schemes: Fast semi-iterative solvers for PDEs and optimisation methods. In: Andres, B., Rosenhahn, B. (eds.) German Conference on Pattern Recognition, LNCS, vol. 9796, pp. 91 – 102. Springer, Cham (2016)

7. Nemirovski, A., Yudin, D.: Problem complexity and method efficiency in optimization. Wiley-Interscience series in discrete mathematics (1983)
8. Nesterov, Y.: A method of solving a convex programming problem with convergence rate $O(1/k^2)$. Soviet Mathematics Doklady **27**, 372–376 (1983)
9. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Springer (2004)
10. Nikolova, M.: A variational approach to remove outliers and impulse noise. Journal of Mathematical Imaging and Vision **20**(1), 99–120 (Jan 2004)
11. Ochs, P., Chen, Y., Brox, T., Pock, T.: iPiano: Inertial proximal algorithm for non-convex optimization. SIAM Journal on Imaging Sciences (SIIMS) **7**, 1388 – 1419 (2014)
12. O'Donoghue, B., Candès, E.: Adaptive restart for accelerated gradient schemes. Foundations of Computational Mathematics **15**(3), 715–732 (Jun 2015)
13. Polyak, B.: Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics **4**, 1–17 (12 1964)
14. Rumelhart, D., Hinton, G., Williams, R.: Learning internal representations by error propagation. In: Rumelhart, D., McClelland, J. (eds.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, chap. 8, pp. 318–362. MIT Press, Cambridge, MA (1986)
15. Su, W., Boyd, S., Candès, E.: A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. Journal of Machine Learning Research **17**, 1–43 (2016)
16. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: Dasgupta, S., McAllester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 28, pp. 1139–1147. PMLR, Atlanta, Georgia, USA (17–19 Jun 2013)