SRKCD: A STABILIZED RUNGE–KUTTA METHOD FOR STOCHASTIC OPTIMIZATION

TONY STILLFJORD AND MÅNS WILLIAMSON

Abstract. We introduce a family of stochastic optimization methods based on the Runge–Kutta–Chebyshev (RKC) schemes. The RKC methods are explicit methods originally designed for solving stiff ordinary differential equations by ensuring that their stability regions are of maximal size. In the optimization context, this allows for larger step sizes (learning rates) and better robustness compared to e.g. the popular stochastic gradient descent method. Our main contribution is a convergence proof for essentially all stochastic Runge–Kutta optimization methods. This shows convergence in expectation with an optimal sublinear rate under standard assumptions of strong convexity and Lipschitz-continuous gradients. For non-convex objectives, we get convergence to zero in expectation of the gradients. The proof requires certain natural conditions on the Runge–Kutta coefficients, and we further demonstrate that the RKC schemes satisfy these. Finally, we illustrate the improved stability properties of the methods in practice by performing numerical experiments on both a small-scale test example and on a problem arising from an image classification application in machine learning.

1. Introduction

In this article we consider the optimization problem

$$\min_w F(w)$$

where F is differentiable. Such problems frequently arise in many contexts, e.g. for training neural networks in the currently popular subject of machine learning. We focus on the large-scale case where computing $\nabla F(w)$ is expensive, and assume that cheap approximations $g(\xi, w) \approx \nabla F(w)$ are available.

At a (local) minimum w_*, it holds that $\nabla F(w_*) = 0$, and such a stationary point of the gradient may be found by evolving the gradient flow

$$\dot{w}(t) = -\nabla F(w(t))$$

over the pseudo-time $t \in [0, \infty)$. The benefit of this reformulation is that many optimization methods for the original problem may now be stated as time-stepping methods for the gradient flow. We recognize e.g. the explicit Euler method with varying step sizes α_k

$$w_{k+1} = w_k - \alpha_k \nabla F(w_k)$$

Centre for Mathematical Sciences, Lund University, P.O. Box 118, 221 00 Lund, Sweden

E-mail addresses: tony.stillfjord@math.lth.se, mans.williamson@math.lth.se.

2010 *Mathematics Subject Classification.* 90C15; 65K05; 65L20.

Key words and phrases. stochastic optimization; convergence analysis; Runge–Kutta–Chebyshev; stability;

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

as the gradient descent (GD) method. The popular *stochastic* gradient descent (SGD) [16] method uses the same formula but with the approximation $g(\xi_k, w_k)$ instead of $\nabla F(w_k)$, where ξ_k is a random variable that typically indicates which randomly chosen parts of F to look at. SGD is therefore a perturbed version of explicit Euler.

As was observed already in [14], the gradient flows arising from neural networks tend to be stiff. As a consequence, explicit methods suffer from severe step size restrictions. This is particularly inconvenient when one wants to reach a stationary state, which typically requires evolving the system for a long time. While it is difficult to quantify exactly how the stochasticity introduced in methods like SGD affects this, they suffer from similar step size restrictions.

A way to avoid such step size restrictions would be to instead use methods with better stability properties, such as A-stable methods. This, however, requires that method is implicit. One such method would be implicit Euler, which, when applied to the gradient flow is equivalent to the proximal point method in the context of optimization [2, 6]. While this can be applied in certain cases when F has a specific structure that allows the arising nonlinear equation systems to be solved efficiently, in general (usually) this is not feasible.

An alternative, which to our knowledge has only been considered to a very small extent in the optimization community, is the use of explicit stabilized schemes. These are constructed such that their stability regions are maximized. Thus, there will still be a step size restriction, but of a more benign type. A large class of such methods are the Runge-Kutta-Chebyshev methods [18], see also [10] for an overview and further references. They are explicit Runge-Kutta methods, i.e. of the form

$$w_{k,i} = w_k - \alpha_k \sum_{j=1}^{i} a_{i,j} \nabla F(\xi_k, w_{k,j-1}), \quad i = 0, \dots, s-1,$$
$$w_{k+1} = w_k - \alpha_k \sum_{i=1}^{s} b_i \nabla F(\xi_k, w_{k,i-1}),$$

where the coefficients $a_{i,j}$ and b_i have been chosen in a very specific way such that the stability region extends as far into the left half-plane as possible. The tradeoff compared to GD is that such a scheme with s stages requires s times as many gradient evaluations. However, it still pays off, because the stability region grows as s^2. An optimization method called the Runge-Kutta-Chebyshev descent (RKCD) based on this idea has recently been investigated in [5]. However, only for the case where ∇F can be computed exactly and for a rather restrictive class of problems. In this article, we propose a stochastic version of such a scheme which we call the stochastic Runge-Kutta-Chebyshev descent (SRKCD). Compared to e.g. SGD, it has superior stability properties.

There are of course other advanced methods that can be applied to the problem, and there is a rather large number of papers on the subject. We refer to [3] for a general overview. Here, we mention for example accelerated gradient-type methods such as the SGD with momentum [15, 17], the stochastic heavy ball method [7] and Nesterov's accelerated gradient method [13]. These do not use only the approximate gradient at the current iteration w_k but modify this gradient using other gradient information acquired in previous steps. A different class of methods are the adaptive

learning rate methods, containing e.g. AdaGrad [4], AdaDelta [19], Adam [11], RMSprop [9] and AdaMax [11]. These are typically formulated as adapting the step size α_k based on a constantly updated model of the local cost landscape, acquired from gradient information computed in previous iterations. However, since most of them adjust the step size for each component of w_k separately, they could in a certain sense be seen as instead modifying the approximation $g(\xi_k, w_k)$ like the accelerated gradient methods.

In contrast to this, the method we propose simply uses the available gradient information without modifications and allows each step to be longer. Just like SGD may be extended to e.g. SGD with momentum, one might also consider SRKCD with momentum, provided that further analysis on the properties of this combined method is performed.

The main contribution of this article is a rigorous proof of convergence for a general Runge-Kutta method, under weak assumptions on its coefficients and standard assumptions on the optimization problem and the approximations $g(\xi, w)$. We emphasize that while the proof applies to SRKCD, it is more widely applicable. We consider two settings. First, the usual strongly convex setting, wherein we can prove optimal convergence orders of the type $\mathcal{O}(1/k)$. Secondly, the fully non-convex setting where we show that the squared norm of $\nabla F(w_k)$ goes to zero in expectation. This is also essentially optimal. In both cases, the results are direct extensions of similar results for SGD.

We note that nonlinear stability analysis is a very complex topic with few generally applicable results, and that the stability region of a method only refers to the setting of linear problems. For these reasons, it is not possible to use the available information on the RKC stability regions to tailor the general convergence proof further for SRKCD. The benefits of the improved stability properties in SRKCD are therefore not directly illustrated by the convergence proof. For this reason, we also perform numerical experiments which demonstrate that in practice they are present also in the stochastic non-linear and non-convex setting.

The outline of the paper is as follows. Section 2 contains the main error analysis for the general Runge-Kutta methods. It begins by formalizing the notation and assumptions on the problem, then presents preliminary results in Subsections 2.1 and 2.2. The actual convergence proofs are presented in Subsections 2.3 (convex case) and 2.4 (nonconvex case). Then we study the SRKCD method specifically in Section 3 and discuss its properties. The numerical experiments follow in Section 4 and we sum up some conclusions in Section 5. Finally, Appendix A contains a few results on Chebyshev polynomials which are needed but which are otherwise not of interest here.

2. General Runge–Kutta error analysis

Let us first fix the notation and specify our assumptions on the underlying problem. We denote by $\|\cdot\|$ the usual Euclidean norm on $\mathbb{R}^d$ and by $\langle\cdot,\cdot\rangle$ the corresponding inner product $\langle u, v\rangle = v^T u$. Let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a complete probability space. For a random variable ξ on Ω, we consider the functions $f(\xi, \cdot) : \Omega \times \mathbb{R}^d \to \mathbb{R}$ and the main objective function $F : \mathbb{R}^d \to \mathbb{R}$,

$$F(w) = \mathbb{E}_\xi[f(\xi, w)]\,.$$

Here, $\mathbb{E}_\xi[\cdot]$ denotes the expectation with respect to the probability distribution of ξ. We note that we have not specified the target space of the random variable ξ, because its properties does not matter for our analysis. However, if $\omega \in \Omega$ then $\xi(\omega)$ should be interpreted as a specific selection of the problem data, in machine learning terminology known as a batch. A typical situation would be to have a finite amount of uniformly distributed data, e.g. $F(w) = \frac{1}{N}\sum_{j=1}^N f(j,w)$. Then a specific realization of $\xi(\omega)$ could be a single j, corresponding to a single data sample. Alternatively, in the common mini-batch setting, a realization of $\xi(\omega)$ could be a $B_\xi \subset \{1,\dots,N\}$, corresponding to a small subset of the data.

We approximate $\nabla F(w)$ by $g(\xi, w)$, where $g(\xi(\cdot),\cdot) : \Omega \times \mathbb{R}^d \to \mathbb{R}^d$ is integrable. In the above typical situation, we would usually have either $g(\xi,w) = \nabla f(\xi,w)$ (single sample) or $g(\xi,w) = \frac{1}{|B_\xi|}\sum_{j\in B_\xi} \nabla f(j,w)$ with $B_\xi \subset \{1,\dots,N\}$ (mini-batch). In general, we consider a sequence of jointly independent random variables $\{\xi_k\}_{k=1}^\infty$ on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with the idea that step k of the method will depend on a realization of ξ_k. For such a sequence we define the total expectation $\mathbb{E}_k[X]$ of a random variable X by

$$\mathbb{E}_k[X] = \mathbb{E}_{\xi_1}\left[\mathbb{E}_{\xi_2}\left[\dots \mathbb{E}_{\xi_{k-1}}[X]\right]\right].$$

As the variables ξ_k are jointly independent, this coincides with the expectation of X with respect to the joint probability distribution of $(\xi_1,\dots,\xi_k)$.

The following assumptions on the full problem are standard:

Assumption 1. *$F : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable and ∇F is Lipschitz continuous with Lipschitz constant $L > 0$:*

$$\|\nabla F(u) - \nabla F(v)\| \le L\|u - v\|, \ \forall u, v \in \mathbb{R}^d.$$

Assumption 2. *F is strongly convex with convexity constant $c > 0$. That is,*

$$F(u) \ge F(v) + \langle \nabla F(v), v - u\rangle + \frac{c}{2}\|v - u\|^2, \ \forall u, v \in \mathbb{R}^d.$$

We also make standard assumptions on the approximation g. The first is that it is Lipschitz-continuous with respect to the second argument:

Assumption 3. *The function g is Lipschitz continuous with respect to the second argument with (for simplicity) the same Lipschitz constant $L > 0$ as ∇F:*

$$\|g(\xi,u) - g(\xi,v)\| \le L\|u - v\|, \ a.s.\ \forall u, v \in \mathbb{R}^d.$$

Next, we assume that g is a reasonable approximation to ∇F in the following sense, following [3]:

Assumption 4. *There exist scalars $\mu_G \ge \mu > 0$, $M \ge 0$ and $M_G \ge \mu^2$ such that the gradient ∇F and its approximation g satisfy the following conditions for all $w \in \mathbb{R}^d$:*

(i) $\langle \nabla F(w), \mathbb{E}_\xi[g(\xi,w)]\rangle \ge \mu\|\nabla F(w)\|^2$,
(ii) $\|\mathbb{E}_\xi[g(\xi,w)]\| \le \mu_G\|\nabla F(w)\|$ *and*
(iii) $\mathbb{E}_\xi\left[\|g(\xi,w)\|^2\right] \le M + M_G\|\nabla F(w)\|^2$.

Assumption 4 (i) and (ii) are fulfilled by assumption with $\mu = \mu_G = 1$ if we are considering (e.g.) the single sample case $g(\xi,w) = \nabla f(\xi,w)$. The third item puts a weak limit on the variance, which means that the approximation to the gradient is not too noisy.

Remark 2.1. We note that the statements "for all $w \in \mathbb{R}^d$" in the above assumptions could be replaced by "for all w_k", where w_k are the method iterates, i.e. the assumptions only need to hold where the method is actually evaluated. However, this is not helpful in practice, since the iterates are not known a priori.

Finally, we make a general assumption on the numerical optimization method. As shown in the previous section, this will be satisfied in particular for the SRKCD method.

Assumption 5. *Given a sequence of step sizes $\{\alpha_k\}_{k\in\mathbb{N}}$ and an initial condition $w_1 \in \mathbb{R}^d$, the optimization method is of the form*

$$w_{k,i} = w_k - \alpha_k \sum_{j=1}^{i} a_{i,j} g(\xi_k, w_{k,j-1}), \quad i = 0, \dots, s-1,$$

$$w_{k+1} = w_k - \alpha_k \sum_{i=1}^{s} b_i g(\xi_k, w_{k,i-1}).$$

For brevity, denote $a_{s,j} := b_j$, $j = 1, \dots, s$. With this notation, the coefficients $a_{i,j}$ satisfy

(i) $\sum_{i=1}^{s} a_{s,i} = 1$,
(ii) $\sum_{j=1}^{i} |a_{i,j}| \leq 1, \quad i = 0, \dots, s.$

We note that item *(i)* would be satisfied for any Runge-Kutta method which is of order 1 when applied to $\dot{w} = -\nabla F(w)$.

2.1. **Preliminary results.** In the following lemma, we list some consequences of the basic assumptions.

Lemma 2.2. *Under Assumption 1 and 2, there exists a unique $w_* \in \mathbb{R}^d$ such that*

$$F(w_*) = \min_{w\in\mathbb{R}^d} F(w)$$

and $\nabla F(w_*) = 0$. Further, it follows that

$$F(u) - F(v) \leq \langle \nabla F(v), u - v \rangle + \frac{L}{2}\|u - v\|^2 \tag{2.1}$$

for all $u, v \in \mathbb{R}^d$. Finally, the difference $F(w) - F(w_*)$ is bounded by

$$2c\,(F(w) - F(w_*)) \leq \|\nabla F(w)\|^2. \tag{2.2}$$

Proof. The existence of a unique minimizer in this benign situation is well-known, see e.g. [1, Corollary 11.17]. The first inequality follows directly from a first-order expansion in Taylor series and Assumption 1. For the final inequality, see e.g. [3, Appendix B]. □

2.2. **Bound on $\|w_{k+1} - w_k\|$.** First, we consider what the method does in one step and bound $\|w_{k+1} - w_k\| = \|w_{k,s} - w_{k,0}\|$. To this end, we now define a sequence of polynomials $P_n(\alpha)$, $n = 0, \dots, s$, by

$$P_0(\alpha) = 0, \quad P_1(\alpha) = \alpha,$$

$$P_n(\alpha) = \alpha + \alpha L \sum_{i=1}^{n} |a_{n,i}| P_{i-1}(\alpha), \text{ where } 1 \leq n \leq s.$$

Note that the sequence depends on s, but for brevity we do not add an extra index to indicate this.

Lemma 2.3. *Let Assumption 3 and 5 be satisfied. Then for a fixed s, it holds that $\|w_{k,n} - w_{k,0}\| \leq P_n(\alpha_k)\|g(\xi_k, w_{k,0})\|$ for all $n \leq s$.*

Proof. We prove the statement by induction over n. In the case $n = 1$ it follows immediately from the definition that $\|w_{k,1} - w_{k,0}\| = |a_{1,1}|\alpha_k\|g(\xi_k, w_{k,0})\|$. Since $|a_{1,1}| \leq 1$ by Assumption 5 (ii), the base case is satisfied. Assume that the claim holds for all $i \leq n$ with $n < s$. Then, using Assumption 3 and the induction assumption

$$
\begin{aligned}
&\|w_{k,n+1} - w_{k,0}\| \\
&\quad = \Big\| - \alpha_k \sum_{i=1}^{n+1} a_{n+1,i} g(\xi_k, w_{k,0}) - \alpha_k \sum_{i=1}^{n+1} a_{n+1,i} \big(g(\xi_k, w_{k,i-1}) - g(\xi_k, w_{k,0})\big) \Big\| \\
&\quad \leq \alpha_k \sum_{i=1}^{n+1} |a_{n+1,i}| \|g(\xi_k, w_{k,0})\| + \alpha_k \sum_{i=1}^{n+1} |a_{n+1,i}| \|(g(\xi_k, w_{k,i-1}) - g(\xi_k, w_{k,0}))\| \\
&\quad \leq \alpha_k \sum_{i=1}^{n+1} |a_{n+1,i}| \|g(\xi_k, w_{k,0})\| + \alpha_k L \sum_{i=1}^{n+1} |a_{n+1,i}| \|w_{k,i-1} - w_{k,0}\| \\
&\quad \leq \alpha_k \sum_{i=1}^{n+1} |a_{n+1,i}| \|g(\xi_k, w_{k,0})\| + \alpha_k L \sum_{i=1}^{n+1} |a_{n+1,i}| P_{i-1}(\alpha_k) \|g(\xi_k, w_{k,0})\| \\
&\quad \leq P_{n+1}(\alpha_k) \|g(\xi_k, w_{k,0})\|,
\end{aligned}
$$

where we used Assumption 5 (ii) in the last step. This concludes the inductive step. □

Lemma 2.4. *Under Assumption 5, it holds for $2 \leq n \leq s$ that*

$$
P_n(\alpha) = \alpha + \alpha \sum_{i=1}^{n-1} (\alpha L)^i c_{n,i}
$$

where the $c_{n,i}$ are constants not depending on α or L. Further, $c_{n,i} \leq 1$ for $2 \leq n \leq s$ and $1 \leq i \leq n-1$.

Proof. Once again, we employ induction. For $n = 2$, we have

$$
P_2(\alpha) = \alpha + \alpha L(|a_{2,1}|\alpha),
$$

which is on the stated form with $c_{2,1} = |a_{2,1}|$, and by Assumption 5 (ii), $c_{2,1} \leq 1$. That is, the claim is valid for $n = 2$. Assume that P_n can be written on the stated form for all $i \leq n$ and that all the constants $c_{n,i}$ are bounded by 1. Then inserting this in the definition of P_{n+1} shows that

$$
\begin{aligned}
P_{n+1} = \alpha &+ \alpha^2 L \sum_{i=2}^{n+1} |a_{n+1,i}| + \alpha^3 L^2 \sum_{i=3}^{n+1} |a_{n+1,i}| c_{i-1,1} \\
&+ \alpha^4 L^3 \sum_{i=4}^{n+1} |a_{n+1,i}| c_{i-1,2} + \cdots + \alpha^{n+1} L^n |a_{n+1,n+1}| c_{n,n-1}.
\end{aligned}
$$

That is, we can write P_{n+1} on the desired form by taking $c_{n+1,1} = \sum_{i=2}^{n+1} |a_{n+1,i}|$ and $c_{n+1,j} = \sum_{i=j+1}^{n+1} |a_{n+1,i}| c_{i-1,j-1}$ for $j = 2, \ldots, n$. By Assumption 5 (ii),

$$
c_{n+1,1} = \sum_{i=2}^{n+1} |a_{n+1,i}| \leq \sum_{i=1}^{n+1} |a_{n+1,i}| \leq 1.
$$

Similarly, since all the $c_{i-i,j-1}$ are bounded by 1 by the induction assumption,

$$c_{n+1,j} = \sum_{i=j+1}^{n+1} |a_{n+1,i}| c_{i-1,j-1} \leq \sum_{i=1}^{n+1} |a_{n+1,i}| \leq 1.$$

for $j = 2, \dots n$. This concludes the induction step. □

We can now bound the difference $F(w_{k,s}) - F(w_{k,0})$ by using (2.1) from Lemma 2.2 to write

$$F(w_{k,s}) - F(w_{k,0}) \leq \langle \nabla F(w_{k,0}), w_{k,s} - w_{k,0} \rangle + \frac{L}{2} \|w_{k,s} - w_{k,0}\|^2.$$

For the first term on the right-hand side, we add and subtract terms to get

$$\begin{aligned}
&\langle \nabla F(w_{k,0}), w_{k,s} - w_{k,0} \rangle \\
&= \left\langle \nabla F(w_{k,0}), -\alpha_k \sum_{i=1}^{s} a_{s,i} g(\xi_k, w_{k,0}) - \alpha_k \sum_{i=1}^{s} a_{s,i} (g(\xi_k, w_{k,i-1}) - g(\xi_k, w_{k,0})) \right\rangle \\
&\leq -\alpha_k \sum_{i=1}^{s} a_{s,i} \langle \nabla F(w_{k,0}), g(\xi_k, w_{k,0}) \rangle + \alpha_k L \sum_{i=1}^{s} |a_{s,i}| \|\nabla F(w_{k,0})\| \|w_{k,i-1} - w_{k,0}\|
\end{aligned}$$

We now use Lemma 2.3 and Young's inequality $ab \leq \frac{a^2}{4} + b^2$ with $a = \alpha_k \sqrt{L} \|\nabla F(w_{k,0})\|$ and $b = \sqrt{L} \|w_{k,i-1} - w_{k,0}\|$ to bound the last sum in the previous expression

$$\begin{aligned}
&\sum_{i=1}^{s} |a_{s,i}| \alpha_k L \|\nabla F(w_{k,0})\| \|w_{k,i-1} - w_{k,0}\| \\
&\leq \frac{\alpha_k^2 L}{4} \sum_{i=1}^{s} |a_{s,i}| \|\nabla F(w_{k,0})\|^2 + L \sum_{i=1}^{s} |a_{s,i}| \|w_{k,i-1} - w_{k,0}\|^2 \\
&\leq \frac{\alpha_k^2 L}{4} \sum_{i=1}^{s} |a_{s,i}| \|\nabla F(w_{k,0})\|^2 + L \sum_{i=1}^{s} |a_{s,i}| P_{i-1}(\alpha_k)^2 \|g(\xi_k, w_{k,0})\|^2.
\end{aligned}$$

In total, we get (using Lemma 2.3 again)

$$\begin{aligned}
&F(w_{k,s}) - F(w_{k,0}) \\
&\quad \leq -\alpha_k \sum_{i=1}^{s} a_{s,i} \langle \nabla F(w_{k,0}), g(\xi_k, w_{k,0}) \rangle + \frac{\alpha_k^2 L}{4} \sum_{i=1}^{s} |a_{s,i}| \|\nabla F(w_{k,0})\|^2 \\
&\qquad + L \sum_{i=1}^{s} |a_{s,i}| P_{i-1}(\alpha_k)^2 \|g(\xi_k, w_{k,0})\|^2 + \frac{L}{2} P_s(\alpha_k)^2 \|g(\xi_k, w_{k,0})\|^2.
\end{aligned}$$

Taking expectations with respect to the distribution of ξ_k (recall that $w_{k,0}$ doesn't depend on ξ_k) leads to

$$\begin{aligned}
&\mathbb{E}_{\xi_k}[F(w_{k,s}) - F(w_{k,0})] \\
&\quad \leq -\alpha_k \sum_{i=1}^{s} a_{s,i} \langle \nabla F(w_{k,0}), \mathbb{E}_{\xi_k}[g(\xi_k, w_{k,0})] \rangle + \frac{\alpha_k^2 L}{4} \sum_{i=1}^{s} |a_{s,i}| \|\nabla F(w_{k,0})\|^2 \\
&\qquad + L \left(\sum_{i=1}^{s} |a_{s,i}| P_{i-1}(\alpha_k)^2 + \frac{1}{2} P_s(\alpha_k)^2 \right) \mathbb{E}_{\xi_k} \left[\|g(\xi_k, w_{k,0})\|^2 \right].
\end{aligned} \tag{2.3}$$

By Assumption 4 we have that

$$\mathbb{E}_{\xi_k} \left[\|g(\xi_k, w_{k,0})\|^2 \right] \leq M + M_G \|\nabla F(w_{k,0})\|^2,$$

and applying this to the last term of (2.3) gives

$$\begin{aligned}
&\mathbb{E}_{\xi_k}[F(w_{k,s}) - F(w_{k,0})] \\
&\qquad \leq -\alpha_k \mu \|\nabla F(w_{k,0})\|^2 + \frac{\alpha_k^2 L}{4} \sum_{i=1}^{s} |a_{s,i}| \|\nabla F(w_{k,0})\|^2 \\
&\qquad + L \left(\sum_{i=1}^{s} |a_{s,i}| P_{i-1}(\alpha_k)^2 + \frac{1}{2} P_s(\alpha_k)^2 \right) \left(M + M_G \|\nabla F(w_{k,0})\|^2 \right).
\end{aligned} \tag{2.4}$$

Here we have also used Assumption 4 (i) and Assumption 5 (i) on the first term on the right-hand side of (2.3) to obtain the $-\alpha_k \mu \|\nabla F(w_{k,0})\|^2$-term in (2.4). Re-ordering the terms, we find

$$\begin{aligned}
&\mathbb{E}_{\xi_k}[F(w_{k,s})] - F(w_{k,0}) \\
&\qquad \leq Q(\alpha_k) \|\nabla F(w_{k,0})\|^2 + L\Big(\sum_{i=1}^{s} |a_{s,i}| P_{i-1}(\alpha_k)^2 + \frac{1}{2} P_s(\alpha_k)^2 \Big) M.
\end{aligned} \tag{2.5}$$

with

$$Q(\alpha_k) = -\alpha_k \mu + L M_G \sum_{i=1}^{s} |a_{s,i}| P_{i-1}(\alpha_k)^2 + \frac{L M_G}{2} P_s(\alpha_k)^2 + \frac{1}{4} \alpha_k^2 L \sum_{i=1}^{s} |a_{s,i}|.$$

Since $P_0(\alpha_k) = 0$ and the smallest power of α_k in $P_i(\alpha_k)^2$ for $i = 1, \dots, s$ is α_k^2, we can choose $\alpha_k > 0$ small enough that

$$Q(\alpha_k) < -\frac{\alpha_k \mu}{2}. \tag{2.6}$$

This means that the first term in (2.5) is negative, and we can estimate it by using the strong convexity property

$$-\|\nabla F(w_{k,0})\|^2 \leq -2c\big(F(w_{k,0}) - F(w_*)\big)$$

from (2.2) in Lemma 2.2. Adding and subtracting $F(w_*)$, rearranging and taking total expectations on both sides thus leads to

$$\begin{aligned}
\mathbb{E}_k[F(w_{k+1}) - F(w_*)] &\leq (1 - \alpha_k \mu c)\, \mathbb{E}_k[F(w_k) - F(w_*)] \\
&\quad + \Big(L\alpha_k^2 + \frac{L}{2} P_s(\alpha_k)^2 \Big) M + \frac{L}{4} \Big(\sum_{i=1}^{s} |a_{s,i}| P_{i-1}(\alpha_k) \Big)^2.
\end{aligned} \tag{2.7}$$

This means that the next error is the previous error multiplied by a factor which is strictly less than one, plus two terms that are small. Hence it will tend to zero as $k \to \infty$, as we show formally in the next section.

Remark 2.5. Let us elaborate on the choice of α_k in (2.6). We can make the choice because the negative term is multiplied with α_k while the positive terms are all multiplied with higher powers of α_k, meaning that for a sufficiently small α_k the negative term will dominate. To make this more concrete, suppose that $\alpha_k \leq \frac{1}{Lm}$ for an integer $m \geq 2$. Then by Lemma 2.4,

$$P_i(\alpha_k)^2 \leq \alpha_k^2 \Big(1 + \frac{1}{m} + \frac{1}{m^2} + \cdots + \frac{1}{m^{s-1}} \Big)^2 = \alpha_k^2 \frac{m^2}{(m-1)^2} \leq 4\alpha_k^2.$$

for every $i = 1, \dots, s$. Thus, since $\sum_{i=1}^{s} |a_{s,i}| \leq 1$ by Assumption 5,

$$\begin{aligned} Q(\alpha_k) &\leq -\alpha_k \mu + \alpha_k^2 \Big(4LM_G + 4\frac{LM_G}{2} + \frac{L}{4} \Big) \\ &\leq -\alpha_k \mu + L\alpha_k^2 (6M_G + \frac{1}{4}) \\ &\leq -\alpha_k \mu + \alpha_k \frac{6M_G + \frac{1}{4}}{m}. \end{aligned}$$

This is bounded by $-\frac{\alpha_k \mu}{2}$ and thereby satisfies (2.6) if

$$m \geq \frac{12M_G + \frac{1}{2}}{\mu}.$$

We can guarantee this by choosing m large enough, and a moderately small m is sufficient unless the estimator of the gradient is very bad (small μ) or the variance of the data is very large (large M_G). In a typical situation, both of these constants can be set to 1, which leads to a step size restriction of $\alpha_k \leq \frac{2}{25L}$. We note that this argument could be further refined to improve the bound, since the current estimations of $P_i(\alpha_k)^2$ are quite crude. For example, clearly $P_1(\alpha_k)^2 = \alpha_k^2$.

2.3. **Convergence proof.**

Theorem 2.6. *Let Assumptions 1–5 be satisfied. Further assume that the scheme is run with the step size $\alpha_k = \frac{\beta}{k+\gamma}$, where $\gamma > 0$, $\beta > \frac{1}{c\mu}$ and α_1 satisfies (2.6). Then with*

$$\nu = \max \left\{ \frac{\Big(\sum_{i=1}^{s} |a_{s,i}| P_{i-1}(\beta)^2 + \frac{L}{2} P_s(\beta)^2 \Big) M}{\beta\mu c - 1}, (\gamma + 1)\left(F(w_1) - F(w_*)\right) \right\},$$

it holds that

$$\mathbb{E}_k[F(w_k) - F(w_*)] \leq \frac{\nu}{k + \gamma}, \tag{2.8}$$

for $k = 1, 2, \dots$.

Remark 2.7. The error constant ν can be bounded by a constant which is independent of s by using Assumption 5 (ii). However, for some methods $a_{s,i}$ decreases rapidly with increasing i (such as the SRKCD methods). In that case, such an estimation would be rather crude. We therefore keep these terms in the statement and leave it to the reader to insert their specific coefficients.

Proof of Theorem 2.6. We prove this using induction, inspired by [3, Theorem 4.7]. Let us abbreviate $\hat{k} = k + \gamma$. For the base case we note that it follows from the definition of ν that

$$\mathbb{E}_k[F(w_1) - F(w_*)] = (\gamma + 1) \frac{F(w_1) - F(w_*)}{\gamma + 1} \leq \frac{\nu}{\gamma + 1},$$

since w_1 is not chosen randomly. For the induction step we assume that (2.8) holds for some k. Using (2.7) we then have

$$\begin{aligned} \mathbb{E}_k[F(w_{k+1}) - F(w_*)] &\leq (1 - \alpha_k \mu c) \frac{\nu}{\hat{k}} \\ &\quad + \Big(\sum_{i=1}^{s} |a_{s,i}| P_{i-1}(\alpha_k)^2 + \frac{L}{2} P_s(\alpha_k)^2 \Big) M. \end{aligned} \tag{2.9}$$

Using that $\alpha_k = \frac{\beta}{\hat{k}}$ and adding and subtracting $\frac{\nu}{\hat{k}^2}$, we find that the right-hand side of (2.9) equals $S_1 + S_2$ where

$$S_1 = \left(\frac{\hat{k}-1}{\hat{k}^2}\right)\nu \quad \text{and} \quad S_2 = -\left(\frac{\beta\mu c - 1}{\hat{k}^2}\right)\nu + \left(\sum_{i=1}^{s} |a_{s,i}| P_{i-1}\left(\frac{\beta}{\hat{k}}\right)^2 + \frac{L}{2} P_s\left(\frac{\beta}{\hat{k}}\right)^2\right) M.$$

By the inequality $\hat{k}^2 \geq \big(\hat{k}-1\big)\big(\hat{k}+1\big)$ we directly have that

$$S_1 \leq \frac{\nu}{\hat{k}+1}.$$

To bound S_2, we first note that the polynomials $\frac{P_i(\alpha)}{\alpha}$ are increasing on the positive real axis since all the coefficients of $P_i(\alpha)$ are non-negative. It thus holds that

$$\hat{k} P_i\left(\frac{\beta}{\hat{k}}\right) \leq P_i(\beta).$$

By the definition of ν, this yields

$$\left(\sum_{i=1}^{s} |a_{s,i}| P_{i-1}\left(\frac{\beta}{\hat{k}}\right)^2 + \frac{L}{2} P_s\left(\frac{\beta}{\hat{k}}\right)^2\right) M \leq \left(\frac{\beta\mu c - 1}{\hat{k}^2}\right)\nu.$$

Thus $S_2 \leq 0$. In conclusion, $S_1 + S_2 \leq \frac{\nu}{\hat{k}+1}$, so the bound (2.8) holds for all $k \geq 1$. □

2.4. **Nonconvex setting.** Without any convexity assumption, it is typically impossible to prove convergence with a certain speed. But we may still prove convergence. The following section is an adaptation of similar arguments in [3] to the Runge-Kutta setting. Since we do not know a priori that there is a unique minimum w_* or even a lower bound on F, we make the following assumption:

Assumption 6. *The sequence of iterates $\{w_k\}_{k\in\mathbb{N}}$ is contained in an open set over which F is bounded from below by F_{inf}.*

Theorem 2.8. *Let Assumption 1 and Assumptions 3–6 be satisfied. Further, let the step sizes $\alpha_k = \frac{\beta}{k+\gamma}$ be given, where $\gamma > 0$, $\beta > \frac{1}{c\mu}$ and α_1 satisfies (2.6). Then the following bound holds:*

$$\lim_{K\to\infty} \frac{1}{A_K} \sum_{k=1}^{K} \alpha_k \mathbb{E}_k\big[\|\nabla F(w_k)\|^2\big] = 0,$$

where $A_K = \sum_{k=1}^{K} \alpha_k$.

Remark 2.9. This means that $\liminf_{k\to\infty} \mathbb{E}_k\big[\|\nabla F(w_k)\|^2\big] = 0$, i.e. w_k tends to a (local) minimum of F in a weak sense. But we do not get any further information on how fast this convergence is.

Proof of Theorem 2.8. If α_1 satisfies (2.6) then so does every α_k, $k \geq 1$, and by taking total expectations in (2.5) we find that

$$\begin{aligned}\mathbb{E}_k[F(w_{k+1})] - \mathbb{E}_k[F(w_k)] \leq &-\frac{1}{2}\alpha_k \mu \mathbb{E}_k\big[\|\nabla F(w_{k,0})\|^2\big] \\ &+ \Big(\sum_{i=1}^{s} |a_{s,i}| P_{i-1}(\alpha_k)^2 + \frac{L}{2} P_s(\alpha_k)^2\Big) M\end{aligned}$$

By the independece of the $\{\xi_k\}_{k=1}^{\infty}$ and the fact that w_k is independent of ξ_K for $K > k$ we have that $\mathbb{E}_K[F(w_k)] = \mathbb{E}_k[F(w_k)]$ for $K \geq k$. Using this, we obtain a

telescopic sum on the left-hand side when we sum over K terms. Along with the fact that

$$F_{\inf} - \mathbb{E}_K[F(w_1)] \leq \mathbb{E}_K[F(w_{K+1})] - \mathbb{E}_K[F(w_1)]$$

and rearranging the terms we thus get

$$\begin{aligned} \frac{1}{2}\mu \sum_{k=1}^{K} \alpha_k \mathbb{E}_K\left[\|\nabla F(w_k)\|^2\right] &\leq \mathbb{E}_K[F(w_1)] - F_{\inf} \\ &\quad + \sum_{k=1}^{K} \Big(\sum_{i=1}^{s} |a_{s,i}| P_{i-1}(\alpha_k)^2 + \frac{L}{2} P_s(\alpha_k)^2 \Big) M. \end{aligned} \tag{2.10}$$

By assumption, we have $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$, which means that also $\sum_{k=1}^{\infty} \alpha_k^i < \infty$ for any integer $i > 2$. But $P_j(\alpha)$ is a polynomial in α of degree j without a constant term, see e.g. Lemma 2.4. Hence

$$P_j(\alpha_k)^2 = \sum_{i=2}^{2j} C_i \alpha_k^i,$$

where C_i are certain constants. This immediately shows that the terms on the second line of (2.10) are finite, and thus we can conclude that

$$\lim_{K\to\infty} \sum_{k=1}^{K} \alpha_k \mathbb{E}_k\left[\|\nabla F(w_k)\|^2\right] < \infty.$$

By assumption, $\sum_{k=1}^{\infty} \alpha_k = \infty$, and (recalling $A_K = \sum_{k=1}^{K} \alpha_k$) hence

$$\lim_{K\to\infty} \frac{1}{A_K} \mathbb{E}_K \left[\sum_{k=1}^{K} \alpha_k \|\nabla F(w_k)\|^2 \right] = 0.$$

□

We may replace the lim inf in Remark 2.9 by a strong limit, if we also assume that F is twice differentiable. We state this result for completeness, but omit the proof since it is very similar to that of [3, Corollary 4.12].

Theorem 2.10. *Let Assumption 1 and Assumptions 3–6 be satisfied, and also assume that F is twice differentiable. Given the step sizes $\alpha_k = \frac{\beta}{k+\gamma}$, where $\gamma > 0$, $\beta > \frac{1}{c\mu}$ and α_1 satisfies (2.6), it follows that*

$$\lim_{k\to\infty} \mathbb{E}_k\left[\|\nabla F(w_k)\|^2\right] = 0.$$

3. Specific SRKCD analysis

The first-order RKC method with s stages applied to the gradient flow $\dot{w} = -\nabla F(w)$ with constant time step α is defined by

$$\begin{aligned} w_{k,0} &= w_k, \\ w_{k,1} &= w_k - \tilde{\mu}_1 \alpha \nabla F(w_{k,0}), \\ w_{k,j} &= (1-\nu_j) w_{k,j-1} + \nu_j w_{k,j-2} - \tilde{\mu}_j \alpha \nabla F(w_{k,j-1}), \quad j = 2, \dots, s, \\ w_{k+1} &= w_{k,s}, \end{aligned} \tag{3.1}$$

see e.g. [10, Section V.1]. Here, $w_{k,j}$ denotes the $(j+1)$st internal stage, and $\nabla F(w_{k,j})$ is the corresponding stage derivative. The scalars $\tilde{\mu}_j$ and ν_j are the method-specific coefficients. They are defined via Chebyshev polynomials T_j as

$$\tilde{\mu}_1 = \frac{\omega_1}{T_1(\omega_0)}, \ \tilde{\mu}_j = \frac{2\omega_1 T_{j-1}(\omega_0)}{T_j(\omega_0)} \text{ and } \nu_j = -\frac{T_{j-2}(\omega_0)}{T_j(\omega_0)}$$

where $\omega_0 = 1 + \frac{\epsilon}{s^2}$ and $\omega_1 = \frac{T_s(\omega_0)}{T_s'(\omega_0)}$. There is thus a single design parameter, ω_0, which is given in terms of ϵ. Setting $\epsilon = 0$ results in the original, un-damped, RKC methods. Instead setting $\epsilon > 0$ introduces extra numerical damping and makes sure that the stability region never degenerates into a single point on the negative real axis. In our numerical experiments, we use the value $\epsilon = 0.01$. We note that we write $\tilde{\mu}_j$ rather than simply μ_j to be consistent with [10], where μ_j would be the quantity $1 - \nu_j$ and an extra term $(1 - \mu_j - \nu_j)w_k$ appears. In our first-order setting, $\mu_j + \nu_j = 1$, and this term cancels. Similarly, the variables ω_0 and ω_1 indicate scalars and should not be confused with elements of the probability space Ω.

Approximating the gradient $\nabla F(w_k)$ by $g(\xi_k, w_k)$ in step k and using the step size α_k now gives us the method we call SRKCD:

$$\begin{aligned} w_{k,0} &= w_k, \\ w_{k,1} &= w_k - \tilde{\mu}_1 \alpha_k g(\xi_k, w_k)), \\ w_{k,j} &= (1-\nu_j) w_{k,j-1} + \nu_j w_{k,j-2} - \tilde{\mu}_j \alpha_k g(\xi_k, w_{k,j-1}), \quad j = 2, \dots, s, \\ w_{k+1} &= w_{k,s}. \end{aligned} \tag{3.2}$$

The method is formulated as a three-term recursion in order to preserve its stability properties under round-off error perturbations. This is similar to how computing the Chebyshev polynomials directly in a naive way quickly leads to a complete loss of precision, whereas evaluating them via a three-term recursion is backwards stable. In order to apply the analysis in the previous section, however, we need to state the method on the standard Runge-Kutta form. This, and verifying Assumption 5, is what the rest of the section is concerned with. Since the SRKCD method has precisely the same coefficients as the RKC method for the full problem $\dot{w} = -\nabla F(w)$, we will consider the RKC formulation for brevity. We will also dispense with the subscript k in α_k, since the varying step size does not matter for the reformulation.

We start by noting that by Lemmas A.1 and A.2 (in the appendix), both $T_s(\omega_0)$ and $T_s'(\omega_0)$ are positive for $s \geq 1$. Hence, $\omega_1 > 0$. Lemma A.1 also shows that $T_j(\omega_0) \geq 1$ for any j, which directly implies that $\tilde{\mu}_1 > 0$, $\tilde{\mu}_j > 0$ and $\nu_j < 0$ for every $j \in \mathbb{N}$. We collect these inequalities in a lemma for later reference:

Lemma 3.1. *With $\omega_0 = 1 + \frac{\epsilon}{s^2}$ chosen as above with $\epsilon \geq 0$, it holds for every $j \in \mathbb{N}$ that $\tilde{\mu}_1 > 0$, $\tilde{\mu}_j > 0$ and $\nu_j < 0$.*

3.1. **One-stage update.** We first derive an alternative expression for the update $w_{k,j} - w_{k,j-1}$, i.e. what happens from one stage to the next.

Lemma 3.2. *The iterates defined by (3.1) satisfy*

$$w_{k,j} - w_{k,j-1} = -\alpha \sum_{i=1}^{j} (-1)^{j+i} \left(\prod_{\ell=i+1}^{j} \nu_\ell \right) \tilde{\mu}_i \nabla F(w_{k,i-1}) \tag{3.3}$$

for $j = 2, .., s$.

Proof. The proof is by induction. For the base case $j = 1$, we have using (3.3) that

$$w_{k,1} - w_{k,0} = -\alpha \nabla F(w_{k,0}),$$

which corresponds to the first update of (3.1). Assume that the identity holds for some j with $2 \le j \le s-1$. According to (3.1), we then have

$$w_{k,j+1} - w_{k,j} = -\nu_{j+1}(w_{k,j} - w_{k,j-1}) - \tilde{\mu}_{j+1}\alpha \nabla F(w_{k,j}).$$

We plug in (3.3) instead of $w_{k,j} - w_{k,j-1}$ and find that the right-hand-side equals

$$-\nu_{j+1}\Bigg(-\alpha \sum_{i=1}^{j} (-1)^{j+i} \Bigg(\prod_{\ell=i+1}^{j} \nu_\ell \Bigg) \tilde{\mu}_i \nabla F(w_{k,i-1}) \Bigg) - \tilde{\mu}_{j+1}\alpha \nabla F(w_{k,j}).$$

Because the product does not depend on i, we can move the ν_{j+1} into it. We can also extend the sum to incorporate the final gradient term, since $i = j+1$ makes the product equal 1. This leaves us with

$$\begin{aligned} w_{k,j+1} - w_{k,j} &= -\alpha \sum_{i=1}^{j} (-1)^{i+j+1} \Bigg(\prod_{\ell=i+1}^{j+1} \nu_\ell \Bigg) \tilde{\mu}_i \nabla F(w_{k,i-1}) - \tilde{\mu}_{j+1}\alpha \nabla F(w_{k,j}) \\ &= -\alpha \sum_{i=1}^{j+1} (-1)^{i+j+1} \Bigg(\prod_{\ell=i+1}^{j+1} \nu_\ell \Bigg) \tilde{\mu}_i \nabla F(w_{k,i-1}). \end{aligned}$$

The identity (3.3) thus holds also for $j+1$ and the proof is complete. □

3.2. **Full update.** Next, we consider the "full" stage updates $w_{k,n} - w_{k,0}$.

Lemma 3.3. *For $1 \le n \le s$, the iterates of the RKC method (3.1) satisfy*

$$w_{k,n} = w_{k,0} - \alpha \sum_{i=1}^{n} a_{n,i} \nabla F(w_{k,i-1}),$$

where

$$a_{n,i} = \sum_{j=i}^{n} (-1)^{j+i} \Bigg(\prod_{\ell=i+1}^{j} \nu_\ell \Bigg) \tilde{\mu}_i. \tag{3.4}$$

In particular,

$$w_{k+1} = w_k - \alpha \sum_{i=1}^{s} a_{s,i} \nabla F(w_{k,i-1}).$$

Additionally, every $a_{n,i} > 0$.

Proof. The particular form of $w_{k,n}$ follows from (3.3) in the preceeding section since

$$w_{k,n} - w_{k,0} = \sum_{j=1}^{n} w_{k,j} - w_{k,j-1} = -\alpha_k \sum_{j=1}^{n} \sum_{i=1}^{j} (-1)^{j+i} \Bigg(\prod_{\ell=i+1}^{j} \nu_\ell \Bigg) \tilde{\mu}_i \nabla F(w_{k,i-1}).$$

Interchanging the order of summation gives

$$w_{k,n} - w_{k,0} = -\alpha_k \sum_{i=1}^{n} \Bigg(\sum_{j=i}^{n} (-1)^{j+i} \Bigg(\prod_{\ell=i+1}^{j} \nu_\ell \Bigg) \tilde{\mu}_i \Bigg) \nabla F(w_{k,i-1}),$$

where we recognize the coefficients $a_{n,i}$. The expression for w_{k+1} follows by setting $n = s$.

For the final assertion, we note that each of the terms

$$(-1)^{j+i}\left(\prod_{\ell=i+1}^{j}\nu_\ell\right)\tilde{\mu}_i$$

in the sum (3.4) is positive, since it is the product of $2j$ negative factors: $j+i$ from $(-1)^{j+i}$ and $j-i$ from the product. Since it is a sum of positive terms, the coefficient $a_{n,i}$ is therefore also positive. □

3.3. **Convergence.** We can now transfer these properties to the SRKCD method and prove that it converges.

Lemma 3.4. *The SRKCD method (3.2) satisfies Assumption 5.*

Proof. The methods (3.1) and (3.2) share the same coefficients. By recalling that $w_{k,0} = w_k$ and replacing ∇F with $g(\xi_k, \cdot)$, Lemma 3.3 proves that the method is given on the desired form.

One of the basic Runge-Kutta order conditions requires that $\sum_{i=1}^{s} b_i = 1$. This can be easily verified by inserting the exact solution into the scheme and expanding in Taylor series, see e.g. [8, Section II.1]. Since the corresponding RKC methods are designed to be of order 1 regardless of which s is chosen, part *(i)* of Assumption 5 is fulfilled.

For part *(ii)*, we note that by (3.4) in Lemma 3.3 we have

$$\begin{aligned}\sum_{i=1}^{n+1} a_{n+1,i} &= \sum_{i=1}^{n+1}\sum_{j=i}^{n+1}(-1)^{j+i}\left(\prod_{\ell=i+1}^{j}\nu_\ell\right)\tilde{\mu}_i \\ &= \sum_{i=1}^{n}\sum_{j=i}^{n+1}(-1)^{j+i}\left(\prod_{\ell=i+1}^{j}\nu_\ell\right)\tilde{\mu}_i + \tilde{\mu}_{n+1} \\ &= \sum_{i=1}^{n} a_{n,i} + \sum_{i=1}^{n}(-1)^{n+1+i}\left(\prod_{\ell=i+1}^{n+1}\nu_l\right)\tilde{\mu}_i + \tilde{\mu}_{n+1}.\end{aligned}$$

By Lemma 3.1, the $\tilde{\mu}_i$-terms are positive, while the ν_l-terms are negative. Each of the terms in the middle sum is thus the product of an even number of negative factors and is therefore positive. From this fact, we conclude that $\sum_{i=1}^{n+1} a_{n+1,i} > \sum_{i=1}^{n} a_{n,i}$. Since the coefficients $a_{n,i}$ are positive by Lemma 3.3 we immediately get also $\sum_{i=1}^{n+1} |a_{n+1,i}| > \sum_{i=1}^{n} |a_{n,i}|$. The sum $\sum_{i=1}^{n} |a_{n,i}|$ is thus strictly increasing with n, and bounded from above by $\sum_{i=1}^{s} |a_{s,i}| = \sum_{i=1}^{s} a_{s,i} = 1$. □

Corollary 3.5. *If Assumptions 1–4 are satisfied, then SRKCD converges as stated in Theorem 2.6. If instead Assumptions 1, 3, 4 and 6 are satisfied, SRKCD converges as stated in Theorem 2.8.*

Proof. By Lemma 3.4, Assumption 5 is satisfied. We can therefore apply either Theorem 2.6 or Theorem 2.8. □

3.4. **Linearization.** We note that Corollary 3.5 does not use the properties of the scheme that makes it an RKC-type method. This is both because we apply it to a nonlinear problem, and because of the stochastic modification. In the rest of this subsection, we will elaborate on this matter.

Consider the full, nonlinear problem $\dot{w} = -\nabla F(w)$ and suppose that F is twice continuously differentiable. Let $z(t)$ be a second, arbitrary solution with

$\dot{z} = -\nabla F(z)$, such that $w(t) = z(t) + y(t)$. A linearization around z is then

$$\dot{y} = -\nabla^2 F(z(t))y, \tag{3.5}$$

where $\nabla^2 F(z(t))$ is the Hessian at $z(t)$. If we further take an equilibrium solution $z(t) \equiv w_*$, we get an autonomous linear initial value problem $\dot{y} = Ay = -\nabla^2 F(w_*)y$. Under Assumption 2, the matrix A has negative eigenvalues, which means that the exact solution $y(t)$ tends to zero as t grows.

If we now apply a Runge-Kutta method and approximate $y(t_k)$ by y_k, then the stability of the scheme is governed by the eigenvalues of A. This is easily seen by diagonalizing A and doing a change of variables. In particular, if R is the stability function of the Runge-Kutta scheme and α_k is the temporal step size, then

$$|R(\alpha_k \lambda_j)| \leq 1$$

should hold for every eigenvalue λ_j of A. With strict inequality, we don't only have stability but that y_k tends to zero just like the exact solution. By considering the situtation in somewhat more detail, one can prove that in fact

$$G(y_{k+1}) - G(0) \leq \max_j R(\alpha_k \lambda_j)^2 \big(G(y_k) - G(0)\big),$$

where $G(y) = y^T \nabla^2 F(w_*) y$ with the minimum $y_* = 0$. This is [5, Proposition 1], which considers the (slightly) more general situation $G(y) = y^T A y - b^T y$ with a constant vector b.

We can now utilize information on the stability functions R. For gradient descent, corresponding to the explicit Euler method, stability is guaranteed for step sizes α_k such that $|1 + \alpha_k \lambda_j| \leq 1$ for all j, which implies that $\alpha_k \leq \min_j \frac{-2}{\lambda_j}$. The RKC methods, on the other hand, are constructed such that their stability regions $\{z \in \mathbb{C} \mid |R(z)| \leq 1\}$ cover as much as possible of the negative real line. With s stages, the stability limit will instead be roughly* $\alpha_k \leq \min_j \frac{-2s^2}{\lambda_j}$, which allows much larger steps than for normal gradient descent. If the linearized system (3.5) is a reasonably good approximation of the full nonlinear problem $\dot{w} = -\nabla F(w)$, then we can expect the same behaviour when applying the methods to the full problem.

If we instead apply SGD to the linearized system, we get the iteration

$$\begin{aligned} y_{k+1} &= y_k - \alpha_k \nabla g(\xi_k, w_*) y_k \\ &= \prod_{i=1}^{k} \Big(I - \alpha_k \nabla g(\xi_i, w_*) \Big) y_1. \end{aligned}$$

This indicates that the scheme would be stable if $\|I - \alpha_k \nabla g(\xi_i, w_*)\| \leq 1$ for every i, i.e. $|1 + \alpha_k \lambda_j^i| \leq 1$ for all i and j, where λ_j^i now denotes the eigenvalues of the matrix $\nabla g(\xi_i, w_*)$. Similarly, for SRKCD we get the stability condition $|R(\alpha_k \lambda_j^i)| \leq 1$ for all i and j, which allows a step size which is roughly s^2 larger.

However, in practice this condition is likely both too restrictive and impractical. It is too restrictive because the maximal eigenvalues $\lambda_{\max}^i = \max_j \lambda_j^i$ typically vary significantly with i, see Figure 1 for an example. The likelihood that the corresponding "worst" $\nabla g(\xi_i, w_*)$ is chosen often enough to be the dominating factor in terms of stability is very small. That is, with high probability, many of the steps could be significantly larger without issue. It is impractical, because there

*The exact value depends on the damping parameter ϵ. For small ϵ it is approximately $\min_j \frac{-(2-4/3\epsilon)s^2}{\lambda_j}$, see [10, Section V.1].

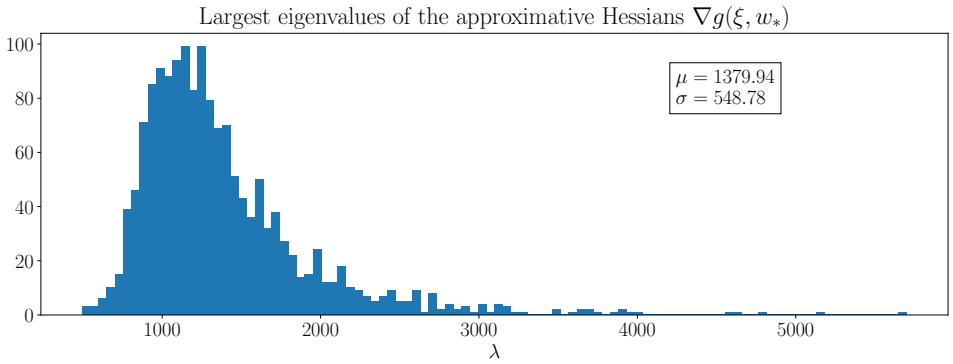

FIGURE 1. Here we see the distribution of the largest eigenvalues of $\nabla g(\xi_i, w_*)$ for an optimization problem arising from using a convolutional neural network for image classification. The data set with 60000 images is split into non-overlapping batches of 32 images each, and each ξ_i corresponds to one such batch. Each bar indicates how many such batches have a maximal eigenvalue in the specific range. The mean is $\mu = 1379.94$ and the standard deviation $\sigma = 548.78$.

is no clear relation between the eigenvalues of $\nabla g(\xi_i, w_*)$ and those of $\nabla^2 F(w_*)$, meaning that any known overall statistics about the data cannot be used. Further, there is no way to a priori find out which $g(\xi_i, \cdot)$ will be chosen such that the above issue could be alleviated.

For these reasons, we find it unlikely that one could find a proof of convergence of SRKCD with a stability condition that is reasonably sharp and illustrates the benefit of the scheme. Nevertheless, since the RKC methods have stability regions that are roughly s^2 times larger than that of the explicit Euler method, we expect to be able to take roughly s^2 times larger steps with SRKCD instead of SGD.

4. Numerical experiments

In order to investigate the stability properties of the SRKCD method in practice, we have performed numerical experiments on a simple academic test example and on a more complex optimization problem arising in a supervised learning applications. The different setups are described in the following subsections.

We have implemented the method in Tensorflow with Keras by observing that (3.2) can be alternatively expressed as SGD with a very specific momentum term that changes with each stage, and where the same batch of data is used in s consecutive steps. The same idea could equally well be applied in other common machine learning frameworks such as PyTorch. However, we note that it is only valid for relatively small values of s; for large s the three-term recursion (3.2) is needed to avoid catastrophic round-off error accumulation. We write the momentum equations as

$$\begin{aligned} v_{k,j} &= \eta_j v_{k,j-1} - \ell_j g(\xi_k, w_{k,j-1}), \\ w_{k,j} &= w_{k,j-1} + v_{k,j}, \end{aligned} \tag{4.1}$$

i.e. $w_{k,j} - w_{k,j-1} = v_{k,j}$. But according to (3.2) we have

$$w_{k,j} - w_{k,j-1} = -\nu_j(w_{k,j-1} - w_{k,j-2}) - \tilde{\mu}_j g(\xi_k, w_{k,j-1}),$$

so we see that the two formulations (4.1) and (3.2) are equivalent if we set

$$\eta_j = \begin{cases} -\nu_j, & 2 \le s, \\ 0, & j = 1, \end{cases} \qquad \text{and} \qquad \ell_j = \tilde{\mu}_j \alpha_k.$$

We will only investigate stability properties in this paper, rather than convergence or efficiency. That is, we will not run necessarily run the methods until we reach a local minimum but rather stop them after a predetermined number of iterations. We do this for two reasons. First, because it is clear also from these tests that the methods converge (in expectation) whenever we have stability, like for e.g. SGD. Secondly, because a proper efficiency comparison would require another paper. Not only because of the number of potential alternative methods and the need to ensure comparably optimized implementations, but also because the optimal choice of step size is intricate. Simply maximizing the step size is not always desirable, as we demonstrate in the next subsection.

Our analysis proves convergence for a step size α_k that decreases with k. In these experiments, however, we will use a fixed step size α, since we only investigate the first phase of the optimization process. The decreasing step size is only needed to cancel the noise arising from the stochastic approximation as we approach the minimum.

4.1. **Small-scale linear convex problem.** In the first experiment, we consider the cost functional

$$F(w) = \frac{1}{N} \sum_{i=1}^{N} f(w, x^i) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{d} \frac{(x_j^i)^2 w_j^2}{d},$$

where $d \in \mathbb{N}$ and $w \in \mathbb{R}^d$ are the optimization parameters and each $x^i \in \mathbb{R}^d$ is a known data vector. We take $N = 1000$ and $d = 50$. The vectors x^i were sampled randomly from normal distributions with standard deviation 1 and means $1 + \frac{10i}{d}$. This means that

$$\nabla F(w) = Aw,$$

where A is a diagonal matrix with the diagonal entries

$$\lambda_j = A_{j,j} = \frac{2}{Nd} \sum_{i=1}^{N} (x_j^i)^2.$$

We note that $\{\lambda_j\}_{j=1}^d$ are also the eigenvalues of A. The system is diagonal by design for simplicity, but any system $\dot{w} = Aw$ with a diagonalizable matrix A can be transformed into this form with the eigenvalues preserved. Thus this choice implies no loss of generality.

Since the system is diagonal, stability is determined by the eigenvalues as discussed in Section 3.4. We define $\lambda_{\min} = \min_j \lambda_j$ and $\lambda_{\max} = \max_j \lambda_j$. Then if we use ∇F instead of stochastic approximations $g(\xi, \cdot)$, for stability we must have $\alpha_k \le 2/L$. Further, the optimal step size which minimizes $\max_j |R(h\lambda_j)|$ is $\alpha = \frac{2}{\lambda_{\min} + \lambda_{\max}}$, see e.g. [5].

With our particular choice of data, one realization resulted in $\lambda_{\min} = 0.0791$ and $\lambda_{\max} = 4.704$. For these values, we ran 15 iterations of GD and 3 epochs of SGD with a batch size of 32 and with different step sizes between 0 and $2/L = 0.4251$. The final values $F(w)$ are plotted in Figure 2. For GD, we can clearly observe the optimal step size choice $\frac{2}{\lambda_{\min} + \lambda_{\max}} = 0.4181$. Closer to $\alpha = 2/L$, the values

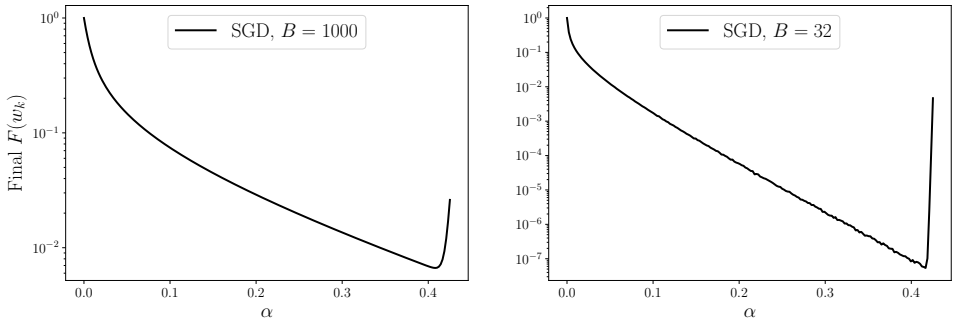

FIGURE 2. SGD with batch size 1000, i.e. GD, (left) and SGD with batch size 32 (right) when applied to the problem described in Section 4.1. Note the different scales on the y-axes and that different number of iterations were used.

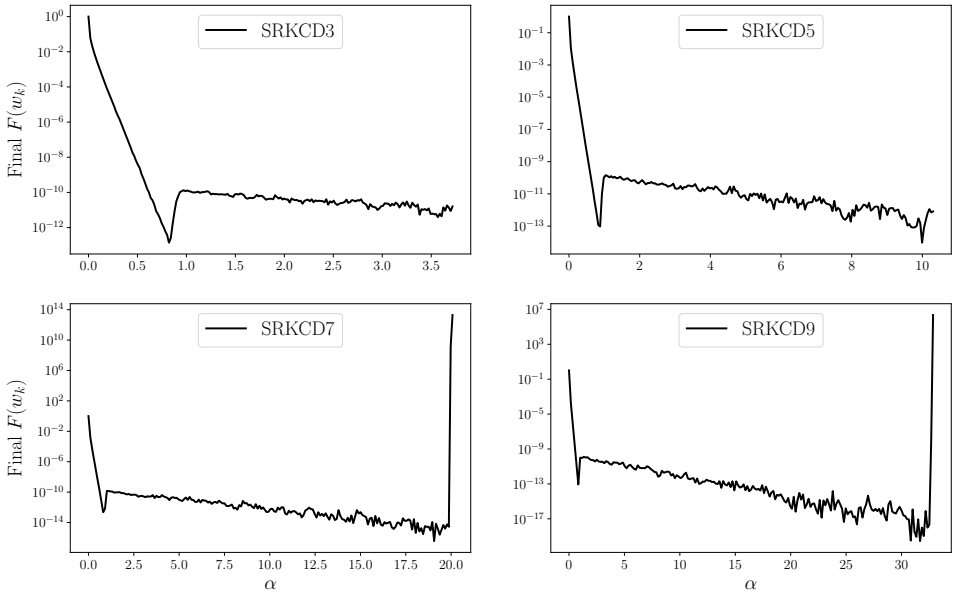

FIGURE 3. SRKCD with batch size 32 for various values of s when applied to the problem described in Section 4.1. In each case, 3 epochs were run.

start to increase again and larger step sizes will lead to instability and divergence. Interestingly, the picture is very similar for SGD. In this case, the step size limit is very slightly smaller than $\alpha = 2/L$ and we can observe some wiggles in the curve due to the stochastic approximations. But the optimal step size choice stays at almost the same position.

In Figure 3, we repeat the experiment with a batch size 32 but now with the SRKCD methods with different s. For each s, we try step sizes $\alpha \in (0, \frac{b_R}{L})$ where b_R is the maximal value such that $(-b_R, 0)$ is included in the stability region for the corresponding RKC method. It can be shown that $b_R = \frac{2\omega_0 T_s'(\omega_0)}{T_s(\omega_0)}$ [10, p.425].

The first thing to note is that as expected, the stability regions are much larger than for SGD. For larger s, they do not quite reach b_R/L in this stochastic setting,

but the differences are extremely small. Secondly, we note that all the methods exhibit a characteristic "dip" at a relatively small step size. This is similar to the optimal step size dip at $\frac{2}{\lambda_{\min}+\lambda_{\max}}$ for SGD. However, since the stability function of the corresponding RKC method has s zeroes instead of only one, there are also many other choices of larger α which yield comparable performance. Indeed, while SGD performs quite well in the interval $\alpha \in (0.3, 0.42)$, SRKCD with $s = 5$ performs roughly equally well for all $\alpha \in (0.5, 10.2)$.

We note that these plots cannot be used for efficiency comparisons, since the latter method has used 5 times as many evaluations of $g(\xi, \cdot)$ as SGD. Nevertheless, it is clear that the improved RKC stability properties makes SRKCD more robust. If, e.g. the values of $\lambda_{\min}$ and $\lambda_{\max}$ were not known, then selecting a good step size for SGD is difficult. For SRKCD, the choice almost does not matter.

4.2. **Convolutional neural network.** Next, we consider also an example arising from a real-world problem, namely the classification of images by convolutional neural networks. Such a problem can also be stated on the form $\min_w F(w)$, where F now depends on the collection of images, the network structure, and the loss function used to penalize mis-classifications. We refer to e.g. [3] for details. For this particular experiment, we set up a simple convolutional neural network consisting of one convolutional layer with a kernel size of 32×32 upon which we stack two fully connected dense layers with 128 and 10 neurons each. The activation function is ReLu for the first dense layer and softmax for the output layer and we use a crossentropy loss function. We train this network on the MNIST dataset [12] using both the SGD and the SRKCD algorithm with various stepsizes and number of stages s.

While a single training sequence is not so expensive, repeating it many times like in the previous section quickly becomes very time-consuming. Instead of illustrating the behaviour of the methods over a whole interval $\alpha \in (0, a)$ for some a, we therefore settle for trying to pin down the practical stability boundary. We recall that since this problem is nonlinear, we can not expect the stability properties to behave as nicely as in the previous experiment. This problem is also larger, but we still use a batch size of 32. As a consequence, the variance is larger than in the previous experiment, i.e. every realization is noisier. To alleviate this, we run each step size 5 times and take the average.

Figure 4 shows the final averaged loss values $F(w_k)$ after 1000 iterations for SGD and SRKCD with $s = 3, 4, 5$, for 10 step sizes close to the stability limit. The loss function F saturates around 2.4 which means that for such values the methods are unstable. Smaller values do not rule out that the methods could diverge in further iterations, but typically it rather indicates that we simply did not yet use enough iterations to decrease the loss further. Thus we can observe that for SGD, the practical stability limit is at around $\alpha = 0.35$. For SRKCD with $s = 3$, we instead estimate it to about $\alpha = 1.9$. For $s = 4$ and $s = 5$, we get about $\alpha = 2.8$ and $\alpha = 3.9$. Clearly these are very rough estimates, but as expected the stability properties of SRKCD are superior also in the nonlinear case. We note that e.g. $1.9 < 3^2 \cdot 0.35 = 3.15$, i.e. the s^2-scaling of the stability regions is not preserved for nonlinear problems. However, this is just one example and other types of problems might behave differently. Fully understanding the general nonlinear setting is a significant research undertaking.

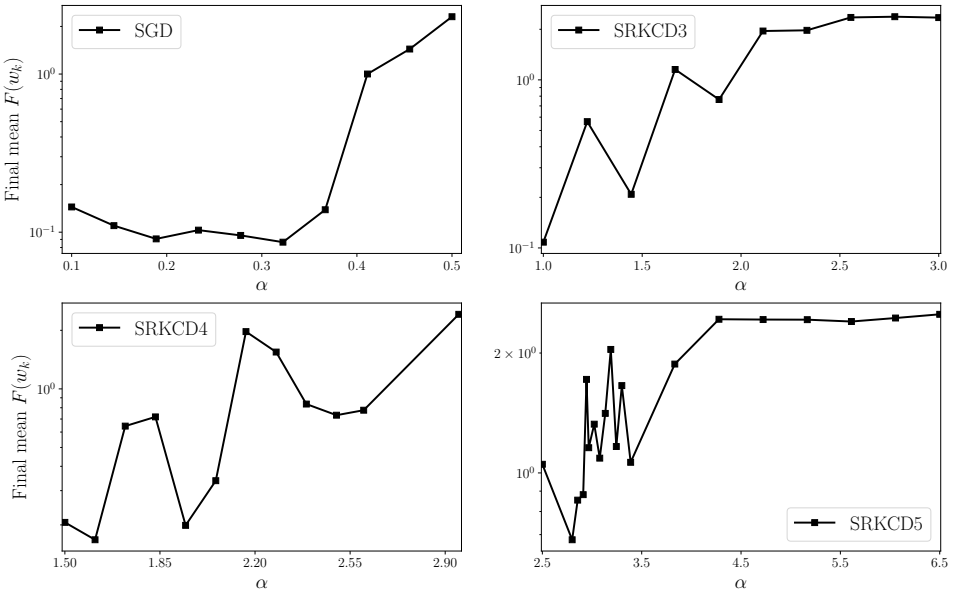

FIGURE 4. SGD and SRKCD with batch size 32 for various values of s when applied to the problem described in Section 4.2. In each case, 1000 iterations were run and the average final value of $F(w_k)$ over 5 paths is plotted versus the step size α.

5. Conclusions

We have introduced and analyzed the stochastic Runge–Kutta–Chebyshev descent (SRKCD) method by showing convergence in expectation to a unique minimum for a strongly convex objective function, and to a stationary point under certain regularity assumptions in the nonconvex case. While we have focused on the SRKCD methods because they exhibit the particular stability properties that were our original motivation, the proof is more general and applies to essentially any Runge-Kutta method. Other such methods may have properties that are of interest in this setting, this remains an open interesting research question.

As we have seen from the numerical experiments, the stability properties of the SRKCD methods are superior to SGD. This remains true also for nonlinear and nonconvex problems. We aim to investigate the efficiency of SRKCD in more detail, and also to compare it more extensively to other popular optimization methods. A key point to take into account here is of course that one iteration of SRKCD requires s approximative gradient evaluations, while most similar methods such as SGD require only one. In the usual setting of stiff ODEs, this is outweighed by being able to take much longer steps. In the current optimization context where it is not necessarily ideal to take the largest possible step, it is no longer as clear. We have, nevertheless, seen from the first numerical experiment that we can expect the SRKCD methods to be more robust in the sense that more step size choices give reasonable results in the absence of good model parameter estimates.

Finally, we note that in this stochastic setting one must use a decreasing step size sequence to actually reach a local minimum. With a fixed step size, we will only reach a neighbourhood of the minimum, whose size depends on the step size and

the variance of the approximative gradients. But with a very small step size, the better stability properties of SRKCD are irrelevant. These methods are therefore best employed in the initial phase where larger step sizes can and should be used, and where the convergence towards the minimum is rapid. We think that a hybrid method which utilizes SRKCD with decreasing values of s, eventually becoming SGD at $s = 1$, could be ideal.

Appendix A. Auxiliary results

In this appendix, we collect a few results that are important to our analysis but which are not of great interest on their own.

A.1. **Chebyshev polynomials.** The Chebyshev polynomials are given by

$$\begin{aligned} &T_0(x) = 1, \quad T_1(x) = x, \\ &T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n \geq 2. \end{aligned}$$

Lemma A.1. *For fixed $x \geq 1$ it holds that $T_n(x) \geq T_{n-1}(x)$ for $n \geq 1$. As a consequence, $T_n(x) \geq 1$ for all $n \geq 0$ if $x \geq 1$.*

Proof. We prove the lemma by induction. The statement is clearly true for $n = 1$. Assume that it is true for $n = k$, i.e. $T_k(x) - T_{k-1}(x) \geq 0$ for $x \geq 1$. Then

$$\begin{aligned} T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x) \\ &\geq 2T_k(x) - T_{k-1}(x) \\ &= T_k(x) + (T_k(x) - T_{k-1}(x)) \geq T_k(x). \end{aligned}$$

The fact that $T_n(x) \geq 1$ then follows directly from $T_0(x) = 1$. □

The RKC-update also depends on the derivatives of the Chebyshev polynomials so we also prove the same result for these:

Lemma A.2. *For fixed $x \geq 1$ it holds that $T_n'(x) \geq T_{n-1}'(x)$ for $n \geq 1$. Further, $T_n'(x) \geq 4$ for $n \geq 2$ if $x \geq 1$.*

Proof. From the definition of T_n, we find the following recursive formula for the derivatives $T_n'(x)$:

$$\begin{aligned} &T_0'(x) = 0, \ T_1'(x) = 1, \\ &T_n'(x) = 2T_{n-1}(x) + 2xT_{n-1}'(x) - T_{n-2}'(x), \quad n \geq 2. \end{aligned}$$

Now we can use induction again like in the previous Lemma. We clearly have $T_1'(x) \geq T_0'(x)$. Assuming that $T_n'(x) \geq T_{n-1}'(x)$ holds we get

$$\begin{aligned} T_{n+1}'(x) &= 2T_n(x) + 2xT_n'(x) - T_{n-1}'(x) \\ &\geq T_n'(x) + \big(T_n'(x) - T_{n-1}'(x)\big) \geq T_n'(x), \end{aligned}$$

where we used $T_n(x) \geq 1$ from Lemma A.1 in the first inequality. The final statement follows directly from the fact that $T_2'(x) = 4x$. □

References

[1] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, Springer, Cham, second ed., 2017, https://doi.org/10.1007/978-3-319-48311-5.

[2] P. Bianchi, *Ergodic convergence of a stochastic proximal point algorithm*, SIAM J. Optim., 26 (2016), pp. 2235–2260, https://doi.org/10.1137/15M1017909.

[3] L. Bottou, F. E. Curtis, and J. Nocedal, *Optimization methods for large-scale machine learning*, SIAM Rev., 60 (2018), pp. 223–311, https://doi.org/10.1137/16M1080173.

[4] J. Duchi, E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research, 12 (2011), pp. 2121–2159, http://jmlr.org/papers/v12/duchi11a.html.

[5] A. Eftekhari, B. Vandereycken, G. Vilmart, and K. C. Zygalakis, *Explicit stabilised gradient descent for faster strongly convex optimisation*, BIT Numerical Mathematics, 61 (2021), pp. 119–139, https://doi.org/10.1007/s10543-020-00819-y.

[6] M. Eisenmann, T. Stillfjord, and M. Williamson, *Sub-linear convergence of a stochastic proximal iteration method in Hilbert space*, arXiv e-prints, (2020), arXiv:2010.12348, https://arxiv.org/abs/arXiv:2010.12348.

[7] S. Gadat, F. Panloup, and S. Saadane, *Stochastic heavy ball*, Electron. J. Stat., 12 (2018), pp. 461–529, https://doi.org/10.1214/18-EJS1395.

[8] E. Hairer, S. P. Nörsett, and G. Wanner, *Solving ordinary differential equations. I*, vol. 8 of Springer Series in Computational Mathematics, Springer, Berlin, 2009, https://doi.org/10.1007/978-3-540-78862-1. Nonstiff problems, Second revised edition, paperback.

[9] G. Hinton, *Coursera neural networks for machine learning lecture 6*, 2018.

[10] W. Hundsdorfer and J. G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer, Berlin, Heidelberg, 2003, https://doi.org/10.1007/978-3-662-09017-6.

[11] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, arXiv e-prints, (2017), arXiv:1412.6980, https://arxiv.org/abs/arXiv:1412.6980. Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[12] Y. LeCun, C. Cortes, and C. Burges, *MNIST handwritten digit database*. Available at http://yann.lecun.com/exdb/mnist, 2010.

[13] Y. E. Nesterov, *A method for solving the convex programming problem with convergence rate $O(1/k^2)$*, Sov. Math., Dokl., 27 (1983), pp. 372–376. Translation from Dokl. Akad. Nauk SSSR 269(3), 543-547 (1983).

[14] A. J. Owens and D. L. Filkin, *Efficient training of the backpropagation network by solving a system of stiff ordinary differential equations*, in International 1989 Joint Conference on Neural Networks, vol. 2, 1989, pp. 381–386, https://doi.org/10.1109/IJCNN.1989.118726.

[15] B. Polyak, *Some methods of speeding up the convergence of iteration methods*, USSR Computational Mathematics and Mathematical Physics, 4 (1964), pp. 1–17, https://doi.org/10.1016/0041-5553(64)90137-5.

[16] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Statistics, 22 (1951), pp. 400–407, https://doi.org/10.1214/aoms/1177729586.

[17] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, *On the importance of initialization and momentum in deep learning*, in Proceedings of the 30th International Conference on Machine Learning, S. Dasgupta and D. McAllester, eds., vol. 28 of Proceedings of Machine Learning Research, Atlanta, Georgia, USA, 17–19 Jun 2013, PMLR, pp. 1139–1147, https://proceedings.mlr.press/v28/sutskever13.html.

[18] P. J. van der Houwen and B. P. Sommeijer, *On the internal stability of explicit, m-stage Runge-Kutta methods for large m-values*, Z. Angew. Math. Mech., 60 (1980), pp. 479–485, https://doi.org/10.1002/zamm.19800601005.

[19] M. D. Zeiler, *ADADELTA: An adaptive learning rate method*, arXiv e-prints, (2012), arXiv:1212.5701, https://arxiv.org/abs/1212.5701.