Why do we regularise in every iteration for imaging inverse problems?

Evangelos Papoutsellis[*1], Zeljko Kereta[2], and Kostas Papafitsoros[3]

[1]Finden Ltd, [2]University College London, [3]Queen Mary University of London

Abstract

Regularisation is a common method in iterative solutions for imaging inverse problems. The majority of algorithms evaluate the proximal operator of the regulariser in every iteration, leading to a significant computational overhead, as such evaluations can be costly. In this context, we investigate skipping the regulariser to reduce the frequency of proximal operator computations. This approach shows a reduction in computational time without compromising convergence or image quality. Here we study for the first time the efficacy of regularisation skipping on a variety of imaging inverse problems. We build upon the ProxSkip algorithm and we also propose a novel skip-version of the PDHG algorithm. Extensive numerical results highlight the potential of these methods to accelerate computations while maintaining high-quality reconstructions.

Keywords— Inverse Problems Iterative Regularisation Proximal Operator Stochastic Optimisation

1 Introduction

Inverse problems involve the process of estimating an unknown quantity $\boldsymbol{u}^\dagger \in \mathbb{X}$ from indirect and often noisy measurements $\boldsymbol{b} \in \mathbb{Y}$ obeying $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{u}^\dagger + \boldsymbol{\eta}$. Here $\mathbb{X}, \mathbb{Y}$ denote finite Euclidean dimensional spaces, $\boldsymbol{A} : \mathbb{X} \to \mathbb{Y}$ is a linear forward operator, $\boldsymbol{u}^\dagger$ is the ground truth and $\boldsymbol{\eta}$ is a random noise component. Given $\boldsymbol{b}$ and $\boldsymbol{A}$, the goal is to approximate $\boldsymbol{u}^\dagger$. Since inverse problems are typically ill-conditioned, prior information about $\boldsymbol{u}$ is often incorporated in the form of regularisation. The inverse problem solution is then approximated by solving

$$\arg\min_{\boldsymbol{u}\in\mathbb{X}} \mathcal{D}(\boldsymbol{A}\boldsymbol{u}, \boldsymbol{b}) + \alpha\mathcal{R}(\boldsymbol{u}). \tag{1}$$

Here $\mathcal{D}$ denotes the fidelity term, measuring the distance between $\boldsymbol{b}$ and the solution $\boldsymbol{u}$ under $\boldsymbol{A}$. Regularisation term $\mathcal{R}$ promotes properties such as smoothness, sparsity, edge preservation, and low-rankness of the solution, and is weighted by a parameter $\alpha > 0$. Established examples for $\mathcal{R}$ in imaging include Total Variation (TV), the Total Generalised Variation [8], the Total Nuclear Variation (TNV) [17], and more general tensor-based structure regularisation [20].

*corresponding author: epapoutsellis@gmail.com

Algorithm 1 GD

Parameters: $\gamma > 0$
Initialize: $\boldsymbol{x}_0 \in \mathbb{X}$
for $k = 0, \dots, K-1$ **do**
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \gamma \nabla f(\boldsymbol{x}_k)$
end for

Algorithm 2 ISTA/PGD/FBS

Parameters: $\gamma > 0$
Initialize: $\boldsymbol{x}_0 \in \mathbb{X}$
for $k = 0, \dots, K-1$ **do**
    $\boldsymbol{x}_{k+1} = \text{prox}_{\gamma g}(\boldsymbol{x}_k - \gamma \nabla f(\boldsymbol{x}_k))$
end for

Algorithm 3 FISTA

Parameters: $\gamma > 0$, $t_0 = 1$
Initialize: $\boldsymbol{x}_0 \in \mathbb{X}$
for $k = 0, \dots, K-1$ **do**
    $\boldsymbol{x}_{k+1} = \text{prox}_{\gamma g}(\boldsymbol{y}_{k+1} - \gamma \nabla f(\boldsymbol{y}_{k+1}))$
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$, $a_k = \frac{t_k - 1}{t_{k+1}}$
    $\boldsymbol{y}_{k+1} = \boldsymbol{x}_{k+1} + a_k (\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)$
end for

Algorithm 4 ProxSkip

Parameters: $\gamma > 0$, probability $p > 0$
Initialize: $\boldsymbol{x}_0, \boldsymbol{h}_0 \in \mathbb{X}$
for $k = 0, 1, \dots, K-1$ **do**
    $\hat{\boldsymbol{x}}_{k+1} = \boldsymbol{x}_k - \gamma(\nabla f(\boldsymbol{x}_k) - \boldsymbol{h}_k)$
    Sample $\theta_k \sim \text{Bernoulli}(p)$, $\theta_k \in \{0, 1\}$
    if $\theta_k = 1$ **then**
        $\boldsymbol{x}_{k+1} = \text{prox}_{\frac{\gamma}{p} g}\left(\hat{\boldsymbol{x}}_{k+1} - \frac{\gamma}{p}\boldsymbol{h}_k\right)$
    else $\boldsymbol{x}_{k+1} = \hat{\boldsymbol{x}}_{k+1}$
    end if
    $\boldsymbol{h}_{k+1} = \boldsymbol{h}_k + \frac{p}{\gamma}(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1})$
end for

In order to obtain a solution for (1), one employs iterative algorithms such as Gradient Descent (GD) for smooth objectives or Forward-Backward Splitting (FBS) [13] for non-smooth ones. Moreover, under the general framework

$$\min_{\boldsymbol{x} \in \mathbb{X}} f(\boldsymbol{x}) + g(\boldsymbol{x}), \tag{2}$$

the Proximal Gradient Descent (PGD) algorithm, also known as Iterative Shrinkage Thresholding Algorithm (ISTA), and its accelerated version FISTA [6, 25] are commonly used when f is convex and L-smooth, and g is proper and convex. Saddle-point methods such as the Primal Dual Hybrid Gradient (PDHG) [11] are commonly used for non-smooth f.

A common property of most of these methods, see Algorithms 2–3, is the evaluation of proximal operators for the regulariser in every iteration, which for $\tau > 0$ is defined as

$$\text{prox}_{\tau g}(\boldsymbol{x}) := \underset{\boldsymbol{z} \in \mathbb{X}}{\arg\min} \left\{ \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{x}\|_2^2 + \tau g(\boldsymbol{z}) \right\}. \tag{3}$$

The proximal operator either has a closed form solution, e.g., when $\mathcal{R}(\boldsymbol{u}) = \|\boldsymbol{u}\|_1$, or requires an iterative solver, e.g., when $\mathcal{R}(\boldsymbol{u}) = \text{TV}(\boldsymbol{u})$. In the latter case, solving (3) can incur a significant computational cost for large data, particularly in medical imaging and material science applications.

The ProxSkip algorithm

Skipping the computation of the proximal operator in certain iterations can accelerate an algorithm without impacting its convergence, as noted in [24]. There, the ProxSkip algorithm was developed to handle federated learning tasks involving costly proximal operator computations. ProxSkip introduces a control variate $\boldsymbol{h}_k$ and a probability parameter p, see Algorithm 4. When the proximal

step is not applied, the control variate is unchanged. For successive iterations where the proximal operator is not applied, the computational cost is not increased. However, this leads to error accumulation. Applying the proximal operator and updating the control variate in certain iterations reduces the error.

In [24] it was shown that ProxSkip converges provided that f in (2) is L-smooth and μ-strongly convex, and the probability p satisfies

$$p \geq \sqrt{\mu/L}. \tag{4}$$

In the case of equality in (4), the algorithm converges (in expectation) at a linear rate with $\gamma = \frac{1}{L}$ and the iteration complexity is $\mathcal{O}(\frac{L}{\mu}\log(\frac{1}{\varepsilon}))$. In addition, the total number of proximal evaluations (in expectation) is only $\mathcal{O}(\frac{1}{\sqrt{p}}\log(\frac{1}{\varepsilon}))$.

Our contribution

Our aim is conduct the first numerical study, via extensive experiments, of the computational benefits of ProxSkip for various imaging inverse problems, including challenging real-world tomographic applications. In particular, we show that ProxSkip can outperform even FISTA. Moreover, we introduce PDHGSkip, a novel skip version of PDHG, see Algorithm 6, which we motivate via numerical experiments. We anticipate that this will spark further research around developing skip-versions of a variety of proximal-based algorithms used nowadays.

For all our imaging experiments we consider a quadratic distance term as data fidelity and as a regulariser we use the (isotropic) total variation

$$\mathrm{TV}(\boldsymbol{u}) = \|\boldsymbol{D}\boldsymbol{u}\|_{2,1} = \sum \sqrt{(\boldsymbol{D}_y\boldsymbol{u})^2 + (\boldsymbol{D}_x\boldsymbol{u})^2},$$

where $\boldsymbol{D} = (\boldsymbol{D}_y, \boldsymbol{D}_x)$ is the discrete forward finite differences operator. Problem (1) is thus written as

$$\arg\min_{\boldsymbol{u}\in\mathbb{X}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{b}\|_2^2 + \alpha\mathrm{TV}(\boldsymbol{u}). \tag{5}$$

2 Skipping light proximals in imaging problems

2.1 Dual TV denoising

To showcase the algorithmic properties we consider a toy example, with the dual formulation of the classical TV denoising (ROF) which reads

$$\min_{\|\boldsymbol{q}\|_{2,\infty}\leq\alpha} \left\{ \mathcal{F}(\boldsymbol{q}) := \frac{1}{2}\|\mathbf{div}\boldsymbol{q} + \boldsymbol{b}\|_2^2 + \frac{1}{2}\|\boldsymbol{b}\|_2^2 \right\} \qquad \text{(Dual-ROF)}. \tag{6}$$

Here $\mathbf{div}$ is the discrete divergence operator such that $\mathbf{div} = -\boldsymbol{D}^T$. The solutions $\boldsymbol{u}^*$ and $\boldsymbol{q}^*$ of the primal and dual ROF problems respectively are linked via $\boldsymbol{u}^* = \boldsymbol{b} + \mathbf{div}\boldsymbol{q}^*$. A simple algorithm to solve (6) was introduced in [12], based on a Projected Gradient Descent (ProjGD) iteration $\boldsymbol{q}_{k+1} = \mathcal{P}_C(\boldsymbol{q}_k - \gamma\boldsymbol{D}(\mathbf{div}\boldsymbol{q}_k + \boldsymbol{b}))$, which is globally convergent under a fixed stepsize $\gamma \leq \frac{2}{\|\boldsymbol{D}\|^2}$ [3], with

$$\mathcal{P}_C(\boldsymbol{x}) = \frac{\boldsymbol{x}}{\max\{\alpha, \|\boldsymbol{x}\|_2\}}, \quad C = \left\{ \|\boldsymbol{q}\|_{2,\infty} \leq \alpha : \boldsymbol{q} \in \mathbb{R}^{2\times d} \right\}, \tag{7}$$

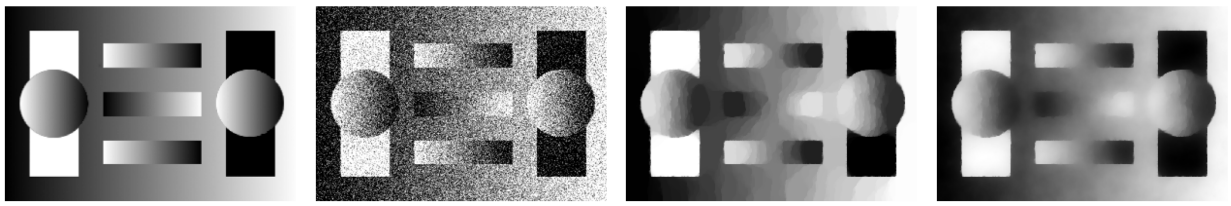

Figure 1: Left to right: Ground truth $\boldsymbol{u}^\dagger \in \mathbb{R}^{200\times 300}$. Noisy image $\boldsymbol{b}$, $\sigma = 0.05$. Dual-ROF $\boldsymbol{u}^*$ with $\alpha = 0.5$. Dual-Huber-ROF $\boldsymbol{u}^*$ (see Section 2.2) with $\alpha = 0.55$, $\hat{\varepsilon} = 0.1$. The parameters α are optimised with respect to SSIM.

and d being the image dimension. The projection $\mathcal{P}_C$ can be identified as the proximal operator of the indicator function of the feasibility set C. Thus, ProjGD is a special case of ISTA/PGD and ProxSkip can be applied. Note that due to the divergence operator this problem is not strongly convex. In fact, this is the case for the majority of the problems of the type (1), typically due to the non-injectivity of $\boldsymbol{A}$. Thus, this example also shows that the strong convexity assumption could potentially be relaxed for imaging inverse problems.

To ensure that any biases from algorithms under evaluation are avoided, the "exact" solution $\boldsymbol{u}^*$ is calculated using an independent high-precision solver, in particular, the MOSEK solver from the CVXpy library [1, 15], see Figure 1.

Both ISTA and ProxSkip use the stepsize $\gamma = \frac{1}{L} = \frac{1}{8}$, where L is the Lipschitz constant of $\mathcal{F}'(\cdot)$. For every iteration we monitor the ℓ_2 error $\|\boldsymbol{u}_k - \boldsymbol{u}^*\|_2$ between the iterate $\boldsymbol{u}_k = \boldsymbol{b} + \mathbf{div}\boldsymbol{q}_k$ and estimated exact solution $\boldsymbol{u}^*$. We use $p = [0.01, 0.1, 0.3, 0.5]$ and 50000 iterations as a stopping criterion.

In Figure 2 (top-left), we observe that the two algorithms are almost identical in terms of the ℓ_2 error. Note that ProxSkip and ISTA coincide only when $p = 1$. Indeed, one can detect some discrepancies during the first 100 iterations, which quickly dissipate with only a few applications of the projection $\mathcal{P}_C$, see bottom row of Figure 2 where we compare ISTA and ProxSkip with $p = 0.1$ for the first and last 100 iterations. In Figure 2 (top-right), we plot the ℓ_2 error with respect to CPU time (computed as the average over 30 independent runs). We observe a clearly superior performance of ProxSkip, in terms of the CPU time, for all values of p. This serves as a first demonstration of the advantages of ProxSkip in terms of computational time, without compromising image quality. In Section 3, we present a more emphatic computational impact using computationally more expensive proximal steps.

2.2 Dual TV denoising with strong convexity

In order to be consistent with the convergence theory of ProxSkip where strong convexity is a requirement, one can add a small quadratic term to the objective function. This is commonly used in imaging applications and allows the use of accelerated versions of first-order methods. For problem (6) this results in

$$\min_{\|\boldsymbol{q}\|_{2,\infty}\leq\alpha} \left\{ \mathcal{F}(\boldsymbol{q}) := \frac{1}{2}\|\mathbf{div}\boldsymbol{q} + \boldsymbol{b}\|_2^2 + \frac{1}{2}\|\boldsymbol{b}\|_2^2 + \frac{\hat{\varepsilon}}{2\alpha}\|\boldsymbol{q}\|_2^2 \right\}. \tag{8}$$

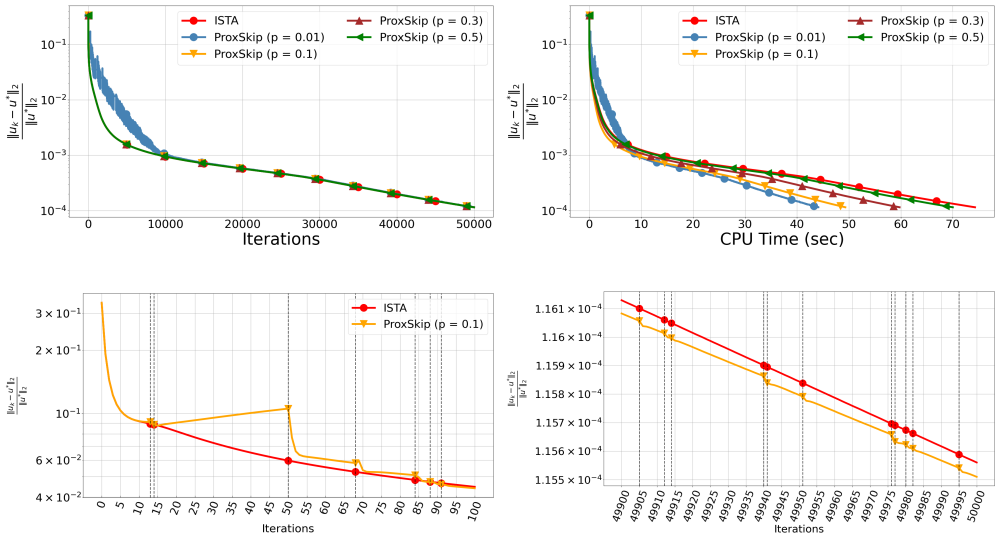

Figure 2: Top: Comparison of ISTA and ProxSkip for multiple values of p for (6) with respect to iterations (left) and CPU time (right). Bottom: Detailed versions for the first 100 (left) and the last 100 iterations (right) when $p = 0.1$. The vertical dotted lines indicate the iterations where $\mathcal{P}_C(\cdot)$ is applied.

It is known that the corresponding primal problem of (8) is the standard Huber-TV denoising [11] which involves a quadratic smoothing of the $\|\cdot\|_{2,1}$-norm around an $\hat{\varepsilon}$-neighbourhood of the origin. Among other effects, this reduces the staircasing artifacts of TV, see the rightmost image in Figure 1.

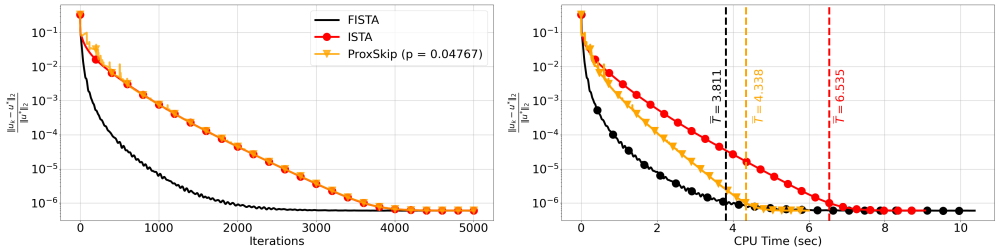

Figure 3: ISTA, FISTA and ProxSkip with optimal p value for the problem (8) with respect to iterations (left) and CPU time (right). $\bar{T}$ indicates the average time required to reach a relative error less than 10^{-6}.

We repeat the experiments of Section 2.1 with 30 independent runs of the algorithm to solve (8) with 5000 iterations and using the probability p given by the lower bound in (4). Using $(\alpha, \hat{\varepsilon}) = (0.55, 0.01)$, we have $p = \sqrt{\frac{\mu}{L}} = \sqrt{\frac{\hat{\varepsilon}}{\alpha L}} = 0.04767$. This results in an average of only 215 projection steps being used over 5000 iterations. In addition to ISTA, we also look at FISTA. For the dual problems (6) and (8) this algorithm is commonly found in the literature under alternative names such as Accelerated ProjGD or Fast Gradient Projection, [5]. In Figure 3 (left), we observe that FISTA performs better than both ISTA and ProxSkip with respect to iteration number. While a

similar behaviour is observed with respect to CPU time, see Figure 3 (right), the average time for FISTA and ProxSkip to reach a relative error less than 10^{-6} is nearly identical. A similar speed-up would be expected when FISTA is combined with skipping techniques but we leave this for future work.

3 Skipping heavy proximals in imaging problems

3.1 TV deblurring

We now consider more challenging imaging tasks, involving proximal operators that do not admit closed form solutions, and thus require computationally intensive iterative solvers, making them perfect candidates for skipping techniques.

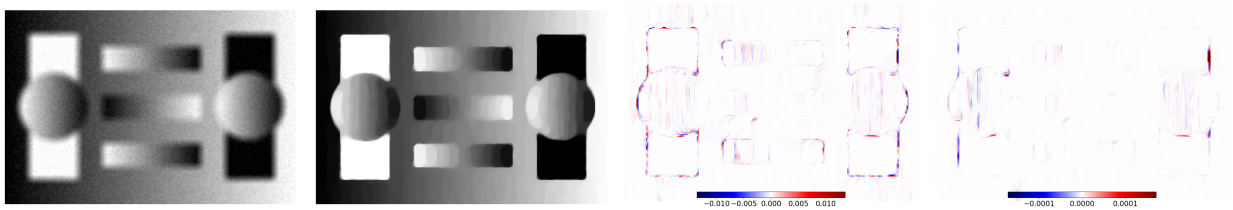

Figure 4: Left to right: Noisy and blurry image. TV deblurred image $\boldsymbol{u}^*$ with $\alpha = 0.025$. Difference error for FISTA when it is less than 10^{-3} and 10^{-5}.

We first consider the deblurring task (5), where $\boldsymbol{A}$ is a convolution operator and $\boldsymbol{b}$ a noisy and blurred image, see Figure 4. We solve (5) and compare ISTA, FISTA and ProxSkip with different values of p. Here the proximal operator of g corresponds to a TV denoising problem applied on the dual formulation (6). We employ the FISTA algorithm with a fixed number of iterations as an inner iterative solver, see next section for alternatives.

In the framework of inexact regularisation, another option is to terminate the inner solver based on some metric and predefined threshold [29]. As noted therein, the number of required inner iterations typically increases up to 10^3, as the outer algorithm progresses, leading to higher computational costs over time. To explore both the computationally easy and hard cases, we run the inner solver with 10 and 100 iterations, and use a warm-start strategy. Warm-start is vital for inexact regularisation as it avoids semi-convergence, where the error stagnates and fails to reach high precision solutions. The "exact" solution $\boldsymbol{u}^*$ is computed with 200000 iterations of PDHG with diagonal preconditioning [28]. Outer iterations are terminated if either 3000 iterations are reached or if the relative error is less than 10^{-5}. The CPU time is averaged over 10 runs.

In Figure 5, we first observe that solving the inner TV problem with 10 iterations significantly affects the convergence of FISTA. Notably, ProxSkip versions are less affected, even though besides the error introduced by the inexact solver, there is also an error from skipping the proximal. By raising the number of inner iterations to 100, and thus increasing the accuracy of the inner solver, we observe that FISTA exhibits a fast initial convergence, albeit with some oscillations. It terminates after around 500 iterations, reaching an accuracy of 10^{-5}, see Figure 5 (bottom-left). On the other hand, ISTA and ProxSkip require many more iterations to reach the same level of accuracy. However, remarkably, in this regime ProxSkip is significantly faster, in terms of CPU time, than ISTA and even outperforms FISTA when $p = 0.05$ and 0.1, see Figure 5 (bottom-right).

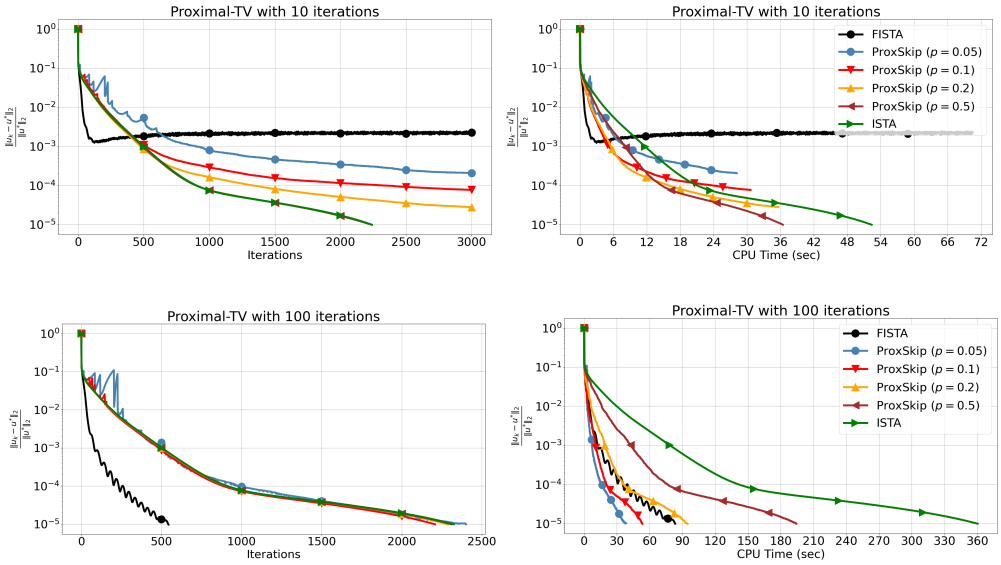

Figure 5: Comparing ISTA, FISTA and ProxSkip for multiple values of p for TV deblurring. The proximal of TV is solved using FISTA with 10 and 100 iterations applied on (6). ProxSkip outperforms FISTA when $p = 0.05$ and 0.1.

We note that there are versions of FISTA [4,10,21,26] with improved performance, also avoiding oscillations. However, our goal here is not an exhaustive comparison but rather to show that a simple version of ProxSkip outperforms a simple version of FISTA. We anticipate the future development of more sophisticated skip-based algorithms including ones based on accelerated methods.

3.2 PDHGSkip

Skipping the proximal can also be applied to primal-dual type algorithms. Here, for convex f, g and a bounded linear operator $\mathcal{K}$, the optimisation framework is

$$\min_{\boldsymbol{x}\in\mathbb{X}} f(\mathcal{K}\boldsymbol{x}) + g(\boldsymbol{x}). \tag{9}$$

In [14], a skip-version of PDHG [11] was proposed, which we denote here by PDHGSkip-1. It allows not only to skip one of the proximal steps but also the forward or backward operations of $\mathcal{K}$, depending on the order of the proximal steps, see Algorithm 5. Again, strong convexity of both f and g is required for convergence. However, our imaging experiments – in the absence of strong convexity – reveal a relatively slow performance, even with optimised step sizes σ, τ and $\omega+1 = 1/p$ according to [14, Theorem 7].

To address the slow convergence, we propose a modification: PDHGSkip-2, see Algorithm 6. The difference to PDHGSkip-1 is that the adjoint $\mathcal{K}^T$ and the proximal operator of g are now separated, see steps 4-7. Note that the control variate $\boldsymbol{h}$ vanishes when $p = 1$ and PDHG is recovered. To illustrate their practical differences, in Figure 6 we compare the two versions on tomography, see Section 3.3 for details. Apart from the clear acceleration of PDHGSkip-2 over PDHGSkip-1, we also observe a staircasing pattern for the relative error for PDGHSkip-1, see detailed zoom of the

Algorithm 5 PDHGSkip-1 [14]

Parameters: $\sigma, \tau, \omega \geq 0$, probability p
Initialize: $\boldsymbol{x}_0 \in \mathbb{X}$, $\boldsymbol{y}_0 \in \mathbb{Y}$
for $k = 0, \ldots, K-1$ **do**
  Sample $\theta_k \sim \text{Bernoulli}(p)$, $\theta_k \in \{0, 1\}$
  if $\theta_k = 1$ **then**
    $\hat{\boldsymbol{x}}_k = \frac{1}{p}(\text{prox}_{\sigma g}(\boldsymbol{x}_k - \sigma \mathcal{K}^T \boldsymbol{y}_k) - \boldsymbol{x}_k)$
  else
    $\hat{\boldsymbol{x}}_k = 0$
  end if
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \frac{1}{1+\omega}\hat{\boldsymbol{x}}_k$
  $\boldsymbol{y}_{k+1} = \text{prox}_{\tau f^*}(\boldsymbol{y}_k + \tau \mathcal{K}(\boldsymbol{x}_{k+1} + \hat{\boldsymbol{x}}_k))$
end for

Algorithm 6 PDHGSkip-2 (ours)

Parameters: $\sigma, \tau > 0$, probability p
Initialise: $\boldsymbol{x}_0, \boldsymbol{h}_0 \in \mathbb{X}$, $\boldsymbol{y}_0 \in \mathbb{Y}$
for $k = 0, \ldots, K-1$ **do**
  $\hat{\boldsymbol{x}}_{k+1} = \boldsymbol{x}_k - \sigma(\mathcal{K}^T \boldsymbol{y}_k - \boldsymbol{h}_k)$
  Sample $\theta_k \sim \text{Bernoulli}(p)$, $\theta_k \in \{0, 1\}$
  if $\theta_k = 1$ **then**
    $\boldsymbol{x}_{k+1} = \text{prox}_{\frac{\sigma}{p} g}\left(\hat{\boldsymbol{x}}_{k+1} - \frac{\sigma}{p}\boldsymbol{h}_k\right)$
  else $\boldsymbol{x}_{k+1} = \hat{\boldsymbol{x}}_k$
  end if
  $\bar{\boldsymbol{x}}_{k+1} = 2\boldsymbol{x}_{k+1} - \boldsymbol{x}_k$
  $\boldsymbol{y}_{k+1} = \text{prox}_{\tau f^*}(\boldsymbol{y}_k + \tau \mathcal{K} \bar{\boldsymbol{x}}_k)$
  $\boldsymbol{h}_{k+1} = \boldsymbol{h}_k + \frac{p}{\sigma}(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1})$
end for

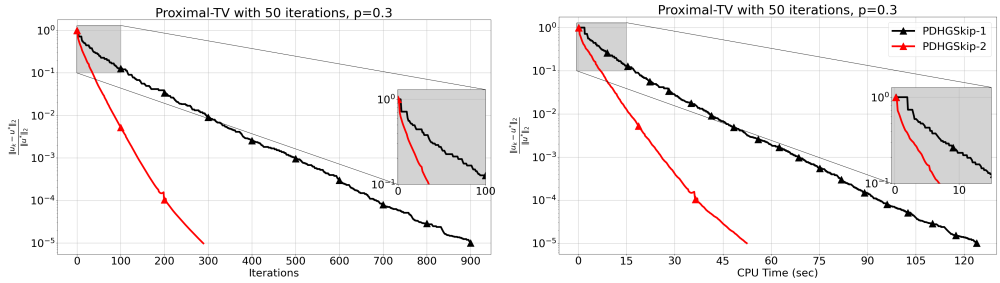

Figure 6: Comparing PDHGSkip-1 by [14] and our proposed PDHGSkip-2 for $p = 0.3$ for the tomography problem of Section 3.3, using 50 iterations for the inner solver of the proximal TV denoising problem.

first iterations. This is expected since in most iterations, where the proximal step is skipped, one variable vanishes without contributing to the next iterate. Hence, the update remains unchanged.

In general, ℓ^2-TV problems can be solved using implicit or explicit formulations of PDHG. In the implicit case, we have $f(\cdot) = \|\cdot\|_2^2$, $\mathcal{K} = \boldsymbol{A}$ and $g(\cdot) = \text{TV}(\cdot)$. In the explicit case, f is a separable sum of $\|\cdot\|_2^2$ and $\|\cdot\|_{2,1}$ composed with the block operator $\mathcal{K} = [\boldsymbol{A}; \boldsymbol{D}]$, and g can be the zero function or a non-negativity constraint. Here every proximal step has an analytical solution. This significantly reduces the per-iteration cost, but also requires more iterations to reach a desired accuracy [9]. Inexact regularisation is usually preferred and helps reduce the number of iterations. Hence, the number of calls of the forward and backward operations of $\boldsymbol{A}$ is reduced, which in certain applications gives a considerable speed-up.

3.3 TV Tomography reconstruction with real-world data

For our final case study, we solve (5) under a non-negativity constraint for $\boldsymbol{u}$ for a real-world tomographic reconstruction. Here $\boldsymbol{A}$ is the discrete Radon transform and $\boldsymbol{b}$ is a noisy sinogram of a real chemical imaging tomography dataset, representing post partial oxidation of methane reaction Ni-Pd/CeO2-ZrO2/Al2O3 catalyst [22], see Figure 7. The initial dataset was acquired for 800

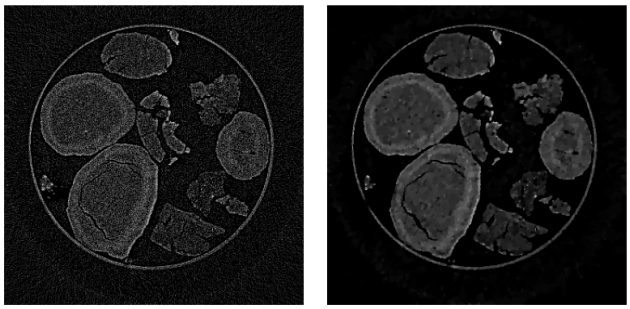

Figure 7: FBP (left) and TV (right) reconstruction using diagonal preconditioned PDHG for 200000 iterations. Regularisation parameter is manually set to balance noise reduction with feature preservation.

projection angles with 695×695 detector size and 700 vertical slices. For demonstration purposes and to be able to perform multiple runs for computing more representative CPU times, we consider one vertical slice with half the projections and 2× rebinned detector size. Same conditions for the inexact solver and stopping rules are used as in the previous section. For these optimisation problems, one can use algorithms that fit within both general frameworks (2) and (9), see [2].

The two algorithms use the same values for step sizes σ and τ, satisfying $\sigma\tau\|\mathcal{K}\|^2 \leq 1$. Comparisons in Figure 8 show a similar trend for proximal-gradient-based algorithms as in Section 3.1. When we use 10 iterations for the inner solver, FISTA fails to reach the required accuracy and stagnates, which is not the case for the ProxSkip algorithms even for the smallest p. This is accompanied by a significant CPU time speed-up, which further increases as more inner iterations are used. In fact, by increasing the number of inner iterations, the computational gain is evident with around 90% speed-up compared to FISTA.

The best performance with respect to CPU time is achieved by PDHGSkip-2. There, distance errors are identical to the case $p = 1$, in terms of iterations, except for the extreme case $p = 0.1$ which oscillates in the early iterations. For 10 inner iterations and $p \in \{0.3, 0.5\}$ we observe a slight delay in terms of the convergence rate and the desired accuracy, due to error accumulation caused by skipping the proximal operator and limited accuracy of the inner solver. This is corrected by increasing the number of inner iterations. Moreover, we see no difference with respect to CPU time for 10 inner iterations and $p \in \{0.7, 1.0\}$. This is expected since the computational cost to run 10 iterations of FISTA is relatively low. However, it demonstrates that we can have the same reconstruction using the proximal operator 70% of the time. Finally, the computational gain is more apparent when we increase the number of inner iterations, thereby increasing the computational cost of the inner solver.

We note that such expensive inner steps are used by open source imaging libraries and are solved with different algorithms and stopping rules. In CIL [18, 27], FISTA is used to solve (6) with 100 iterations. In PyHST2 [23], FISTA is used with 200 iterations and a duality gap is evaluated every 3 iterations as a stopping rule. In TIGRE [7], the PDHG algorithm with adaptive step sizes, [30], is applied to (6), with 50 iterations. In DeepInv[1], PDHG is applied to (5) (with $\boldsymbol{A} = \mathbf{Id}$) with 1000 iterations and error distance between two consecutive iterates. Finally, in Tomopy [16], PDHG is

[1]https://github.com/deepinv/deepinv

applied to (5) (with $\boldsymbol{A} = \mathbf{Id}$) and the number of iterations is specified by the user. All these default options are optimised and tested for particular real-world data cases like the ones encountered in synchroton facilities for instance. Alternatively, one can avoid inexact inner solvers for TV denoising. In [19] the proximal is replaced by a combination of wavelet and scaling transforms and is computed using a componentwise shrinkage operator. Even in this case, we expect a computational gain by skipping this operator.

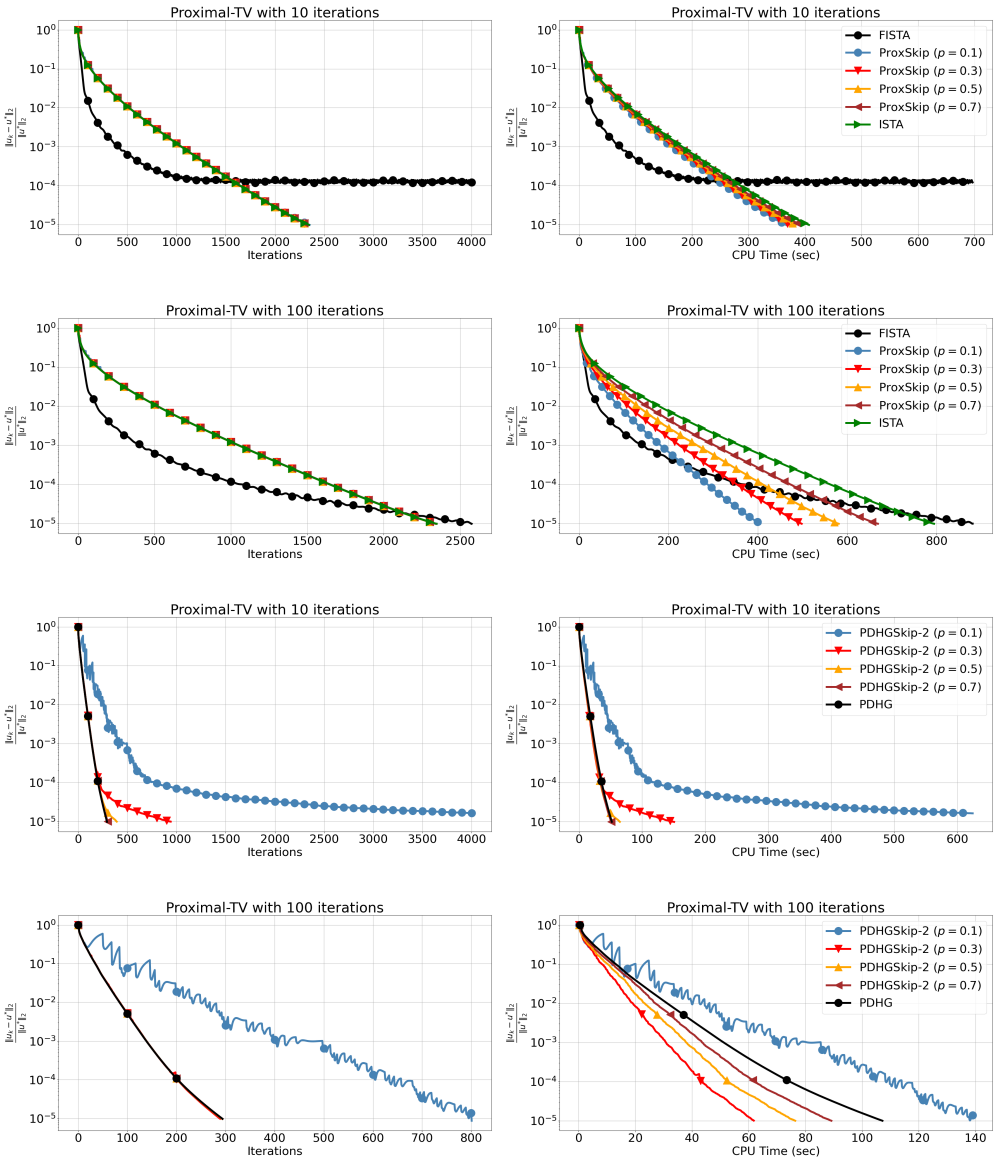

Figure 8: Comparing ISTA, FISTA, PDHG, ProxSkip and PDHGSkip-2 for multiple values of p for the TV tomography problem. The proximal of TV is solved using FISTA applied on (6) for 10 and 100 iterations.

4 Code Reproducibility

The code and datasets needed to reproduce the results are available at `https://github.com/epapoutsellis/Why-do-we-regularise-in-every-iteration`. For all the experiments we use an Apple M2 Pro, 16Gb without GPU usage to avoid measuring data transferring time between the host and the device which can be misleading.

5 Discussion and Future work

In this paper, we explored the use of the ProxSkip algorithm for imaging inverse problems. ProxSkip allows skipping costly proximal operators, usually related to the regulariser, without impacting the convergence and the solution. In addition, we presented a new skipped version of PDHG. This can be useful when the L-smoothness assumption is not satisfied, e.g., when using the Kullback-Leibler divergence. Here we focused on showcasing its advantages in practice rather than proving its convergence which we leave it for future work. Although we demonstrated that avoiding computing the proximal leads to better computational times, this speed-up can be further increased when dealing with larger datasets and more costly regularisers, e.g., TGV and TNV. In addition to skipping, one can combine stochastic optimisation methods and variance reduction estimators that use only a subset of the data per iteration. For example, in tomography applications, where the per-iteration cost is dominated by the forward and backward operations, one can randomly select a smaller subset of projection angles in addition to a random evaluation of the proximal operator. In this scenario, the per-iteration cost is significantly decreased and preliminary experiments have shown that it outperforms deterministic algorithms in terms of CPU time. Finally, we note that further computational gains could be achieved by applying skipping techniques to the inner solver as well. Notice that Algorithm (6) by default skips the proximal of g. For example, in the case of TV denoising, one can use the approach of Section 2.1, employing ProxSkip on its dual.
In summary, proximal-based algorithms presented herein, along with potential future extensions, open the door to revisiting a range of optimisation solutions for a plethora of imaging modalities developed over the last decades, with the potential to greatly reduce actual computation times.

Acknowledgments

E. Pap acknowledges funding through the Innovate UK Analysis for Innovators (A4i) program: *Denoising of chemical imaging and tomography data* under which the experiments were initially conducted. E. Pap acknowledges also CCPi (EPSRC grant EP/T026677/1). Ž. K. is supported by the UK EPSRC grant EP/X010740/1.

References

[1] The MOSEK optimization toolbox for MATLAB manual. Version 10.1. (2024)

[2] Anthoine, S., Aujol, J.F., Boursier, Y., Melot, C.: Some proximal methods for Poisson intensity CBCT and PET. Inverse Probl. Imaging **6**(4), 565–598 (2012)

[3] Aujol, J.F.: Some first-order algorithms for total variation based image restoration. J. Math. Imaging Vis. **34**(3), 307–327 (2009)

[4] Aujol, J.F., Calatroni, L., et al.: Parameter-free FISTA by adaptive restart and backtracking. SIAM J. Optim. **34**(4), 3259–3285 (2024)

[5] Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Transactions on Image Processing **18**(11), 2419–2434 (2009)

[6] Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)

[7] Biguri, A., Dosanjh, M., et al.: TIGRE: a MATLAB-GPU toolbox for CBCT image reconstruction. Biomed. Phys. Eng. Express **2**(5), 055010 (2016)

[8] Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. SIAM J. Imaging Sci. **3**(3), 492–526 (2010)

[9] Burger, M., Sawatzky, A., Steidl, G.: First Order Algorithms in Variational Image Processing, pp. 345–407. Springer International Publishing (2016)

[10] Chambolle, A., Dossal, C.: On the convergence of the iterates of the "fast iterative shrinkage/thresholding algorithm". J. Optim. Theory Appl. **166**(3), 968–982 (2015)

[11] Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**(1), 120–145 (2011)

[12] Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vis. **20**(1), 89–97 (2004)

[13] Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. Multiscale Model. Sim. **4**(4), 1168–1200 (2005)

[14] Condat, L., Richtárik, P.: Randprox: Primal-dual optimization algorithms with randomized proximal updates. In: 11th ICLR (2023)

[15] Diamond, S., Boyd, S.: CVXPY: A Python-embedded modeling language for convex optimization. J. Mach. Learn. Res. **17**(83), 1–5 (2016)

[16] Gürsoy, D., De Carlo, F., et al.: TomoPy: a framework for the analysis of synchrotron tomographic data. J. Synchrotron Radiat. **21**(5), 1188–1193 (2014)

[17] Holt, K.M.: Total nuclear variation and Jacobian extensions of total variation for vector fields. IEEE Trans. Image Process. **23**(9), 3975–3989 (2014)

[18] Jørgensen, J.S., Ametova, E., et al.: Core Imaging Library - part I: a versatile Python framework for tomographic imaging. Philos. Trans. A Math. Phys. Eng. Sci. **379**(2204), 20200192 (2021)

[19] Kamilov, U.S.: Minimizing isotropic total variation without subiterations. Mitsubishi Electric Research Laboratories (MERL) (2) (2016)

[20] Lefkimmiatis, S., Roussos, A., Unser, M., Maragos, P.: Convex generalizations of total variation based on the structure tensor with applications to inverse problems. In: Scale Space and Variational Methods in Computer Vision. pp. 48–60 (2013)

[21] Liang, J., Luo, T., et al.: Improving Fast Iterative Shrinkage-Thresholding Algorithm: Faster, smarter, and greedier. SIAM J. Sci. Comput. **44**(3), A1069–A1091 (2022)

[22] Matras, D., Vamvakeros, A., et al.: Multi-length scale 5D diffraction imaging of Ni-Pd/CeO2-ZrO2/Al2O3 catalyst during partial oxidation of methane. J. Mater. Chem. A **9**(18), 11331–11346 (2021)

[23] Mirone, A., Brun, E., et al.: The PyHST2 hybrid distributed code for high speed tomographic reconstruction with iterative reconstruction and a priori knowledge capabilities. Nucl. Instrum. Methods Phys. Res. B **324**, 41–48 (2014)

[24] Mishchenko, K., Malinovsky, G., et al.: ProxSkip: Yes! Local gradient steps provably lead to communication acceleration! Finally! In: PMLR 39. vol. 162, pp. 15750–15769 (2022)

[25] Nesterov, Y.E.: A method of solving a convex programming problem with convergence rate $o\big(\frac{1}{k^2}\big)$. Dokl. Akad. Nauk SSSR **269**(3), 543–547 (1983)

[26] O'Donoghue, B., Candès, E.: Adaptive restart for accelerated gradient schemes. Found. Comput. Math. **15**(3), 715–732 (2013)

[27] Papoutsellis, E., Ametova, E., et al.: Core Imaging Library - part II: multichannel reconstruction for dynamic and spectral tomography. Philos. Trans. A Math. Phys. Eng. Sci. **379**(2204), 20200193 (2021)

[28] Pock, T., Chambolle, A.: Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In: 2011 International Conference on Computer Vision. pp. 1762–1769 (2011)

[29] Rasch, J., Chambolle, A.: Inexact first-order primal-dual algorithms. Comput. Optim. Appl. **76**(2), 381–430 (2020)

[30] Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. CAM Reports 08-34, UCLA, Center for Applied Math., 2008. (2) (2008)