Approximate Proximal-Gradient Methods

Anis Hamadouche, Yun Wu, Andrew M. Wallace, and João F. C. Mota

2021 Sensor Signal Processing for Defence Conference (SSPD) | 978-1-6654-3314-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/SSPD51364.2021.9541509

***Abstract*—We study the convergence of the *Proximal-Gradient* algorithm for convex composite problems when both the gradient and the proximal mapping are computed approximately. This scenario occurs when the gradient is computationally expensive and the proximal operator is not available in closed form and may be computed only up to a certain fixed precision. We establish tight deterministic bounds and propose new probabilistic upper bounds on the suboptimality of the function values along the iterations under some statistical assumptions on the perturbed iterates. We use the *Proximal-Gradient* algorithm to solve randomly generated LASSO problems while varying the fixed-point machine representation and the proximal computation precision.**

***Index Terms*—Convex Optimization, Proximal Gradient, Approximate Algorithms.**

I. INTRODUCTION

Many problems in statistics, machine learning, and engineering can be posed as *composite optimization problems*:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\;\; f(x) := g(x) + h(x)\,, \tag{1}$$

where $x \in \mathbb{R}^n$ is the optimization variable, $g : \mathbb{R}^n \to \mathbb{R}$ a differentiable convex function, and $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a closed, proper, and convex function, which is not necessarily differentiable but which enables the inclusion of constraints into (1).

An important example is empirical risk minimization (ERM), the foundational framework in machine learning. There, $g(x) = (1/m)\sum_{i=1}^m \ell(w(z_i;x), y_i)$, where $\{(z_i, y_i)\}_{i=1}^m$ is a collection of training feature vectors z_i and associated labels y_i that we wish to fit with a parametric function $w(\cdot, x)$, and $h(x)$ is a regularizer on the parameters x, e.g., a norm of x. A concrete example of this framework is logistic regression [1].

Another example is compressed sensing [2], in which one attempts to reconstruct a sparse vector $x^\star \in \mathbb{R}^n$ from linear measurements $y = Ax^\star$, where $A \in \mathbb{R}^{m\times n}$ has more columns than rows, i.e., $m < n$. One way to achieve this is by solving (1) with $g(x) = \|Ax - y\|_2^2$ and $h(x) = \|x\|_1$.

Finally, composite problems like (1) arise in control applications, for example in the control of the trajectory of a drone, in which x encodes both a state-vector (e.g., position and velocity of a drone) and the input (e.g., the acceleration in a given direction and steering). In this case, g often encodes a final goal for the state-vector as well as energy penalties, while h encodes state-space dynamics and control constraints [3].

Resource-constrained platforms. Most algorithms that solve (1) assume that computations can be performed with infinite (or near-infinite) precision. While such precision can be achieved in standard computation devices, power-efficient platforms like FPGAs, which are commonly deployed in battery-operated equipment, have much lower precision. Solving problems like (1) under these scenarios often requires completely new strategies [4]. For example, if we solve (1) with standard algorithms, e.g., proximal-gradient or interior-point methods, the resulting solution will satisfy the finite precision constraints of the computing machine rather than infinitely precise solutions satisfying optimal convergence rates. Early termination of iterative algorithms and reduced precision (RP) via finite precision arithmetic can save computational time or power while tolerating losses in accuracy in resource-constrained systems. Furthermore, many optimization software solvers are approximate and this must be accounted for in convergence analysis [5].

Problem statement. The aforementioned approximation techniques come at a cost of reduced accuracy and increased algorithmic perturbations. Given the convex composite optimization problem (1), we define the approximate gradient step operator $T_k^G : \mathbb{R}^n \to \mathbb{R}^n$ at iteration k as

$$T_k^G(x) := x - s_k \nabla^{\epsilon_1^k} g(x)\,, \tag{2}$$

where $s_k > 0$ is the stepsize, $\nabla^{\epsilon_1^k} g := \nabla g + \epsilon_1^k$, and $\epsilon_1^k \in \mathbb{R}^n$ is the gradient error. We also define the approximate proximal operator of h, $T_k^P : \mathbb{R}^n \to \mathbb{R}^n$, as

$$T_k^P(x) := \text{prox}_h^{\epsilon_2^k}(x)\,, \tag{3}$$

where

$$\text{prox}_{\frac{1}{s_k}h}^{\epsilon_2^k}(y) := \Big\{x \in \mathbb{R}^n \,:\, h(x) + \frac{1}{2s_k}\|x-y\|_2^2 \le \epsilon_2^k + \inf_z\; h(z) + \frac{1}{2s_k}\|z-y\|_2^2\Big\}, \tag{4}$$

where $\epsilon_2^k \in \mathbb{R}$ is the error associated to the proximal computation. Then, the approximate proximal gradient operator is given by the following operator product

$$T_k^{PG} = T_k^P T_k^G\,. \tag{5}$$

The approximate proximal gradient algorithm sequence is generated by sequentially applying the mapping sequence $\{T_k^{PG}\}_{k>0}$, i.e.,

$$x^{k+1} = T_k^{PG}(x^k) = \text{prox}_h^{\epsilon_2^k}(x^k - s_k\nabla^{\epsilon_1^k} g(x^k)). \tag{6}$$

Work supported by UK's EPSRC (EP/T026111/1, EP/S000631/1), and the MOD University Defence Research Collaboration.

Anis Hamadouche, Yun Wu, Andrew M. Wallace, and João F. C. Mota are with the School of Engineering & Physical Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK. (e-mail: {ah225,y.wu,a.m.wallace,j.mota}@hw.ac.uk).

More precisely, our goal is to establish conditions on the problem and on the errors ϵ_1^k and ϵ_2^k under which (6) *converges. In such cases, we also aim to obtain the respective rate of convergence.* **Summary of prior work.** It is known that when (1) is convex and g has an *L-Lipschitz-continuous* gradient, then the exact proximal method, i.e., with $\epsilon_1^k = 0_n$ and $\epsilon_2^k = 0$ for all k, and its accelerated counterpart, require, respectively, $O(1/\rho)$ and $O(\sqrt{1/\rho})$ iterations to achieve an error ρ in the objective function [6], [7]. Although this seems promising in noise-free applications, running the same type of algorithms in resource-constrained environments often leads to unexpected outcomes, as the well-known optimal convergence bounds no longer hold in the presence of gradient and proximal computation errors.

Following [8], the work in [9] showed that the above nearly optimal rates can still be achieved when the computation of the gradients and proximal operators are approximate. This variant is also known as the *Inexact Proximal-Gradient* algorithm. The analysis in [9] requires the errors to decrease with iterations k at rates $O(1/k^{a+1})$ for the basic PG, and $O(1/k^{a+2})$ for the accelerated PG for any $a > 0$ in order to satisfy the summability assumptions of both error terms. The work in [9] established the following ergodic convergence bound in terms of function values of the averaged iterates for the basic *approximate PG*:

$$f\Big(\frac{1}{k}\sum_{i=1}^{k} x^i\Big) - f(x^\star) \leq \frac{L}{2k}\Big[\Big\|x^\star - x^0\Big\|_2 + 2A_k + \sqrt{2B_k}\Big]^2, \tag{7}$$

where $x^\star$ is any optimal solution of (1), L is the Lipschitz constant of ∇g, x^0 is the initialization vector, and

$$A_k = \sum_{i=1}^{k}\Big(\frac{\|\epsilon_1^i\|_2}{L} + \sqrt{\frac{2\epsilon_2^i}{L}}\Big) \qquad B_k = \sum_{i=1}^{k}\frac{\epsilon_2^i}{L}\,.$$

Our approach. In the case of deterministic errors ϵ_1^k and ϵ_2^k, we get inspiration from [7] to derive, using simple arguments, upper bounds on $f\big(\frac{1}{k}\sum_{i=1}^{k} x^i\big) - f(x^\star)$ throughout the iterations. The resulting bounds not only are simpler and tighter than (7), but also decouple the contribution of the two types of errors, ϵ_1^k and ϵ_2^k. In the case of random errors, we show that we can bypass the need to assume that ϵ_1^k and ϵ_2^k converge to zero. We believe this line of reasoning is novel in the analysis of approximate PG algorithms.

Contributions. We summarize our contributions as:

- We establish convergence bounds for the approximate PG in the presence of deterministic errors.
- We extend the analysis to incorporate random errors and propose new parameterized probabilistic convergence bounds with a tuning parameter.
- We propose new models for the proximal and gradient errors that satisfy interesting martingale properties in consistence with experimental results.

II. MAIN RESULTS

All the proofs of the results in this section will appear in a subsequent publication.

Consider the approximate PG algorithm in (6). Before stating our convergence guarantees for approximate PG, we specify our main assumptions on the problem and describe the class of algorithms that our analysis covers.

All of our results assume the following:

Assumption II.1 (Assumptions on the problem)**.**

- *The function $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is closed, proper, and convex.*
- *The function $g : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, and its gradient $\nabla g : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz-continuous with constant $L > 0$, that is,*

$$\big\|\nabla g(y) - \nabla g(x)\big\|_2 \leq L\big\|y - x\big\|_2\,, \tag{8}$$

  for all x, $y \in \mathbb{R}^n$, where $\|\cdot\|_2$ stands for the standard Euclidean norm.
- *The set of optimal solutions of* (1) *is nonempty.*

The above assumptions are standard in the analysis of PG algorithms and are actually required for convergence to an optimal solution from an arbitrary initialization.

Error models and assumptions. Our analysis assumes two different scenarios:

1) The sequences of errors $\{\epsilon_1^k\}_{k\geq 1}$ and $\{\epsilon_2^k\}_{k\geq 1}$ are deterministic, or
2) The sequences of errors $\{\epsilon_1^k\}_{k\geq 1}$ and $\{\epsilon_2^k\}_{k\geq 1}$ are discrete stochastic processes, in which case we use $\epsilon_{1_\Omega}^k$ and $\epsilon_{2_\Omega}^k$ to denote their respective realizations at iteration k.

In scenario 2), the sequences $\{x^k\}_{k\geq 1}$ and $\{y^k\}_{k\geq 1}$ become random as well. And we also use x_Ω^k and y_Ω^k to denote the respective random vectors at iteration k, where Ω denotes the sample space of a given probability space. We make the following assumption in this case:

Assumption II.2. *In scenario 2), we assume that each random vector $\epsilon_{1_\Omega}^k$, for $k \geq 1$, satisfies*

$$\mathbb{E}\big[\epsilon_{1_\Omega}^k \,\big|\, \epsilon_{1_\Omega}^1, \ldots, \epsilon_{1_\Omega}^{k-1}\big] = \mathbb{E}\big[\epsilon_{1_\Omega}^k\big] = 0\,, \tag{9a}$$

$$\mathbb{P}\big(|\epsilon_{1_\Omega j}^k| \leq \delta\big) = 1\,, \qquad \text{for all } j = 1, \ldots, n, \tag{9b}$$

$$\mathbb{E}\big[{\epsilon_{1_\Omega}^k}^\top x_\Omega^k \,\big|\, \epsilon_{1_\Omega}^1, \ldots, \epsilon_{1_\Omega}^{k-1},\, x_{1_\Omega}^1, \ldots, x_{1_\Omega}^{k-1}\big] = \mathbb{E}\big[{\epsilon_{1_\Omega}^k}^\top x_\Omega^k\big] = 0\,, \tag{9c}$$

where $\epsilon_{1_\Omega j}^k$ in (9b) *denotes the j-th entry of $\epsilon_{1_\Omega}^k$, and $\delta > 0$ is some finite constant.*

The first assumption, (9a), states that $\epsilon_{1_\Omega}^k$ is independent from past realizations and has zero mean. The second assumption, (9b), states that the absolute value of each entry of $\epsilon_{1_\Omega}^k$ is bounded by δ almost surely. Finally, the third assumption, (9c), states that $\epsilon_{1_\Omega}^1, \ldots, \epsilon_{1_\Omega}^{k-1}, \epsilon_{1_\Omega}^k$ and $x_{1_\Omega}^1, \ldots, x_{1_\Omega}^{k-1}, x_{1_\Omega}^k$ are mutually independent.

Let us define the residual error vector as follows:

$$r_\Omega^k = x_\Omega^k - \overline{x}^k, \tag{10}$$

where x_Ω^k and $\overline{x}^k$ stand for the perturbed and gradient-error-free iterates, respectively. Similar assumptions can be made about r_Ω^k, mainly:

$$\mathbb{E}\big[r_\Omega^k \,\big|\, r_\Omega^1, \dots, r_\Omega^{k-1}\big] = \mathbb{E}\big[r_\Omega^k\big] = 0\,, \tag{11a}$$

$$\mathbb{E}\big[{r_\Omega^k}^\top x_\Omega^k \,\big|\, r_\Omega^1, \dots, r_\Omega^{k-1},\, x_{1_\Omega}^1, \dots, x_{1_\Omega}^{k-1}\big] = \mathbb{E}\big[{r_\Omega^k}^\top x_\Omega^k\big] = 0\,. \tag{11b}$$

Lemma II.3. *Let x^k and $\overline{x}^k$ be the approximate and exact proximal-gradient iterates and let ϵ_2^k be the proximal error at instant k. Assume a constant stepsize $s_k = s > 0$, for all k. Then, the norm of the residual vector $r^k = x^k - \overline{x}^k$ satisfies*

$$\big\|r^k\big\|_2 \leq \sqrt{2s\epsilon_2^k}, \quad \forall_{k>0}. \tag{12}$$

Lemma II.3 bounds the norm of the residual vector r^k as a function of ϵ_2^k. Therefore, boundedness of the latter implicitly implies boundedness of the norm of the former.

We start by considering deterministic error sequences $\{\epsilon_1^k\}_{k\geq 1}$ and $\{\epsilon_2^k\}_{k\geq 1}$, and then we consider the case in which these sequences are random, as in Assumption II.2.

Deterministic errors. Our first result provides a bound for the ergodic convergence of the sequence of function values, and decouples the contribution of the errors in the computation of gradient and in the computation of the proximal operator.

Theorem II.4 (PG, deterministic errors). *Consider problem* (1) *and let Assumption II.1 hold. Then, for arbitrary error sequences $\{\epsilon_1^k\}_{k\geq 1}$ and $\{\epsilon_2^k\}_{k\geq 1}$, the sequence generated by approximate PG in* (6) *with constant stepsize $s_k := s \leq 1/L$, for all k, satisfies*

$$\begin{aligned} f\big(\frac{1}{k+1}\sum_{i=0}^{k} x^{i+1}\big) - f(x^\star) \leq \frac{1}{k+1}\Big[\sum_{i=0}^{k}\epsilon_2^i \\ + \sum_{i=0}^{k}\Big(\epsilon_1^i - \frac{1}{s}r^{i+1}\Big)^\top (x^\star - x^{i+1}) + \frac{1}{2s}\Big\|x^\star - x^0\Big\|_2^2\Big] \\ - \frac{1}{k+1}\Big[\frac{1}{2s}\sum_{i=0}^{k}\big\|r^{i+1}\big\|_2^2 + \frac{1}{2s}\big\|x^\star - x^{k+1}\big\|_2^2\Big], \end{aligned} \tag{13}$$

where $x^\star$ is any solution of (1) *and r^i is the residual vector associated with error ϵ_2^i defined in* (10)*.*

This result implies that the well-known $O(1/k)$ convergence rate for the gradient method without errors still holds when both ϵ_2^k and $(\epsilon_1^k - \frac{1}{s}r^{k+1})^\top(x^\star - x^k)$ are summable. Note that a faster convergence of these two errors will not improve the convergence rate but will yield a better coefficient.

Consider now the case in which the sequence $\{r^k\}_{k>0}$ cannot be observed [as $\overline{x}^k$ in (10) is usually unobservable], but is bounded, e.g., if $\{\epsilon_2^k\}$ is bounded as in Lemma II.3. Then, to obtain a convergence bound that is independent of the particular sequences $\{x^k\}_{k\geq 0}$ and $\{r^k\}_{k>0}$, we can apply Cauchy-Schwarz's inequality to the first term involving r^k in the right-hand side of (13) followed by Féjer's inequality (see [7, Thm. 10.23]):

Corollary II.5. *Under the same conditions as Theorem II.4, the sequence generated by approximate PG in* (6) *satisfies*

$$\begin{aligned} f\big(\frac{1}{k+1}\sum_{i=0}^{k} x^{i+1}\big) - f(x^\star) \leq \frac{1}{k+1}\Big[\sum_{i=0}^{k}\epsilon_2^i \\ + \sum_{i=0}^{k}\big\|\epsilon_1^i - \frac{1}{s}r^{i+1}\big\|_2\big\|x^\star - x^{i+1}\big\|_2 + \frac{1}{2s}\Big\|x^\star - x^0\Big\|_2^2\Big] \\ - \frac{1}{k+1}\Big[\frac{1}{2s}\sum_{i=0}^{k}\big\|r^{i+1}\big\|_2^2 + \frac{1}{2s}\big\|x^\star - x^{k+1}\big\|_2^2\Big], \end{aligned} \tag{14}$$

where, again, $x^\star$ is any solution of (1) *and we used Lemma II.3 to bound $\big\|r^k\big\|$.*

Notice that, by Féjer's inequality [7, Thm. 10.23], $\big\|x^\star - x^0\big\|_2$ upper bounds all residuals $\big\|x^\star - x^i\big\|_2$. Since $\{\epsilon_1^k\}_{k\geq 1}$ is a centered sequence, the use of Cauchy-Schwarz's inequality followed by Féjer's inequality yields a bound looser than the one in (13). Yet, the $O(1/k)$ convergence rate is still guaranteed with weaker summability assumptions of $\{\epsilon_2^k\}_{k\geq 1}$ and $\{\big\|\epsilon_1^k\big\|\}_{k\geq 1}$. If we set both errors to zero for all $k \geq 1$, we recover the error-free optimal upper bound $\frac{L}{2k}\big\|x^\star - x^0\big\|_2^2$ [7].

Next we relax the summability assumption on ϵ_1^k and ϵ_2^k and replace it with the weaker assumption of *boundedness*.

Random errors. Let us now consider the case in which ϵ_1^k, ϵ_2^k and therefore x^k, are random, and let $\epsilon_{1_\Omega}^k$, $\epsilon_{2_\Omega}^k$ and x_Ω^k be the corresponding random variables/vectors, respectively.

Theorem II.6 (Random errors). *Consider problem* (1) *and let Assumption II.1 hold. Assume that the rounding error $\{\epsilon_{1_\Omega}^k\}_{k\geq 1}$ and residual error $\{r_\Omega^k\}_{k\geq 1}$ sequences satisfy Assumption II.2 and $\mathbb{P}\big(\epsilon_{2_\Omega}^k \leq \varepsilon_0\big) = 1$, for all $k > 0$, and for some $\varepsilon_0 \in \mathbb{R}$. Then, for any $\gamma > 0$, the sequence generated by approximate PG in* (6) *with constant stepsize $s_k := s \leq 1/L$, for all k, satisfies*

$$\begin{aligned} f\Big(\frac{1}{k}\sum_{i=1}^{k} x_\Omega^i\Big) - f(x^\star) \leq \frac{1}{k}\sum_{i=1}^{k}\epsilon_{2_\Omega}^i + \\ \frac{\gamma}{\sqrt{k}}\left(\sqrt{n}|\delta| + \sqrt{\frac{2\varepsilon_0}{s}}\right)\Big\|x^\star - x^0\Big\|_2 + \frac{1}{2sk}\Big\|x^\star - x^0\Big\|_2^2\,, \end{aligned} \tag{15}$$

with probability at least $1 - 2\exp(-\frac{\gamma^2}{2})$, where $x^\star$ is any solution of (1)*.*

For large scale problems,[1] we typically have $n \gg \frac{1}{s} \geq L$; therefore, we obtain the following approximated bound,

$$\begin{aligned} f\Big(\frac{1}{k}\sum_{i=1}^{k} x_\Omega^i\Big) - f(x^\star) \lesssim \frac{1}{k}\sum_{i=1}^{k}\epsilon_{2_\Omega}^i + \gamma\sqrt{\frac{n}{k}}|\delta|\Big\|x^\star - x^0\Big\|_2 \\ + \frac{1}{2sk}\Big\|x^\star - x^0\Big\|_2^2\,, \end{aligned} \tag{16}$$

[1] And for same levels of error magnitudes δ and ε_0.

with the same probability. In the absence of computational errors, (15) coincides with the results of Theorem II.4 and Corollary II.5, which reduce to the deterministic noise-free convergence upper bound, i.e, $\frac{L}{2k}\left\|x^\star - x^0\right\|_2^2$. With exact proximal operation and approximate gradient computations (i.e., $\epsilon_2^k = 0$ and $\epsilon_2^k \neq 0$ for all $k \geq 0$), if we let the machine precision δ to decrease at $O(\frac{1}{k^{0.5+\delta}})$, i.e, progressively increase computation accuracy, then we obtain the optimal convergence rate $O(\frac{1}{k})$. In order to recover the same convergence rate for the approximate proximal case, we also need the sum of the ensemble means $\mathbb{E}(\epsilon^i_{2_\Omega})$ to decrease as $O(\frac{1}{k^{1+\delta}})$, which is a weaker than what [9] [cf.(7)] requires: $O(\frac{1}{k^{2+\delta}})$. This result also suggests that a slower $O(\frac{1}{\sqrt{k}})$ convergence rate (same as noise-free subgradient method) is achieved when the sequence of ensemble means $\{\mathbb{E}(\epsilon^k_{2_\Omega})\}$ is summable for all centered and bounded sequences $\{\epsilon^k_{1_\Omega}\}$, and consequently the proximal error is the main contributor to any divergence from the optimal set $X^\star$.

Notice that for a fixed machine precision δ and probability parameter γ we obtain a computable error residual constant rather than variable running error terms as in Theorem II.4, Corollary II.5 or (7) without making any summability assumptions on $\{\left\|\epsilon^k_{1_\Omega}\right\|\}$ (as in Corollary II.5) or $\{\epsilon^k_{1_\Omega}\}$ in general.

Moreover, the effect of the dimension n of the problem variable appears explicitly in (15), but neither in Theorem II.4, nor in Corollary II.5, nor in (7). The latter suggests that using progressively sparser gradient vectors[2] can potentially accelerate the convergence speed (e.g, by using $n' \ll n$), but never faster than the optimal (limit) speed of $O(\frac{1}{k})$. Overall, better design parameter selections would result in better error residuals rather than exceeding the optimal convergence rate.

The following result applies if we relax the summability of $\{\epsilon^k_{2_\Omega}\}$ but still assume statistical stationarity.[3]

Theorem II.7 (**Random errors**). *Consider problem* (1) *and let Assumption II.1 hold. Assume that the rounding error* $\{\epsilon^k_{1_\Omega}\}_{k\geq 1}$ *and residual error* $\{r^k_\Omega\}_{k\geq 1}$ *sequences satisfy Assumption II.2, and that the proximal computation error is upper bounded, i.e.,* $\epsilon^k_{2_\Omega} \leq \varepsilon_0$ *for all* $k \geq 1$*, and also stationary with constant mean* $\mathbb{E}(\epsilon_{2_\Omega})$*. Then, the sequence generated by approximate PG in* (6) *with constant stepsize* $s_k := s \leq 1/L$*, for all* k*, satisfies*

$$f\left(\frac{1}{k}\sum_{i=1}^{k} x^i_\Omega\right) - f(x^\star) \leq E\big(\epsilon_{2_\Omega}\big) + \frac{\gamma}{\sqrt{k}}\left(\frac{\varepsilon_0}{2} + \sqrt{n}|\delta|\left\|x^\star - x^0\right\|_2\right) + \frac{1}{2sk}\left\|x^\star - x^0\right\|_2^2, \tag{17}$$

with probability at least $1 - 2\exp(-\frac{\gamma^2}{2})$*, where* $x^\star$ *is any solution of* (1)*.*

The following corollary applies to approximate PG with uniformly distributed proximal error.

[2]As is the case in *Proximal-Gradient* algorithm when applied to LASSO.

[3]This means that the ensemble mean and the variance are time-invariant.

Corollary II.8 (**Random uniformly distributed proximal error**). *Let the proximal computation error be upper bounded, i.e.,* $\epsilon^k_{2_\Omega} \leq \varepsilon_0$*, for all* $k \geq 1$*. If the latter is stationary and uniformly distributed over its range, i.e.,* $\epsilon^k_{2_\Omega} \sim \mathcal{U}\{0, \varepsilon_0\}$*, then substituting* $\mathbb{E}(\epsilon_{2_\Omega}) = \frac{\varepsilon_0}{2}$ *in the bound of Theorem II.7 gives*

$$f\left(\frac{1}{k}\sum_{i=1}^{k} x^i_\Omega\right) - f(x^\star) \leq \frac{\varepsilon_0}{2} + \frac{\gamma}{\sqrt{k}}\left(\frac{\varepsilon_0}{2} + \sqrt{n}|\delta|\left\|x^\star - x^0\right\|_2\right) + \frac{1}{2sk}\left\|x^\star - x^0\right\|_2^2, \tag{18}$$

with probability at least $1 - 2\exp(-\frac{\gamma^2}{2})$*, where* $x^\star$ *is any solution of* (1)*.*

In terms of the proximal error $\epsilon^k_{2_\Omega}$, using a concentration-based probabilistic bound, i.e., $\epsilon^k_{2_\Omega} \leq (1/2 + \gamma/\sqrt{k})\varepsilon_0$ results in a sharper bound than what we would have obtained if we used the more conservative deterministic upper bound $\epsilon^k_{2_\Omega} \leq \varepsilon_0$.

III. Quantization Errors

We start by briefly reviewing the theory behind hardware quantization. Quantization is a critical step between data-level and hardware-level, which can be thought of as a type of contract between the two levels of the application in order to allocate a certain (finite) number of bits (resource) to represent an infinitely precise parameter or a value from a continuous signal in the digital circuit with finite precision. This very initial step of data type/hardware design plays a major role in determining the overall precision for the application as well as the complexity of the implementation.

Quantization errors can be reduced by choosing an appropriate number of bits (usually the higher the number of used bits the less the error will be). However, this reduction comes at the cost of using more hardware resources for storage and computation. The trade-off between error attenuation and calculation precision, energy and speed (or latency), is subject to application constraints.

Fixed-point number system. Fixed-point machine representation can be interpreted in different ways, and we discuss here the two most common approaches. In the first case, all the word (W) bit elements are allocated for value representation assuming all signal values are always positive. As can be deduced from the machine representation system's name, i.e, "fixed-point," the set of bits in W are split by a fixed-point into I most significant bits (MSBs) and F least significant bits (LSBs) representing the integer and the fractional parts, respectively. This quantization is called unsigned I.F or *uI.F* for short. The quantized value is evaluated by the following formula,

$$uI.F(x) = \sum_{i=0}^{W-1} b_i(x)2^{i-F}, \tag{19}$$

with $F, W \in \mathbb{N}_+$. The corresponding dynamic range (DR) is given by $DR_{uI.F} = [0; 2^I - 2^{-F}]$. The signed I.F, or *sI.F* can be obtained from I.F by encoding the sign of the value

using one bit and this is typically done by taking the most significant bit (MSB) of the integer part I as being a sign bit. Although this operation would reduce the integer part's number of bits I to $I-1$, the two's complement approach handles negative numbers and therefore extends the DR in the negative direction, i.e., $DR_{sI.F} = [-2^{I-1}; 2^{I-1} - 2^{-F}]$, and the quantized value is now given by

$$sI.F(x) = \sum_{i=0}^{W-2} b_i 2^{i-F} - b_{W-1} 2^{I-1}. \tag{20}$$

Quantization Rules. Once a machine representation is defined, quantization rules need to be established in order to define the quantization behaviour under decreasing accuracy and/or overflow in subsequent operations. Although the IEEE 754 standard uses *round-to-nearest integer*[4], other modes of rounding also exist in the literature. For instance, rounding towards the standard limits $(0, +\infty, -\infty)$ is also known as *directed rounding*. Moreover, rounding can be deterministic in a pre-defined rule or stochastic according to a random distribution.

IV. EXPERIMENTAL RESULTS

We now apply the proposed bounds to analyze the convergence of the approximate *proximal gradient* algorithm when applied to solve randomly generated LASSO problems:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \ \frac{1}{2}\|Ax-y\|_2^2 + \lambda\|x\|_1 \,,$$

where $n = 100$ (dimension of x) and $A \in \mathbb{R}^{m\times n}$ has $m = 500$ rows. We run a total of 5 random experiments for every algorithm parameter selection. We mainly vary the bitwidth (BIT), the fraction width (FRAC.) of the fixed-point representation in (20), the CVX [10] solver's precision (PRECISION) to approximate the proximal step (4), and the tolerance bound of the approximate PG (ABSTOL in Table I). We record and take the average over all 5 experiments of the residual error in the iterates $\|x - x^\star\|_2$, the residual error (suboptimality) in the function values $f - f^\star$, and the total number of iterations k $(k, \textit{iters.})$. The results are summarized in Table I. We also plot the proposed convergence bounds, the error-free optimal bound as well as the original bound in (7) for the different tests as depicted in Figs. 1-3. Note that for the probabilistic bounds we tune the parameter γ to obtain 3 different bounds which hold with probabilities 1, 0.5 and 0.25, respectively. From Figs.1-3, we can clearly see that our proposed bounds give better approximations of the discrepancy caused by perturbations, and consequently we obtain better error terms. As a necessary condition for convergence, we only required the partial sums $\sum_{i=1}^k \epsilon_2^i$ and $\sum_{i=1}^k \left\|\epsilon_1^i\right\|_2$ to be in $o(k)$, in contrast to the stronger condition $o(\sqrt{k})$ of (7). For the probabilistic bounds, we do not assume summability of the error terms but only require them to be

[4]If the correct answer is exactly halfway between the two, the system chooses the output where the least significant bit of the fraction (mantissa M) is zero.

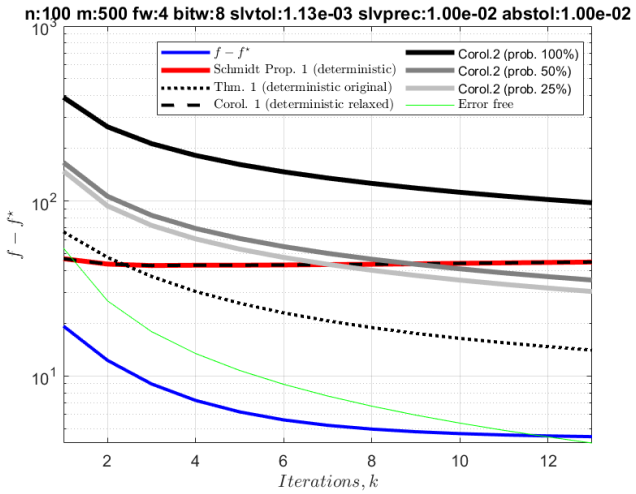

Fig. 1. Upper bounds based on Theorems II.4 & II.7 and their corresponding corollaries vs (7)

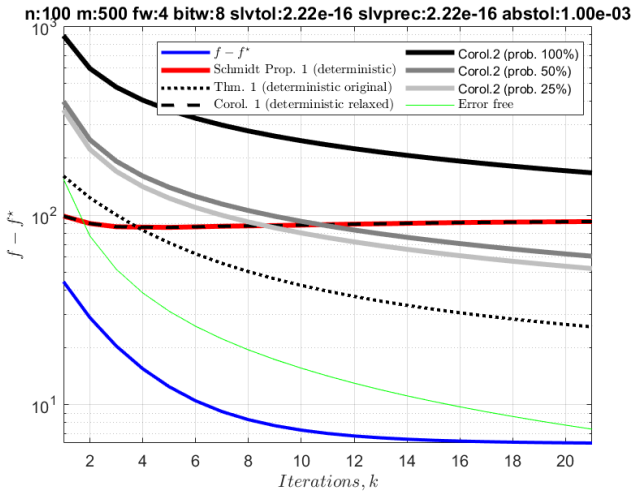

Fig. 2. Upper bounds based on Theorems II.4 & II.7 and their corresponding corollaries vs (7)

Table I
RESULTS OF OUR EXPERIMENTS.

PRECISION	BIT	FRAC.	ABSTOL	$\|x - x^\star\|_2$	$f - f^\star$	$k, iters.$
2.22e-16	8	4	2.22e-16	0.052609	1.3818	85
			0.001	0.15256	6.2202	20
	16	6	2.22e-16	0.13947	5.1224	79
			0.01	0.1418	5.6894	15
		8	2.22e-16	0.09152	3.1661	79
0.001	8	4	0.001	0.14508	5.555	20
			0.01	0.14271	5.6136	14
	16	8	2.22e-16	0.096077	4.1512	84
			0.001	0.13155	4.5246	18
0.01	8	4	0.01	0.12847	4.4239	14
	16	6	0.01	0.13084	4.3855	14
		8	0.01	0.15369	5.8671	15

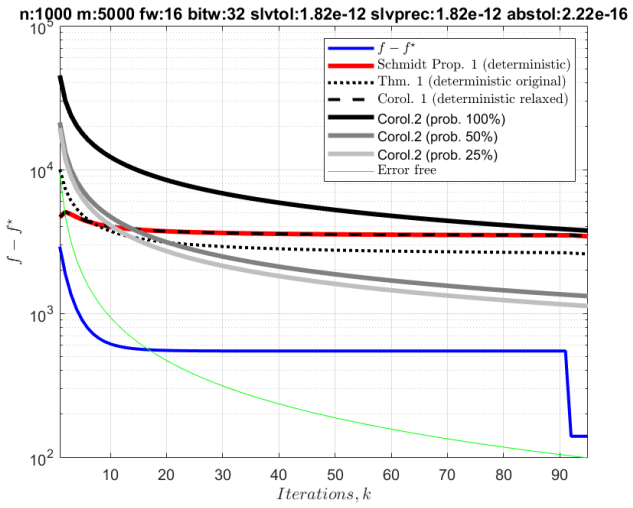

Fig. 3. Upper bounds based on Theorems II.4 & II.7 and their corresponding corollaries vs (7)

bounded. Consequently, the probabilistic bounds achieve better approximations over iterations and are less sensitive to error variations and become tighter with decreasing probability. If we relax our original bound of Theorem II.4 and use Lemma II.3 to bound the sequence of the proximal residual error $\{r^k\}$, then our bound coincides with the one in (7), as depicted by the overlapping dashed and red lines in Figs. 1-3.

Increasing the tolerance bound of the approximate PG from 2.22×10^{-16} to 10^{-3} improved the algorithm's running time by 65 iterations for the 8 bits representation and 66 iterations for the 16 bits representation without affecting too much the residuals.

Table I shows that, in general, varying the internal loop (CVX solver's) precision does not largely affect the number of outer iterations of the PG, but leads to substantial bias around the optimum when increased from 0.01 by a factor of 10.

Reducing the hardware precision from 16 to 8 bits accelerated the algorithm by 6 iterations, but slightly increased the residual in the solution $\left\|x - x^\star\right\|_2$ by 8.6861×10^{-2} while added 3.7406 extra bias error to the function value. Increasing the hardware precision by allocating more bits for the fractional part in the fixed-point representation of 16 bits caused the residual error in $\left\|x - x^\star\right\|_2$ to drop by 17.46% and the error in $\left\|f - f^\star\right\|_2$ by 33.78% without remarkable effect on the number of iterations.

V. CONCLUSIONS

We considered the proximal-gradient algorithm in the case in which the gradient of the differentiable function and the proximal operator are computed with errors. We obtained new bounds, tighter than the ones in [9] and demonstrated their validity on a practical optimization example (LASSO) solved on a reduced-precision machine combined with reduced-precision solver. While we established worst-case performance bounds, we also established probabilistic upper bounds catering for random computational errors. Interesting directions for future work include relaxing the assumptions in order to incorporate more general perturbations into the analysis and considering accelerated versions (i.e., Nesterov) of the approximate proximal-gradient algorithm.

REFERENCES

[1] J. S. Cramer, "The origins of logistic regression," 2002.

[2] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] Y. Wu, J. F. Mota, and A. M. Wallace, "Approximate lasso model predictive control for resource constrained systems," in *2020 Sensor Signal Processing for Defence Conference (SSPD)*, IEEE, 2020, pp. 1–5.

[4] P. Machart, S. Anthoine, and L. Baldassarre, "Optimal computational trade-off of inexact proximal methods," *arXiv preprint arXiv:1210.5034*, 2012.

[5] M. Ulbrich and B. van Bloemen Waanders, "An introduction to partial differential equations constrained optimization," *Optimization and Engineering*, vol. 19, no. 3, pp. 515–520, 2018.

[6] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[7] A. Beck, *First-order methods in optimization*. SIAM, 2017, vol. 25.

[8] A. Beck and M. Teboulle, "Gradient-based algorithms with applications to signal recovery," *Convex optimization in signal processing and communications*, pp. 42–88, 2009.

[9] M. Schmidt, N. L. Roux, and F. R. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization," in *Advances in neural information processing systems*, 2011, pp. 1458–1466.

[10] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming*, cvxr.com/cvx, 2011.