NAG-GS: SEMI-IMPLICIT, ACCELERATED AND ROBUST STOCHASTIC OPTIMIZER

Valentin Leplat[*]
Innopolis University
Innopolis, Russia
v.leplat@innopolis.ru

Daniil Merkulov
Skoltech, Moscow Institute of Physics and Technology
Moscow, Russia

Aleksandr Katrutsa
Skoltech, AIRI
Moscow, Russia

Daniel Bershatsky
Skoltech
Moscow, Russia

Olga Tsymboi
Skoltech, Sber AI Lab
Moscow, Russia

Ivan Oseledets
AIRI, Skoltech
Moscow, Russia

ABSTRACT

Classical machine learning models like deep neural networks are usually trained using Stochastic Gradient Descent-based (SGD) algorithms. The classical SGD can be interpreted as a discretization of the stochastic gradient flow. In this paper, we propose a novel, robust, and accelerated stochastic optimizer that relies on two key elements: (1) an accelerated Nesterov-like Stochastic Differential Equation (SDE) and (2) its semi-implicit Gauss-Seidel type discretization. The convergence and stability of the obtained method, called NAG-GS, are first extensively studied in the case of minimizing a quadratic function. This analysis allows us to come up with an optimal learning rate in terms of the convergence rate while ensuring the stability of NAG-GS. This is achieved by the careful analysis of the spectral radius of the iteration matrix and the covariance matrix at stationarity with respect to all hyperparameters of our method. Further, we show that NAG-GS is competitive with state-of-the-art methods such as momentum SGD with weight decay and AdamW for the training of machine learning models such as the logistic regression model, the residual networks models on standard computer vision datasets, Transformers in the frame of the GLUE benchmark and the recent Vision Transformers.

1 INTRODUCTION

Nowadays, deep learning has achieved promising results on a broad spectrum of AI application domains. Such models require training on vast amounts of data, and the training process usually corresponds to solving a complex optimization problem. The development of fast methods is urgently needed to speed up the learning process and obtain efficiently trained models (Gusak et al., 2022). This paper introduces a new optimization framework for solving such problems.

Main contributions of our paper:

- We present the stochastic extension of the algorithm from (Luo & Chen, 2021) based on the Gauss-Seidel discretization of the ODE related to the accelerated gradient method.
- We provide an asymptotic convergence analysis of the proposed method for the strongly convex quadratic objective and report the maximum feasible learning rate.
- We experimentally show that the proposed method converges faster in the first epochs and provides similar or better final test accuracy for training the logistic regression, VGG11, and ResNet18 models for image classification problems.

[*]Corresponding author

Organization of our paper:

- Section 1.1 gives the theoretical background for our method.
- In Section 2, we propose an accelerated system of Stochastic Differential Equations (SDE) and a corresponding solver based on a specific discretization method. This method, called NAG-GS (Nesterov Accelerated Gradient with Gauss-Seidel Splitting), is initially discussed in terms of convergence for quadratic functions. Additionally, we apply NAG-GS to solve a 1-dimensional non-convex SDE and provide strong numerical evidence of its superior acceleration compared to classical SDE solvers in Appendix B.
- In Section 3, NAG-GS is tested to tackle stochastic optimization problems of increasing complexity and dimension, starting from the logistic regression model to the training of large machine learning models such as ResNet-20, VGG-11, and Transformers.

1.1 Preliminaries

We start here with some general considerations in the deterministic setting for obtaining accelerated Ordinary Differential Equations (ODE) that will be extended in the stochastic setting in Section 2.1. We consider iterative methods for solving the unconstrained minimization problem:

$$\min_{x \in V} f(x), \tag{1}$$

where V is a Hilbert space, and $f : V \to \mathbb{R} \cup \{+\infty\}$ is a proper closed convex extended real-valued function. In the following, for simplicity, we shall consider the particular case of $\mathbb{R}^n$ for V and consider function f smooth on the entire space. We also suppose V is equipped with the canonical inner product $\langle x, y\rangle = \sum_{i=1}^n x_i y_i$ and the correspondingly induced norm $\|x\| = \sqrt{\langle x, x\rangle}$. Finally, we will consider in this section the class of functions $\mathcal{S}^{1,1}_{L,\mu}$ which stands for the set of strongly convex functions of parameter $\mu > 0$ with Lipschitz-continuous gradients with constant $L > 0$. For such class of functions, it is well-known that the global minimum exists and is unique (Nesterov, 2018). One well-known approach to deriving the Gradient Descent (GD) method is discretizing the gradient flow:

$$\dot{x}(t) = -\nabla f(x(t)), \quad t > 0. \tag{2}$$

The simplest forward (explicit) Euler method with step size $\alpha_k > 0$ leads to the GD method

$$x_{k+1} \leftarrow x_k - \alpha_k \nabla f(x_k).$$

In numerical analysis, it is widely recognized that this method is conditionally A-stable. Moreover, when considering $f \in \mathcal{S}^{1,1}_{L,\mu}$ with $0 \leq \mu \leq L \leq \infty$, the utilization of a step size $\alpha_k = 1/L$ leads to a linear convergence rate. It is important to highlight that the highest rate of convergence is attained when $\alpha_k = \frac{2}{\mu+L}$. In such a scenario, we have $\|x_k - x^\star\|^2 \leq \left(\frac{Q_f-1}{Q_f+1}\right)^{2k} \|x_0 - x^\star\|^2$, where Q_f is defined as $Q_f = \frac{L}{\mu}$ and is commonly referred to as the condition number of a function f (Nesterov, 2018). Another approach that can be considered is the backward (implicit) Euler method, which is represented as:

$$x_{k+1} \leftarrow x_k - \alpha_k \nabla f(x_{k+1}), \tag{3}$$

This method is unconditionally A-stable. In a nutshell, A-stability in numerical ordinary differential equations characterizes a method's performance in the asymptotic regime as time approaches infinity. An unconditionally A-stable method is one where the integration step can be arbitrarily large, yet the global error of the method converges to zero. We give more details about the notion in Appendix A.3. Hereunder, we summarize the methodology proposed by Luo & Chen (2021) to come up with a general family of accelerated gradient flows by focusing on the following simple problem:

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} x^\top A x \tag{4}$$

for which the gradient flow in (2) reads as:

$$\dot{x}(t) = -Ax(t), \quad t > 0, \tag{5}$$

where $A \in \mathbb{R}^{n \times n}$ is symmetric positive semi-definite matrix ensuring that $f \in \mathcal{S}^{1,1}_{L,\mu}$ where μ and L respectively correspond to the minimum and maximum eigenvalues of matrix A, which are real and positive by assumption. Instead of directly resolving (5), authors of Luo & Chen (2021) opted to address a general linear ODE system as follows:

$$\dot{y}(t) = Gy(t), \quad t > 0. \tag{6}$$

The central concept is to search for a system (6) with an asymmetric block matrix G that transforms the spectrum of A from the real line to the complex plane, reducing the condition number from $\kappa(A) = \frac{L}{\mu}$ to $\kappa(G) = O\left(\sqrt{\frac{L}{\mu}}\right)$. Subsequently, accelerated gradient methods can be constructed from A-stable methods to solve (6) with a significantly larger step size, improving the contraction rate from $O\left(\left(\frac{Q_f-1}{Q_f+1}\right)^{2k}\right)$ to $O\left(\left(\frac{\sqrt{Q_f}-1}{\sqrt{Q_f}+1}\right)^{2k}\right)$. Moreover, to handle the convex case $\mu = 0$, the authors in Luo & Chen (2021) combine the transformation idea with a suitable time scaling technique. In this paper, we consider one transformation that relies on the embedding of A into some 2×2 block matrix G with a rotation built-in (Luo & Chen, 2021):

$$G_{NAG} = \begin{bmatrix} -I & I \\ \mu/\gamma - A/\gamma & -\mu/\gamma I \end{bmatrix}, \tag{7}$$

where γ is a positive time scaling factor that satisfies

$$\dot{\gamma}(t) = \mu - \gamma(t), \quad \gamma(0) = \gamma_0 > 0. \tag{8}$$

Note that, given A positive definite, we can easily show that for the considered transformation, we have that $\mathcal{R}(\lambda) < 0$ that is the real part of λ is strictly negative, and this for all $\lambda \in \sigma(G)$ with $\sigma(G)$ denotes the spectrum of G, i.e., the set of all eigenvalues of G. Further, we will denote by $\rho(G) := \max_{\lambda \in \sigma(G)} |\lambda|$ the spectral radius of matrix G. Let us now consider the NAG block Matrix and let $y = (x, v)$, the dynamical system given in (6) with $y(0) = y_0 \in \mathbb{R}^{2n}$ reads:

$$\begin{aligned} \frac{dx}{dt} &= v - x, \\ \frac{dv}{dt} &= \frac{\mu}{\gamma}(x - v) - \frac{1}{\gamma}Ax \end{aligned} \tag{9}$$

with initial conditions $x(0) = x_0$ and $v(0) = v_0$. Note that this linear ODE can be expressed as:

$$\gamma\ddot{x} + (\gamma + \mu)\dot{x} + Ax = 0, \tag{10}$$

where Ax is therefore the gradient of f w.r.t. x. Thus, one could generalize this approach for any function $f \in \mathcal{S}^{1,1}_{L,\mu}$ by replacing Ax by $\nabla f(x)$, respectively, within (7), (9) and (10). Finally, some additional and useful insights are discussed in Appendix A.

2 Model and Theory

2.1 Accelerated Stochastic Gradient flow

In the previous section, we presented a family of accelerated Gradient flows obtained by an appropriate spectral transformation G of matrix A, see (9). Let us recall that Ax in (9) can be replaced by $\nabla f(x)$ for any function $f \in \mathcal{S}^{1,1}_{L,\mu}$. Within the context of training neural networks, function $f(x)$ corresponds to some loss function, and its gradient $\nabla f(x)$ is contaminated by noise due to a finite-sample gradient estimate. The study of accelerated Gradient flows is now adapted to include and model the effect of the noise; to achieve this, we consider the dynamics given in (6) perturbed by a general martingale process. This leads us to consider the following Accelerated Stochastic Gradient (ASG) flow:

$$\begin{aligned} \frac{dx}{dt} &= v - x, \\ \frac{dv}{dt} &= \frac{\mu}{\gamma}(x - v) - \frac{1}{\gamma}Ax + \frac{dZ}{dt}, \end{aligned} \tag{11}$$

which corresponds to an (Accelerated) system of SDE's, where $Z(t)$ is a continuous Ito martingale. We assume that $Z(t)$ has the simple expression $dZ = \sigma dW$, where $W = (W_1, \ldots, W_n)$ is a standard n-dimensional Brownian Motion. As a simple and first approach, we consider the parameter σ constant. The next section presents the discretizations corresponding to the ASG flow given in (11).

2.2 Discretization: Gauss-Seidel Splitting and Semi-Implicitness

This section presents the primary strategy to discretize the Accelerated SDE's system from (11). The motivation behind the discretization method is to derive integration schemes that are, in the best case, unconditionally A-stable or conditionally A-stable with the highest possible integration step. In the classical terminology of (discrete) optimization methods, this value ensures convergence of the obtained methods with the largest possible step size. Consequently, it improves the contraction rate (or the rate of convergence). In Section 1.1, we have briefly recalled that the most well-known unconditionally A-stable scheme was the backward Euler method (see equation 3), which is an implicit method and hence can achieve faster convergence rate. However, this requires solving a linear system or, in the case of a general convex function, computing the root of a non-linear equation, leading to a high computational cost. This is why few implicit schemes are used to solve high-dimensional optimization problems. Still, an explicit scheme closer to the implicit Euler method is expected to have good stability with a larger step size than a forward Euler method. Furthermore, assuming a Gaussian noise process, it is crucial to propose a solver capable of handling a wide range of step size values. Specifically, allowing for a larger ratio α/b (with b as the mini-batch size) increases the likelihood of converging to wider local minima, ultimately enhancing the generalization performance of the trained model, see Appendix A for additional details on that matter. Motivated by the Gauss-Seidel (GS) method for solving linear systems, we consider the matrix splitting $G = M+N$ with M being the lower triangular part of G and $N = G - M$, we propose the following Gauss-Seidel splitting scheme for (6) perturbated with noise:

$$\frac{y_{k+1} - y_k}{\alpha_k} = My_{k+1} + Ny_k + \begin{bmatrix} 0 \\ \sigma \frac{W_{k+1}-W_k}{\alpha_k} \end{bmatrix} \tag{12}$$

which for $G = G_{NAG}$ (see (7)), gives the following semi-implicit scheme with step size $\alpha_k > 0$:

$$\begin{aligned} \frac{x_{k+1} - x_k}{\alpha_k} &= v_k - x_{k+1}, \\ \frac{v_{k+1} - v_k}{\alpha_k} &= \frac{\mu}{\gamma_k}(x_{k+1} - v_{k+1}) - \frac{1}{\gamma_k}Ax_{k+1} + \sigma\frac{W_{k+1} - W_k}{\alpha_k}. \end{aligned} \tag{13}$$

Note that due to the properties of Brownian motion, we can simulate its values at the selected points by: $W_{k+1} = W_k + \Delta W_k$, where ΔW_k are independent random variables with distribution $\mathcal{N}(0, \alpha_k)$. Furthermore, ODE (8) corresponding to the parameter γ is also discretized implicitly:

$$\frac{\gamma_{k+1} - \gamma_k}{\alpha_k} = \mu - \gamma_{k+1}, \quad \gamma_0 > 0. \tag{14}$$

As already mentioned earlier, heuristically, for general $f \in \mathcal{S}^{1,1}_{L,\mu}$ with $\mu \geq 0$, we just replace Ax in (13) with $\nabla f(x)$ and obtain the following NAG-GS scheme:

$$\begin{aligned} \frac{x_{k+1} - x_k}{\alpha_k} &= v_k - x_{k+1}, \\ \frac{v_{k+1} - v_k}{\alpha_k} &= \frac{\mu}{\gamma_k}(x_{k+1} - v_{k+1}) - \frac{1}{\gamma_k}\nabla f(x_{k+1}) + \sigma\frac{W_{k+1} - W_k}{\alpha_k}. \end{aligned} \tag{15}$$

Finally, we introduce the NAG-GS method (see Algorithm 1). In this method, we consider the presence of unknown noise when computing the gradient $\nabla f(x_{k+1})$. We denote this noisy gradient as $\nabla \tilde{f}(x_{k+1})$ in Algorithm 1. Notably, to achieve strict equivalence with the scheme described in (15), we have the relationship $\nabla \tilde{f}(x_{k+1}) = \nabla f(x_{k+1}) + \sigma\mu(1 - \frac{1}{b_k})(W_{k+1} - W_k)$, where b_k is defined as $b_k := \alpha_k\mu(\alpha_k\mu + \gamma_{k+1})^{-1}$.

Remark 1 (Complexity of NAG-GS vs. AdamW)**.** *According to Algorithm 1, the NAG-GS algorithm requires* one *auxiliary vector with a dimension equal to the number of the trained parameters. In contrast, AdamW requires* two *auxiliary vectors of the same dimension and updates them respectively. Hence, NAG-GS has lower computational complexity and memory requirements than AdamW.*

Moreover, the step size update can be performed using different strategies. For instance, one may choose the method proposed in (Nesterov, 2018, Method 2.2.7) which specifies to compute $\alpha_k \in (0, 1)$ such that $L\alpha_k^2 = (1 - \alpha_k)\gamma_k + \alpha_k\mu$. Note that for $\gamma_0 = \mu$, hence the sequences $\gamma_k = \mu$ and $\alpha_k = \sqrt{\frac{\mu}{L}}$ for all $k \geq 0$. In Section 2.3, we discuss how to compute the step size for Algorithm 1.

Algorithm 1 Nesterov Accelerated Gradients with Gauss–Seidel splitting (NAG-GS).

```
Input: Choose point $x_0 \in \mathbb{R}^n$, some $\mu \geq 0, \gamma_0 > 0$.
  Set $v_0 := x_0$.
  for $k = 1, 2, \ldots$ do
    Choose step size $\alpha_k > 0$.
    ▷ Update parameters and state $x$:
    Set $a_k := \alpha_k(\alpha_k + 1)^{-1}$.
    Set $\gamma_{k+1} := (1 - a_k)\gamma_k + a_k\mu$.
    Set $x_{k+1} := (1 - a_k)x_k + a_k v_k$.
    ▷ Update state $v$:
    Set $b_k := \alpha_k\mu(\alpha_k\mu + \gamma_{k+1})^{-1}$.
    Set $v_{k+1} := (1 - b_k)v_k + b_k x_{k+1} - \mu^{-1} b_k \nabla\tilde{f}(x_{k+1})$.
  end for
```

Let us mention that the authors have considered and studied full-implicit discretizations; see details in Appendix A.2. However, interest in such methods is limited to ML applications since the obtained implicit schemes use second-order information about f. Therefore, such schemes are typically intractable for real-life ML models.

2.3 Convergence analysis of quadratic case

We propose to study how to select a maximum step size that ensures an optimal contraction rate while guaranteeing the convergence or the stability of the NAG-GS method once used to solve SDE's system (11). Ultimately, we show that the choice of the optimal step size is mainly influenced by the values of μ, L, and γ. These (hyper)parameters are central, and to show this, we study two key quantities, namely the spectral radius of the iteration matrix and the covariance matrix associated with the NAG-GS method summarized by Algorithm 1. Note that this theoretical study only concerns the case $f(x) = \frac{1}{2}x^\top Ax$. Considering the size limitation of the paper, we present below only the main theoretical result and place its proof in Appendix A.1.4:

Theorem 1. *For G_{NAG} (7), given $\gamma \geq \mu$, and assuming $0 < \mu = \lambda_1 \leq \ldots \leq \lambda_n = L < \infty$; if $0 < \alpha \leq \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}$, then the NAG-GS method summarized by Algorithm 1 is convergent for the n-dimensional case, with $n > 2$.*

Remark 2. *It is essential to mention that the optimal contraction rate of NAG-GS aiming at minimizing a strongly convex quadratic function is reached for $\alpha^* = \frac{\gamma+\mu+\sqrt{(\gamma-\mu)^2+4\gamma L}}{L-\mu}$.*

All the steps of the convergence analysis are fully detailed Appendix A.1 and organized as follows:

- Appendix A.1.1 and Appendix A.1.2 respectively provide the full analysis of the spectral radius of the iteration matrix associated with the NAG-GS method and the covariance matrix at stationarity w.r.t. hyperparameters μ, L, γ and σ, for the case of the dimension $n = 2$. The theoretical results obtained are summarized in Appendix A.1.3 to come up with an optimal step size in terms of contraction rate. The extension to $n > 2$ is detailed in Appendix A.1.4 along with the proof of Theorem 1.
- Numerical tests are performed and detailed in Appendix A.1.5 to support the theoretical results obtained for the quadratic case.

3 Experiments

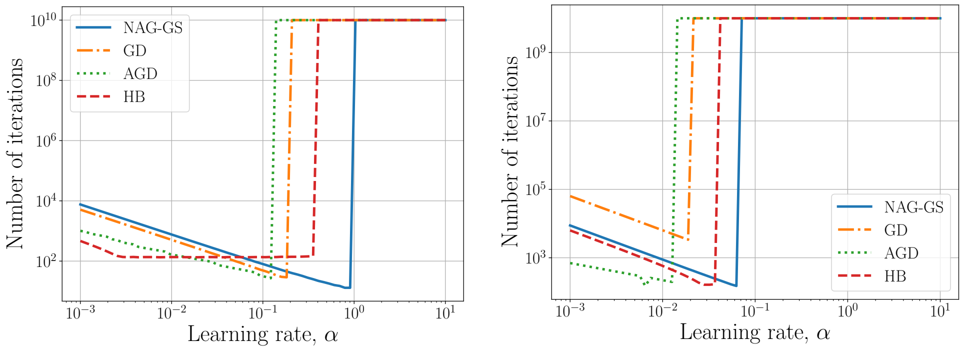
We test the NAG-GS method on several neural architectures: logistic regression, transformer model for natural language processing (RoBERTa model) and computer vision (ViT model) tasks, and residual networks for computer vision tasks (ResNet20). To ensure a fair benchmark of our method on these neural architectures, we replace only the optimizer with NAG-GS while preserving all other hyperparameters of the training process. Our experiments can be reproduced using the source code[1].

[1] https://github.com/skolai/nag-gs

3.1 Toy problem

We compare the convergence of the NAG-GS method for a strongly convex quadratic function with other first-order methods. This test demonstrates that NAG-GS can work with larger learning rates. If a large learning rate is used, NAG-GS converges faster than competitors with smaller learning rates.

Strongly convex quadratic function. Consider the problem $\min_x f(x)$, where $f(x) = \frac{1}{2}x^\top Ax - b^\top x$ is convex quadratic function. The matrix $A \in \mathbb{S}^n_{++}$ is symmetric and positive semidefinite, $L = \lambda_{\max}(A)$, $\mu = \lambda_{\min}(A)$ and $n = 100$. We assume that method converges if $f(x_k) - f^* \leq 10^{-4}$, where $f^* = f(x^*)$ is the optimum function value. If some learning rate leads to divergence, we set the number of iterations to 10^{10}. In this experiment, we use the accelerated gradient descent (AGD) version from Su et al. (2014). In NAG-GS we use constant $\gamma = \mu = \lambda_{\min}(A)$. In the HB method, we use $\beta = 0.9$. Also, we test 70 learning rates distributed uniformly in the logarithmic grid in the interval $[10^{-3}, 10]$. Figure 1 shows the dependence of the number of iterations needed for convergence of NAG-GS, gradient descent (GD), and accelerated gradient descent (AGD) on the learning rates for different μ and L. We observe that NAG-GS provides two benefits. First, it converges for larger learning rates than GD, AGD, and HB methods. Second, in the large learning rate regime, NAG-GS converges faster in terms of the number of iterations than GD and AGD. Although the AGD method is optimal among the first-order methods, its feasible learning rate is strictly bounded by $1/L$. In contrast, the non-asymptotic convergence of the NAG-GS method is still the subject of future work. However, NAG-GS with a large learning rate, which is infeasible for the AGD method, shows faster convergence even for the ill-conditioned problem, see Figure 1b.



(a) $\mu = 1$, $L = 10$

(b) $\mu = 10^{-1}$, $L = 100$

Figure 1: Dependence of the number of iterations needed for convergence on the learning rate. NAG-GS is more robust with respect to the learning rate than gradient descent (GD) and accelerated gradient descent (AGD). Also, NAG-GS converges faster than competitors if the learning rate is sufficiently large. The number of iterations 10^{10} indicates the divergence.

3.2 Logistic regression

In this section, we benchmark the NAG-GS method against SGD with momentum (SGD-M) and AdamW optimizers on the multiclass logistic regression model and MNIST dataset (LeCun et al., 2010). Since this problem is convex and non-quadratic, we consider this problem as the natural next test case after the theoretical analysis of the NAG-GS method in Section 2.3 and numerical tests for the quadratic convex problem. We use the following hyperparameters: batch size is 1024, momentum term in SGD-M equals 0.9, $\mu = 1$ and $\gamma = 1$ in NAG-GS, and no weight decay in SGD-M and AdamW, i.e. AdamW coincides with Adam optimizer. In Table 1, we present the learning rates, final test accuracy, and the number of epochs necessary for convergence for every optimizer. We assume that the training process has converged if the change in the test accuracy is minor. We observe that the larger the learning rate, the better the performance of the NAG-GS. Note that the highest test accuracy is achieved by competitors only after 20 epochs while NAG-GS with larger learning rate achieves the similar performance already after ten epochs. At the same time, we observe that the constant small learning rate leads to poor performance of the NAG-GS method. These observations motivate using the NAG-GS optimizer in the first stage of the training to achieve high test accuracy after a small number of epochs. If this test accuracy is still insufficient, switching to another optimizer or decreasing learning rate could be a good strategy for performance improvement. In this section, we confirm numerically that the NAG-GS method allows larger learning rate values than the SGD-M and AdamW optimizers. Moreover, the results indicate that the semi-implicit nature of the NAG-GS method indeed ensures the acceleration effect through the use of larger learning rates while preserving the high performance of the model. This behavior holds not only for convex quadratic problems but also for non-quadratic convex ones.

Table 1: Test accuracies for NAG-GS, SGD-M, and AdamW that train the multiclass logistic regression model. NAG-GS with a learning rate $\alpha = 0.5$ converges after ten epochs to high test accuracy, similar to the performance of AdamW, which converges for 20 epochs and uses $\alpha = 10^{-3}$. We skip AdamW performance for $\alpha = 0.1$ and $\alpha = 0.5$ since it is much worse than SGD-M.

Learning rate, α	Optimizer	# epochs	Test accuracy
10^{-3}	NAG-GS	60	0.8767
	SGD-MW	30	0.9142
	AdamW	20	**0.9255**
10^{-2}	NAG-GS	20	0.9014
	SGD-M	20	**0.9224**
	AdamW	10	0.9215
0.1	NAG-GS	10	0.9171
	SGD-M	10	**0.9204**
0.5	NAG-GS	10	**0.9224**
	SGD-M	10	0.9113

3.3 VGG-11 AND RESNET-20

We compare NAG-GS and SGD with momentum and weight decay (SGD-MW) on VGG-11 (Simonyan & Zisserman, 2014) and ResNet-20 (He et al., 2016) models. We do not consider AdamW here since it consumes more memory than SGD-MW, and this drawback becomes significant for these models. Training these models leads to non-convex stochastic optimization problems, which appear to be the next complexity level for the testing NAG-GS optimizer. Below, we demonstrate numerically the superior performance of NAG-GS in the first epochs of training the considered models using a large learning rate.

VGG-11. We test the VGG-11 model on the CIFAR-10 image classification problem (Krizhevsky, 2009) and demonstrate the robustness of NAG-GS to large learning rates compared to SGD-MW. The hyperparameters are the following: batch size equals 400, and the number of epochs is 50. We use the constant $\gamma = 1.$ and $\mu = 10^{-4}$ for NAG-GS, momentum term equal 0.9 and weight decay is 10^{-4} in SGD-MW. In comparison, we use two approaches: the number of epochs necessary to achieve the best test accuracy (the maximum number of epochs is 50) and the convergence in the first epochs.

Table 2: Best test accuracies are given by NAG-GS and SGD-MW in training VGG11 model and CIFAR10 dataset and the number of epochs to achieve it. NAG-GS gives higher test accuracy for large learning rates *faster* than SGD-MW (39 vs 49 epochs) for the smaller learning rate. Dash means divergence of the training.

Learning rate, α	Optimizer	Best test accuracy	# epochs
10^{-3}	NAG-GS	41.52	50
	SGD-MW	75.07	48
10^{-2}	NAG-GS	75.09	50
	SGD-MW	80.11	49
0.1	NAG-GS	80.14	49
	SGD-MW	–	–
0.15	NAG-GS	80.29	39
	SGD-MW	–	–

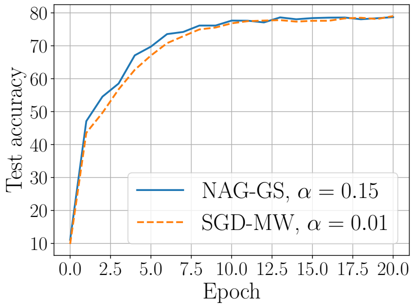


Figure 2: Comparison of convergence NAG-GS and SGD-MW with different learning rates. NAG-GS gives a higher test accuracy faster than SGD-MW (see 1–10 epochs) while converging to a similar test accuracy in the middle of training.

The first approach is illustrated with Table 2, where we show the best test accuracy achieved in training and the number of epochs required for it. From this table, it follows that NAG-GS achieved higher best test accuracy for a smaller number of epochs than SGD-M due to the larger learning rate. The second approach is presented in Figure 2, where one can see that NAG-GS with a large learning rate ($\alpha = 0.15$) converges faster in the first ten epochs than SGD-MW with a smaller learning rate. We show only the first 20 epochs instead of the complete 50 epochs to highlight the gap at the beginning of the training. In further epochs, the test accuracy given by NAG-GS and SGD-MW remains similar.

ResNet-20. We carried out intensive experiments to evaluate the performance of NAG-GS in training residual networks. In particular, we use the ResNet-20 model and CIFAR-10 dataset and the corresponding parameters reported in the literature. The experimental setup is the same except for the optimizer and its parameters. The best test score for NAG-GS is achieved for $\alpha = 0.11$, $\gamma = 17$, and $\mu = 0.01$. We compare the convergence of NAG-GS and SGD-MW in terms of loss and test accuracy in Figure 3. The proposed NAG-GS optimizer provides better test accuracy than SGD-MW during the first 150 epochs. In the following epochs, the performance of NAG-GS and SGD-MW becomes similar. This observation confirms that the potential application of the NAG-GS optimizer is to improve the performance in the first epochs to speed up convergence and reduce the costs for the warm-up stage. A detailed analysis of this effect will be the subject of future work.

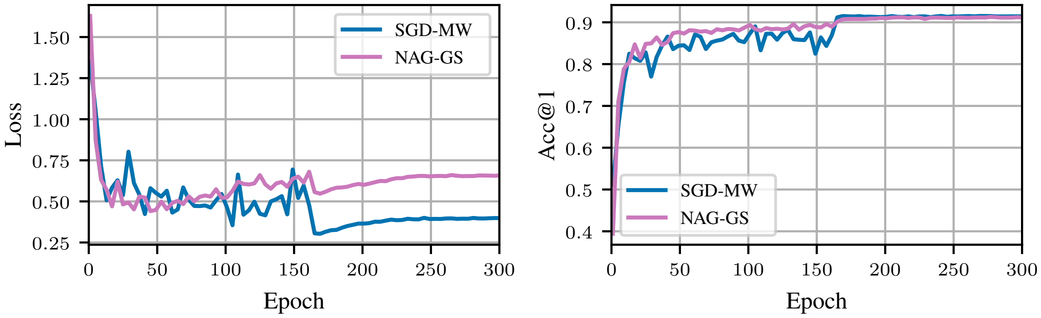


Figure 3: Comparison of NAG-GS and SGD-MW on ResNet-20 model and CIFAR-10. NAG-GS outperforms SGD-MW uniformly in the first 150 epochs and provides the same accuracy further.

3.4 Transformer Models

3.4.1 RoBERTa

In this section, we test NAG-GS for fine-tuning the pre-trained RoBERTa model from TRANSFORMERS library (Wolf et al., 2020) on GLUE benchmark datasets (Wang et al., 2018). In this benchmark, the reference optimizer is AdamW (Loshchilov & Hutter, 2019) with a polynomial learning rate schedule. The training setup defined in Liu et al. (2019) is used for both NAG-GS and AdamW optimizers. We search for an optimal learning rate ad γ for the NAG-GS optimizer with fixed μ to get the best performance on the task. Search space consists of learning rate α from $[10^{-3},\, 10^{0}]$, factor γ from $[10^{-2},\, 10^{0}]$, and $\mu = 1$. Note that NAG-GS is used with a constant learning rate to simplify hyperparameter search. Regarding learning rate values, the one allowed by AdamW is around 10^{-5} while NAG-GS allows a much larger value of 10^{-2}. Evaluation results on GLUE tasks are presented in Table 3. Despite a rather restrained search space for NAG-GS hyperparameters, it demonstrates better performance on some tasks and competitive performance on others.

3.4.2 Vision Transformer model

We used the Vision Transformer model (Wu et al., 2020), which was pre-trained on the ImageNet dataset (Deng et al., 2009), and fine-tuned it on the `food101` dataset (Bossard et al., 2014) using NAG-GS and AdamW. This task involves classifying a dataset of 101 food categories, with 1000 images per class. To ensure a fair comparison, we first conducted an intensive hyperparameter search (Biewald, 2020) for all possible hyperparameter configurations on a subset of the data for each method and selected the best configuration. After the hyperparameter search, we performed the

Table 3: Comparison of AdamW and NAG-GS optimizers in fine-tuning on GLUE benchmark. We use reported hyperparameters for AdamW. We search hyperparameters of NAG-GS for the best performance metric. NAG-GS outperforms or is competitive with AdamW on the considered tasks.

OPTIMIZER	COLA	MNLI	MRPC	QNLI	QQP	RTE	SST2	STS-B	WNLI
ADAMW	**61.60**	**87.56**	88.24	**92.62**	**91.69**	**78.34**	**94.95**	**90.68**	**56.34**
NAG-GS	**61.60**	87.24	**90.69**	92.59	91.01	77.97	94.50	90.21	**56.34**

experiments on the entire dataset. The results are presented in Table 4. We observed that properly-tuned NAG-GS outperformed AdamW in both training and evaluation metrics. Also, NAG-GS reached higher accuracy than AdamW after one epoch. The optimal hyperparameters found for NAG-GS are $\alpha = 0.07929, \gamma = 0.3554, \mu = 0.1301$; for AdamW $\text{lr} = 0.00004949, \beta_1 = 0.8679, \beta_2 = 0.9969$.

Table 4: Test accuracies of NAG-GS and AdamW for Vision Transformer model fine-tuned on `food101` dataset. The NAG-GS outperforms AdamW after the presented number of epochs.

Stage	NAG-GS	AdamW
After 1 epoch	**0.8419**	0.8269
After 25 epochs	**0.8606**	0.8324

4 RELATED WORKS

The approach of interpreting and analyzing optimization methods from the ODEs discretization perspective is well-known and widely used in practice (Muehlebach & Jordan, 2019; Wilson et al., 2021; Shi et al., 2021). The main advantage of this approach is that it constructs a direct correspondence between the properties of some classes of ODEs and their associated optimization methods. In particular, gradient descent and Nesterov accelerated methods are discussed in Su et al. (2014) as a particular discretization of ODEs. In the same perspective, many other optimization methods were analyzed, we can mention the mirror descent method and its accelerated versions (Krichene et al., 2015), the proximal methods (Attouch et al., 2019) and ADMM (Franca et al., 2018). It is well known that a discretization strategy is essential for transforming a particular ODE into an efficient optimization method. In particular, Shi et al. (2019); Zhang et al. (2018) investigate the most proper discretization techniques for different classes of ODEs. A similar analysis for stochastic first-order methods is presented in Laborde & Oberman (2020); Malladi et al. (2022). Recent advances in deriving optimal optimizers (Taylor & Drori, 2023; Zhou et al., 2020) do not exploit the ODE interpretations and only consider a deterministic setup.

5 CONCLUSION AND FURTHER WORKS

We have presented a new and theoretically motivated stochastic optimizer called NAG-GS. It comes from the semi-implicit Gauss-Seidel discretization of a well-chosen accelerated Nesterov-like SDE. These building blocks ensure two central properties for NAG-GS: (1) the ability to accelerate the optimization process and (2) better robustness to large learning rates. We demonstrate these features theoretically and provide a detailed analysis of the convergence of the method in the quadratic case. Moreover, we show that NAG-GS is competitive with state-of-the-art methods for training a small logistic regression model and larger models like ResNet-20, VGG-11, and Transformers. In numerical tests, NAG-GS demonstrates faster convergence in the first epochs due to the larger learning rate and the similar final scores to standard optimizers, which work only with smaller learning rates. Further work will focus on the non-asymptotic convergence analysis of NAG-GS and the development of proper learning rate schedulers for it.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

Anton Arnold, Peter Markowich, Giuseppe Toscani, and Andreas Unterreiter. On convex sobolev inequalities and the rate of convergence to equilibrium for fokker-planck type equations. 2001.

Hedy Attouch, Zaki Chbani, and Hassan Riahi. Fast proximal methods via time scaling of damped inertial dynamics. *SIAM Journal on Optimization*, 29(3):2227–2256, 2019.

Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, John Quan, George Papamakarios, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Luyu Wang, Wojciech Stokowiec, and Fabio Viola. The DeepMind JAX Ecosystem, 2020. URL `http://github.com/deepmind`.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical Gauss-Newton optimisation for deep learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 557–565. PMLR, 06–11 Aug 2017.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable Transformations of Python+NumPy Programs, 2018. URL `http://github.com/google/jax`.

Sinho Chewi, Thibaut Le Gouic, Chen Lu, Tyler Maunu, and Philippe Rigollet. Svgd as a kernelized wasserstein gradient flow of the chi-squared divergence. *Advances in Neural Information Processing Systems*, 33:2098–2109, 2020.

Germund G Dahlquist. A special stability problem for linear multistep methods. *BIT Numerical Mathematics*, 3(1):27–43, 1963.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Guilherme Franca, Daniel Robinson, and Rene Vidal. Admm and accelerated admm as continuous dynamical systems. In *International Conference on Machine Learning*, pp. 1559–1567. PMLR, 2018.

Gene H. Golub and Charles F. van Loan. *Matrix Computations*. JHU press, 2013.

Julia Gusak, Daria Cherniuk, Alena Shilova, Alexandr Katrutsa, Daniel Bershatsky, Xunyi Zhao, Lionel Eyraud-Dubois, Oleh Shliazhko, Denis Dimitrov, Ivan V Oseledets, et al. Survey on efficient training of large neural networks. In *IJCAI*, pp. 5494–5501, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=H1oyRlYgg`.

Walid Krichene, Alexandre Bayen, and Peter L Bartlett. Accelerated mirror descent in continuous and discrete time. *Advances in neural information processing systems*, 28, 2015.

Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. Master's thesis, University of Toronto, 2009.

Maxime Laborde and Adam Oberman. A lyapunov analysis for accelerated gradient methods: from deterministic to stochastic case. In *International Conference on Artificial Intelligence and Statistics*, pp. 602–612. PMLR, 2020.

Marc Lambert, Sinho Chewi, Francis Bach, Silvère Bonnabel, and Philippe Rigollet. Variational inference via wasserstein gradient flows. *arXiv preprint arXiv:2205.15902*, 2022.

Jonas Latz. Analysis of stochastic gradient descent in continuous time. *Statistics and Computing*, 31 (4):1–25, 2021.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, 2, 1944.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Hao Luo and Long Chen. From differential equation solvers to accelerated first-order methods for convex optimization. *Mathematical Programming*, pp. 1–47, 2021.

Sadhika Malladi, Kaifeng Lyu, Abhishek Panigrahi, and Sanjeev Arora. On the sdes and scaling rules for adaptive gradient algorithms. *arXiv preprint arXiv:2205.10287*, 2022.

Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

Gisiro Maruyama. Continuous markov processes and stochastic equations. *Rendiconti del Circolo Matematico di Palermo*, 4(1):48–90, 1955.

Michael Muehlebach and Michael Jordan. A dynamical systems perspective on nesterov acceleration. In *International Conference on Machine Learning*, pp. 4656–4662. PMLR, 2019.

Yurii Nesterov. *Lectures on Convex optimization*, volume 137. Springer Optimization and Its Applications, 2018.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic Differentiation in PyTorch, 2017.

Fernando Pérez-Cruz. Kullback-leibler divergence estimation of continuous distributions. In *2008 IEEE international symposium on information theory*, pp. 1666–1670. IEEE, 2008.

G.O. Roberts and O. Stramer. Langevin diffusions and metropolis-hastings algorithms. *Methodology And Computing In Applied Probability*, 4:337–357, 2002.

Bin Shi, Simon S Du, Weijie Su, and Michael I Jordan. Acceleration via symplectic discretization of high-resolution differential equations. *Advances in Neural Information Processing Systems*, 32, 2019.

Bin Shi, Simon S Du, Michael I Jordan, and Weijie J Su. Understanding the acceleration phenomenon via high-resolution differential equations. *Mathematical Programming*, pp. 1–70, 2021.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: theory and insights. *Advances in neural information processing systems*, 27, 2014.

Adrien Taylor and Yoel Drori. An optimal gradient method for smooth strongly convex minimization. *Mathematical Programming*, 199(1-2):557–594, 2023.

Maksim Velikanov, Denis Kuznedelev, and Dmitry Yarotsky. A view of mini-batch sgd via generating functions: conditions of convergence, phase transitions, benefit from negative momenta. 2022. doi: 10.48550/arxiv.2206.11124.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Ashia C Wilson, Ben Recht, and Michael I Jordan. A lyapunov analysis of accelerated methods in optimization. *J. Mach. Learn. Res.*, 22:113–1, 2021.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020.

Matthew D. Zeiler. Adadelta: An adaptive learning rate method, 2012.

Jingzhao Zhang, Aryan Mokhtari, Suvrit Sra, and Ali Jadbabaie. Direct runge-kutta discretization achieves acceleration. *Advances in neural information processing systems*, 31, 2018.

Kaiwen Zhou, Anthony Man-Cho So, and James Cheng. Boosting first-order methods by shifting objective: new schemes with faster worst-case rates. *Advances in Neural Information Processing Systems*, 33:15405–15416, 2020.

A Additional remarks related to theoretical background

An accelerated ODE has been presented in the main text Section 1.1, which relied on a specific spectral transformation. In this brief section, we add some useful insights:

- Equation (10) is a variant of the heavy ball model with variable damping coefficients in front of $\ddot{x}$ and $\dot{x}$.
- Thanks to the scaling factor γ , both the convex case $\mu = 0$ and the strongly convex case $\mu > 0$ can be handled in a unified way.
- In the continuous time, one can solve easily (8) as follows: $\gamma(t) = \mu + (\gamma_0 - \mu)e^{-t}, \quad t \geq 0$. Since $\gamma_0 > 0$, we have that $\gamma(t) > 0$ for all $t \geq 0$ and $\gamma(t)$ converges to μ exponentially and monotonically as $t \to +\infty$. In particular, if $\gamma_0 = \mu > 0$, then $\gamma(t) = \mu$ for all $t \geq 0$. We remark here the links between the behavior of the scaling factor $\gamma(t)$ and the sequence $\{\gamma_k\}_{k=0}^{\infty}$ introduced by Nesterov Nesterov (2018) in its analysis of optimal first-order methods in discrete-time, see (Nesterov, 2018, Lemma 2.2.3).

- Authors from Luo & Chen (2021) prove the exponential decay property $\mathcal{L}(t) \leq e^{-t}\mathcal{L}_0, \quad t > 0$ for a Taylored Lyapunov function $\mathcal{L}(t) := f(x(t)) - f(x^\star) + \frac{\gamma(t)}{2}\|v(t) - x^\star\|^2$ where $x^\star \in$ argmin f is a global minimizer of f. Again we note the similarity between the Lyapunov function proposed here and the estimating sequence $\{\phi_k(x)\}_{k=0}^{\infty}$ of function f introduced by Nesterov in its optimal first-order methods analysis Nesterov (2018). In (Nesterov, 2018, Lemma 2.2.3), this sequence that takes the form $\phi_k(x) = \phi_k^\star(x) + \frac{\gamma_k}{2}\|v_k - x\|^2$ where $\gamma_{k+1} := (1-\alpha_k)\gamma_k + \alpha_k\mu$ and $v_{k+1} := \frac{1}{\gamma_{k+1}}[(1-\alpha_k)\gamma_k v_k + \alpha_k\mu y_k - \alpha_k\nabla f(y_k)]$ which stand for a forward Euler discretization respectively of (8) and second ODE of (9).

We ask the attentive reader to remember that this discussion mainly concerns the continuous time case. A second central part of our analysis was based on the methods of discretization of (9). Indeed, these discretizations ensure, together with the spectral transformation (7), the optimal convergence rates of the methods and their particular ability to handle noisy gradients.

Finally, we delve into a crucial insight motivating the proposition of an optimizer that exhibits robustness concerning the choice of the step size, enabling the utilization of a wide range of values for the step size (or learning rate). An established approach for analyzing Stochastic Gradient Descent (SGD) involves viewing it as a discretization of a continuous-time process, expressed as:

$$\mathrm{d}x_t = -\nabla f(x_t)\mathrm{d}t + \sqrt{\alpha\sigma^2}\mathrm{d}B_t,$$

where B_t denotes the standard Brownian motion. This stochastic differential equation (SDE) is a variant of the well-known Langevin diffusion. Under mild regularity assumptions on f, it can be shown that the Markov process $(x_t)_{t\geq 0}$ is ergodic, with its unique invariant measure having a density proportional to $\exp\left(-f(x)/(\alpha\sigma^2)\right)$ for any $\alpha > 0$ (Roberts & Stramer, 2002).

Building on this observation, existing research has shed light on the relationship between the invariant measure and algorithm parameters. Jastrzębski et al. (2017) focused on the interplay of the invariant measure with step-size α and mini-batch size as a function of σ^2. They concluded that the ratio of learning rate to batch size serves as the control parameter determining the width of the minima found by SGD.

Additionally, Keskar et al. (2017) explored sharp and flat minimizers, their impact on generalization, and the distinctions between large-batch and small-batch methods, especially in deep neural networks. Their key observations include:

- Sharp and Flat Minimizers: Flat minimizers exhibit slow function variation in a wide neighborhood, while sharp minimizers show rapid increases in a small neighborhood. Flat minimizers can be described with lower precision, contrasting with the higher precision needed for sharp minimizers.
- Effect on Generalization: Sharp minimizers negatively affect model generalization due to their large sensitivity in the training function. The Minimum Description Length (MDL) theory suggests that lower complexity models generalize better, making flat minimizers preferable.
- Large-batch vs. Small-batch Methods: Large-batch methods tend to converge to sharp minimizers, reducing generalization ability. Conversely, small-batch methods converge to flat minimizers, generally leading to better generalization.
- Observation in Deep Neural Networks: The loss function landscape in deep neural networks attracts large-batch methods towards regions with sharp minimizers, trapping them and impeding their escape from these basins of attraction.

Motivated by these insights and assuming a Gaussian noise process, it becomes evident that proposing a solver capable of handling a broad range of step size values is crucial. Specifically, allowing for a larger ratio α/b (with b as the mini-batch size) increases the likelihood of converging to wider local minima, ultimately enhancing the generalization performance of the trained model.

A.1 Convergence/Stability analysis of the quadratic case: details

As briefly mentioned in Section 2.3 of the main text, the two key elements to come up with a maximum (constant) step size for Algorithm 1 are the study of the spectral radius of iteration matrix associated with NAG-GS scheme (Appendix A.1.1) and the covariance matrix at stationarity (Appendix A.1.2) w.r.t. all the significant parameters of the scheme. These parameters are the step size (integration step/time step) α, the convexity parameters $0 \leq \mu \leq L \leq \infty$ of the function $f(x)$, the variance of the noise σ^2 and the positive scaling parameter γ. Note that this theoretical study only concerns the case $f(x) = \frac{1}{2}x^\top Ax$.

Reproducibility

- In Appendix A.1.1, we start by determining the explicit formulation of the spectral radius of the iteration matrix $\rho(E(\alpha))$, specifically for the 2-dimensional quadratic case. This formulation allows us to derive the optimal step size α_c that minimizes $\rho(E(\alpha))$, resulting in the highest convergence rate for the NAG-GS method. Notably, Lemma 2 presents a crucial outcome for the asymptotic convergence analysis of NAG-GS, revealing that $\rho(E(\alpha))$ is a strictly monotonically increasing function of α within a certain interval, under mild assumptions.
- In Appendix A.1.2, we conduct an in-depth analysis of the covariance matrix at stationarity, which enables us to establish sufficient conditions for α_c to ensure the asymptotic convergence of the NAG-GS method. The formal proof for this convergence is presented in Lemma 3 for the case of $n = 2$.
- In Appendix A.1.4, we provide the formal proof of Theorem 1, which is enunciated in the main text. This theorem stated the asymptotic convergence of the NAG-GS method for dimensions $n > 2$.

A.1.1 Spectral radius analysis

Let us assume $f(x) = \frac{1}{2}x^\top Ax$ and since $A \in \mathbb{S}^n_+$ by hypothesis, it is diagonalizable and can be presented as $A = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ without loss of generality, that is to say, that we will consider a system of coordinates composed of the eigenvectors of matrix A. Let us note that $\mu = \lambda_1 \leq \ldots \leq \lambda_n = L$.

For the following, we restrict the discussion to the case $n = 2$. In this setting, $y = (x, v) \in \mathbb{R}^4$ and the matrices M and N from the Gauss-Seidel splitting of G_{NAG} (7) are:

$$M = \begin{bmatrix} -I_{2\times 2} & 0_{2\times 2} \\ \mu/\gamma I_{2\times 2} - A/\gamma & -\mu/\gamma I_{2\times 2} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -\mu/\gamma & 0 \\ 0 & \mu/\gamma - L/\gamma & 0 & -\mu/\gamma \end{bmatrix},$$

$$N = \begin{bmatrix} 0_{2\times 2} & I_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} \end{bmatrix}$$

For the minimization of $f(x) = \frac{1}{2}x^\top Ax$, given the property of Brownian motion $\Delta W_k = W_{k+1} - W_k = \sqrt{\alpha_k}\eta_k$ where $\eta_k \sim \mathcal{N}(0, 1)$, (12) reads:

$$y_{k+1} = (I_{4\times 4} - \alpha M)^{-1}(I_{4\times 4} + \alpha N)y_k + (I_{4\times 4} - \alpha M)^{-1} \begin{bmatrix} 0 \\ \sigma\sqrt{\alpha}\eta_k \end{bmatrix} \tag{16}$$

Since matrix M is lower-triangular, matrix $I_{4\times 4} - \alpha M$ is as well and can be factorized as follows:

$$\begin{aligned} I_{4\times 4} - \alpha M &= DT \\ &= \begin{bmatrix} (1+\alpha)I_{2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & (1 + \frac{\alpha\mu}{\gamma})I_{2\times 2} \end{bmatrix} \begin{bmatrix} I_{2\times 2} & 0_{2\times 2} \\ \frac{\alpha(A - \mu I_{2\times 2})}{\gamma(1+\frac{\alpha\mu}{\gamma})} & I_{2\times 2} \end{bmatrix} \end{aligned}$$

Hence $(I_{4\times 4} - \alpha M)^{-1} = T^{-1}D^{-1}$ where D^{-1} can be easily computed. It remains to compute T^{-1}; T can be decomposed as follows: $T = I_{4\times 4} + Q$ with Q a nilpotent matrix such that $QQ = O_{4\times 4}$. For such decomposition, it is well known that:

$$T^{-1} = (I_{4\times 4} + Q)^{-1} = I_{4\times 4} - Q = \begin{bmatrix} I_{2\times 2} & 0_{2\times 2} \\ \frac{\alpha(\mu I_{2\times 2} - A)}{\gamma(1+\tau_k)} & I_{2\times 2} \end{bmatrix} \tag{17}$$

where $\tau_k = \frac{\alpha\mu}{\gamma}$. Combining these results, equation 16 finally reads:

$$\begin{aligned} y_{k+1} &= \begin{bmatrix} \frac{1}{\alpha+1} & 0 & \frac{\alpha}{1+\alpha} & 0 \\ 0 & \frac{1}{\alpha+1} & 0 & \frac{\alpha}{1+\alpha} \\ 0 & 0 & \frac{1}{1+\tau} & 0 \\ 0 & \frac{\alpha(\mu-L)}{\gamma(\tau+1)(\alpha+1)} & 0 & \frac{\alpha^2(\mu-L)}{\gamma(1+\tau)(1+\alpha)} + \frac{1}{1+\tau} \end{bmatrix} y_k + \begin{bmatrix} 0 \\ \sigma\frac{\sqrt{\alpha}}{1+\tau}\eta_k \end{bmatrix} \\ &= E y_k + \begin{bmatrix} 0 \\ \sigma\frac{\sqrt{\alpha}}{1+\tau}\eta_k \end{bmatrix} \end{aligned} \tag{18}$$

with E denoting the iteration matrix associated with the NAG-GS method. Hence equation 18 includes two terms, the first is the product of the iteration matrix times the current vector y_k and the second one features the effect of the noise. For the latter, it will be studied in Appendix A.1.2 from the point of view of maximum step size for the NAG-GS method through the key quantity of the covariance matrix. Let us focus on the first term. It is clear that in order to get the maximum contraction rate, we should look for α that minimizes the spectral radius of E. Since the spectral radius is the maximum absolute value of the eigenvalues of iteration matrix E, we start by computing them. Let us find the expression of $\lambda_i \in \sigma(E)$ for $1 \leq i \leq 4$ that satisfies $\det(E - \lambda I_{4\times4}) = 0$ as functions of the scheme's parameters. Solving

$$\begin{aligned} &\det(E - \lambda I_{4\times4}) = 0 \\ &\equiv \frac{(\gamma\lambda - \gamma + \alpha\lambda\mu)(\lambda + \alpha\lambda - 1)(\gamma - 2\gamma\lambda + \gamma\lambda^2 + \alpha^2\lambda^2\mu - \alpha\gamma\lambda - \alpha\lambda\mu + L\alpha^2\lambda + \alpha\gamma\lambda^2 + \alpha\lambda^2\mu - \alpha^2\lambda\mu)}{(\alpha+1)^2(\gamma+\alpha\mu)^2} = 0 \end{aligned} \tag{19}$$

leads to the following eigenvalues:

$$\begin{aligned} \lambda_1 &= \frac{\gamma}{\gamma + \alpha\mu} \\ \lambda_2 &= \frac{1}{1+\alpha} \\ \lambda_3 &= \frac{2\gamma + \alpha\gamma + \alpha\mu - L\alpha^2 + \alpha^2\mu}{2(\gamma + \alpha\gamma + \alpha\mu + \alpha^2\mu)} + \\ &\frac{\alpha\sqrt{L^2\alpha^2 - 2L\alpha^2\mu - 2L\alpha\mu - 2\gamma L\alpha - 4\gamma L + \alpha^2\mu^2 + 2\alpha\mu^2 + 2\gamma\alpha\mu + \mu^2 + 2\gamma\mu + \gamma^2}}{2(\gamma + \alpha\gamma + \alpha\mu + \alpha^2\mu)} \\ \lambda_4 &= \frac{2\gamma + \alpha\gamma + \alpha\mu - L\alpha^2 + \alpha^2\mu}{2(\gamma + \alpha\gamma + \alpha\mu + \alpha^2\mu)} - \\ &\frac{\alpha\sqrt{L^2\alpha^2 - 2L\alpha^2\mu - 2L\alpha\mu - 2\gamma L\alpha - 4\gamma L + \alpha^2\mu^2 + 2\alpha\mu^2 + 2\gamma\alpha\mu + \mu^2 + 2\gamma\mu + \gamma^2}}{2(\gamma + \alpha\gamma + \alpha\mu + \alpha^2\mu)} \end{aligned} \tag{20}$$

Let us first mention some general behavior or these eigenvalues. Given γ and μ positive, we observe that:

1. λ_1 and λ_2 are positive decreasing functions w.r.t. α. Moreover, for bounded γ and μ, we have $\lim_{\alpha\to\infty} |\lambda_1(\alpha)| = 0 = \lim_{\alpha\to\infty} |\lambda_2(\alpha)|$.
2. One can show that for $\alpha \in [\frac{\mu+\gamma-2\sqrt{\gamma L}}{L-\mu}, \frac{\mu+\gamma+2\sqrt{\gamma L}}{L-\mu}]$, functions $\lambda_3(\alpha)$ and $\lambda_4(\alpha)$ are complex values and one can easily show that both share the same absolute value. Note that the lower bound of the interval $\frac{\mu+\gamma-2\sqrt{\gamma L}}{L-\mu}$ is negative as soon as $\gamma \in [2L - \mu - 2\sqrt{L^2 - \mu L}, 2L - \mu + 2\sqrt{L^2 - \mu L}] \subseteq \mathbb{R}_+$. Moreover, one can easily show that $\lim_{\alpha\to\infty} |\lambda_3(\alpha)| = 0$ and $\lim_{\alpha\to\infty} |\lambda_4(\alpha)| = \frac{L-\mu}{\mu} = \kappa(A) - 1$. The latter limit shows that eigenvalue λ_4 plays a central role in the convergence of the NAG-GS method since it is the one that can reach the value one and violate the convergence condition, as soon as $\kappa(A) > 2$. The analysis of λ_4 also allows us to come up with a good candidate for the step size α that minimizes the spectral radius of matrix E, especially and obviously at critical point $\alpha_{max} = \frac{\mu+\gamma+2\sqrt{\gamma L}}{L-\mu}$ which is positive since $L \geq \mu$ by hypothesis. Note that the case $L \to \mu$ gives some preliminary hints that the maximum step size can be almost "unbounded" in some particular cases.

Now, let us study these eigenvalues in more detail, it seems that three different scenarios must be studied:

1. For any variant of Algorithm 1 for which $\gamma_0 = \mu$, then $\gamma = \mu$ for all $k \geq 0$ and therefore $\lambda_1(\alpha) = \lambda_2(\alpha)$. Moreover, at $\alpha = \frac{\mu+\gamma+2\sqrt{\gamma L}}{L-\mu} = \frac{2\mu+2\sqrt{\mu L}}{L-\mu}$, we can easily check that $|\lambda_1(\alpha)| = |\lambda_2(\alpha)| = |\lambda_3(\alpha)| = |\lambda_4(\alpha)|$. Therefore $\alpha = \frac{2\mu+2\sqrt{\mu L}}{L-\mu}$ is the step size ensuring the minimal spectral radius and hence the maximum contraction rate. Figure 4 shows the evolution of the absolute values of the eigenvalues of iteration matrix E w.r.t. α for such a setting.

2. As soon as $\gamma < \mu$, one can easily show that $\lambda_1(\alpha) < \lambda_2(\alpha)$. Therefore the step size α with the minimal spectral radius is such that $|\lambda_4(\alpha)| = |\lambda_2(\alpha)|$. One can show that the equality holds for $\alpha = \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}$. One can easily check that $\frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu} - \frac{\mu+\gamma+2\sqrt{\gamma L}}{L-\mu} = (\mu-\gamma)^2 > 0$. Hence the second candidate for step size α will be bigger than the first one and the distance between them increases as the squared distance between γ and μ. Figure 5 shows the evolution of the absolute values of the eigenvalues of iteration matrix E w.r.t. α for this setting.

3. For $\gamma > \mu$: the analysis of this case gives the same results as the previous point. According to Algorithm 1, γ is either constant and equal to μ or decreasing to μ along iterations. Hence, the case $\gamma > \mu$ will be considered for the theoretical analysis when $\gamma \neq \mu$.

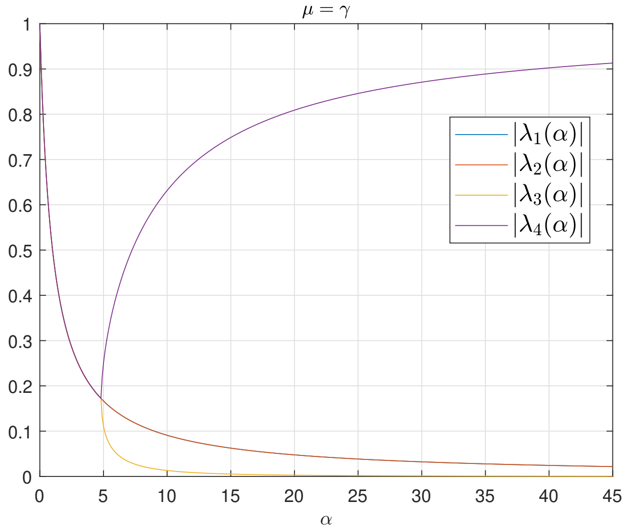


Figure 4: Evolution of absolute values of λ_i w.r.t α; $\mu = \gamma$.

As a first summary, the detailed analysis of the eigenvalues of iteration matrix E w.r.t. the significant parameters of the NAG-GS method leads us to come up with two candidates for the step size that minimize the spectral radius of E, hence ensuring the highest contraction rate possible. These results will be gathered with those obtained in Appendix A.1.2 dedicated to the covariance matrix analysis.

Let us now look at the behavior of the dynamics in expectation; given the properties of the Brownian motion and by applying the Expectation operator $\mathbb{E}$ on both sides of the system of SDE's (11), the resulting "averaged" equations identify with the "deterministic" setting studied by Luo & Chen (2021). For such a setting, authors from Luo & Chen (2021) demonstrated that, if $0 \leq \alpha \leq \frac{2}{\sqrt{\kappa(A)}}$, then a Gauss–Seidel splitting-based scheme for solving (9) is A-stable for quadratic objectives in the deterministic setting. We conclude this section by showing that the two candidates we derived above for step size are higher than the limit $\frac{2}{\sqrt{\kappa(A)}}$ given in (Luo & Chen, 2021, Theorem 1). It can be intuitively understood in the case $L \to \mu$, however, we give a formal proof in Lemma 1.

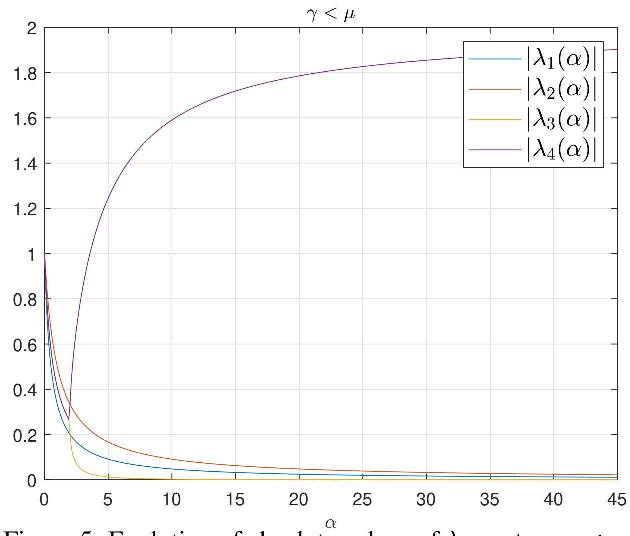


Figure 5: Evolution of absolute values of λ_i w.r.t α; $\gamma < \mu$.

Lemma 1. *Given $\gamma > 0$, and assuming $0 < \mu < L$, then for $\gamma = \mu$ and $\gamma > \mu$ the following inequalities respectively hold:*

$$\begin{aligned}\frac{2\mu + 2\sqrt{\mu L}}{L-\mu} &> \frac{2}{\sqrt{\kappa(A)}} \\ \frac{\mu + \gamma + \sqrt{(\mu-\gamma)^2 + 4\gamma L}}{L-\mu} &> \frac{2}{\sqrt{\kappa(A)}}\end{aligned} \tag{21}$$

where $\kappa(A) = \frac{L}{\mu}$.

Proof. Let us start for the case $\mu = \gamma$, hence first inequality from equation 21 becomes:

$$\begin{aligned}&\frac{2\mu + 2\sqrt{L\mu}}{L-\mu} > \frac{2}{\sqrt{L/\mu}} \\ \equiv&(\mu + \sqrt{L\mu})\sqrt{L/\mu} > (L-\mu) \\ \equiv&\sqrt{\mu L} + L > L - \mu \\ \equiv&\sqrt{\mu L} > -\mu\end{aligned}$$

which holds for any positive μ, L and satisfied by hypothesis. For the case $\gamma > \mu$, we have:

$$\begin{aligned}&\frac{\mu + \gamma + \sqrt{(\mu-\gamma)^2 + 4\gamma L}}{L-\mu} > \frac{2}{\sqrt{L/\mu}} \\ \equiv&\sqrt{(\mu-\gamma)^2 + 4\gamma L} > \frac{2}{\sqrt{L/\mu}}(L-\mu) - \gamma - \mu \\ \equiv&(\mu-\gamma)^2 + 4\gamma L > (\mu + 2\sqrt{\frac{\mu}{L}}(\mu - L) + \gamma)^2 \\ \equiv&\gamma > \frac{-2\mu^2 + \mu^3/L + \mu^2\sqrt{\mu/L} + \mu L - \mu L\sqrt{\mu/L}}{-\mu - \sqrt{\mu/L}(\mu - L) + L}\end{aligned}$$

where second inequality hold since $L \geq \mu$ and last inequality holds since $-\mu - \sqrt{\mu/L}(\mu - L) + L > 0$ (one can easily check this by using $L > \mu$). It remains to show that:

$$\mu > \frac{-2\mu^2 + \mu^3/L + \mu^2\sqrt{\mu/L} + \mu L - \mu L\sqrt{\mu/L}}{-\mu - \sqrt{\mu/L}(\mu - L) + L}$$

which holds for any μ and L positive (technical details are skipped; it mainly consists of the study of a table of signs of a polynomial equation in μ).

Since $\gamma > \mu$ by hypothesis, therefore inequality

$$\gamma > \frac{-2\mu^2 + \mu^3/L + \mu^2\sqrt{\mu/L} + \mu L - \mu L\sqrt{\mu/L}}{-\mu - \sqrt{\mu/L}(\mu - L) + L}$$

holds for any μ and L positive as well, conditions satisfied by hypothesis. This concludes the proof.

□

Furthermore, let us note that both step size candidates, that are $\{\frac{2\mu+2\sqrt{\mu L}}{L-\mu}, \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}\}$ respectively for the cases $\gamma = \mu$ and $\gamma > \mu$ show that NAG-GS method converges in the case $L \to \mu$ with a step size that tends to ∞, this behavior cannot be anticipated by the upper-bound given by (Luo & Chen, 2021, Theorem 1). Some simple numerical experiments are performed in Appendix A.1.5 to support this theoretical result.

Finally, based on previous discussions, let us remark that for $\alpha \in [\frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}, \infty]$ when $\gamma \neq \mu$ or $\alpha \in [\frac{2\mu+2\sqrt{\mu L}}{L-\mu}, \infty]$ when $\gamma = \mu$, we have $\rho(E(\alpha)) = |\lambda_4(\alpha)|$ and one can show that $\rho(E)$ is strictly monotonically increasing function of α for all $L > \mu > 0$ and $\gamma > 0$, see Lemma 2 for the formal proof.

Lemma 2. *Given $\gamma > 0$, and assuming $0 < \mu < L$, then for $\gamma = \mu$ and $\gamma > \mu$, the spectral radius $\rho(E(\alpha))$ is a strict monotonic increasing function of α for $\alpha \in [\alpha_c, \infty]$ with $\alpha_c = \frac{2\mu+2\sqrt{\mu L}}{L-\mu}$ or $\alpha_c = \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}$.*

Proof. Let us first recall that on $[\alpha_c, \infty]$, the spectral radius $\rho(E(\alpha))$ is equal to $|\lambda_4|$, the expression of λ_4 as a function of parameters of interests for the convergence analysis of NAG-GS method was given in equation 20 and recalled here-under for convenience:

$$\lambda_4 = \frac{2\gamma + \alpha\gamma + \alpha\mu - L\alpha^2 + \alpha^2\mu}{2(\gamma + \alpha\gamma + \alpha\mu + \alpha^2\mu)} - \\ \frac{\alpha\sqrt{L^2\alpha^2 - 2L\alpha^2\mu - 2L\alpha\mu - 2\gamma L\alpha - 4\gamma L + \alpha^2\mu^2 + 2\alpha\mu^2 + 2\gamma\alpha\mu + \mu^2 + 2\gamma\mu + \gamma^2}}{2(\gamma + \alpha\gamma + \alpha\mu + \alpha^2\mu)} \tag{22}$$

Let start by showing that λ_4 is negative on $[\alpha_c, \infty]$. Firstly, one can easily observe that the denominator of λ_4 is positive, secondly let us compute the values for α such that:

$$\begin{aligned} &2\gamma + \alpha\gamma + \alpha\mu - L\alpha^2 + \alpha^2\mu - \\ &\alpha\sqrt{L^2\alpha^2 - 2L\alpha^2\mu - 2L\alpha\mu - 2\gamma L\alpha - 4\gamma L + \alpha^2\mu^2 + 2\alpha\mu^2 + 2\gamma\alpha\mu + \mu^2 + 2\gamma\mu + \gamma^2} = 0 \\ \equiv &- 4\gamma^2 - 4\alpha\gamma(\mu + \gamma) + \alpha^2(\gamma^2 - 4\gamma L + 2\gamma\mu + \mu^2) - \alpha^2(\gamma^2 - 4\gamma L + 6\gamma\mu + \mu^2) = 0 \\ \equiv &(-4\gamma\mu)\alpha^2 - 4\gamma(\mu + \gamma)\alpha - 4\gamma^2 = 0 \end{aligned} \tag{23}$$

The expression above is negative as soon as $\alpha < -1$ or $\alpha > \frac{-\gamma}{\mu} < 0$ since $\gamma, \mu > 0$ by hypothesis. The latter is always satisfied since $\alpha \geq \alpha_c > 0$ by hypothesis. Therefore $\rho(E(\alpha)) = -\lambda_4$ for $\alpha \in [\alpha_c, \infty]$.

To show the monotonic increasing behavior of $\rho(E(\alpha))$ w.r.t. $\alpha \in [\alpha_c, \infty]$, it remains to show that:

$$\frac{d(\rho(E(\alpha))}{d\alpha} = \frac{d(-\lambda_4)}{d\alpha} > 0. \tag{24}$$

To ease the analysis, let us decompose $-\lambda_4(\alpha) = t_1(\alpha) + t_2(\alpha)$ such that:

$$\begin{aligned} t_1(\alpha) &= -\frac{2\gamma + \alpha\gamma + \alpha\mu - L\alpha^2 + \alpha^2\mu}{2(\gamma + \alpha\gamma + \alpha\mu + \alpha^2\mu)} \\ t_2(\alpha) &= \frac{\alpha\sqrt{L^2\alpha^2 - 2L\alpha^2\mu - 2L\alpha\mu - 2\gamma L\alpha - 4\gamma L + \alpha^2\mu^2 + 2\alpha\mu^2 + 2\gamma\alpha\mu + \mu^2 + 2\gamma\mu + \gamma^2}}{2(\gamma + \alpha\gamma + \alpha\mu + \alpha^2\mu)} \end{aligned} \tag{25}$$

Let us now show that $\frac{dt_1(\alpha)}{d\alpha} > 0$ and $\frac{dt_2(\alpha)}{d\alpha} > 0$ for any $L > \mu > 0$. We first obtain:

$$\begin{aligned} \frac{dt_1(\alpha)}{d\alpha} &= \frac{(2\gamma + 2\mu + 4\alpha\mu)(2\gamma + \alpha\gamma + \alpha\mu - L\alpha^2 + \alpha^2\mu)}{(2\gamma + 2\alpha\gamma + 2\alpha\mu + 2\alpha^2\mu)^2} - \\ &\quad \frac{\gamma + \mu - 2L\alpha + 2\alpha\mu}{2\gamma + 2\alpha\gamma + 2\alpha\mu + 2\alpha^2\mu} \\ &= \frac{(L\alpha^2 + \gamma)(\gamma + \mu) + 2\alpha\gamma(L + \mu)}{2(\alpha + 1)^2(\gamma + \alpha\mu)^2} \end{aligned} \tag{26}$$

which is strictly positive since $L > \mu > 0$ and $\gamma > 0$ by hypothesis. Furthermore:

$$\begin{aligned} &\frac{dt_2(\alpha)}{d\alpha} = \\ &\frac{(\gamma + \mu)(L - \mu)(\alpha^3 L - 3\alpha\gamma) + \alpha^2(L(-\gamma^2 - \mu^2) + 2\gamma(L^2 - L\mu + \mu^2)) + \gamma(\gamma^2 - 2\gamma(2L - \mu) + \mu^2)}{2(\alpha + 1)^2(\alpha\mu + \gamma)^2\sqrt{\alpha^2(L^2 - 2L\mu + \mu^2) - 2\alpha(\gamma + \mu)(L - \mu) + \gamma^2 - 2\gamma(2L - \mu) + \mu^2}} \end{aligned} \tag{27}$$

The remaining demonstration is significantly long and technically heavy in the case $\gamma > \mu$. Then we limit the last part of the demonstration for the case $\mu = \gamma$ for which we have shown previously than $\alpha_c = \frac{\mu + \gamma + 2\sqrt{\gamma L}}{L - \mu} = \frac{2\mu + 2\sqrt{\mu L}}{L - \mu}$. In practice, with respect to the NAG-GS method summarized by Algorithm 1, γ quickly decreases to μ and equality $\mu = \gamma$ holds for the most part of the iterations of the Algorithm, hence this case is more important to detail here. However, the reasoning explained herein ultimately leads to identical final conclusions when considering the case where γ is greater than μ.

The first term of the numerator of Equation 27 is positive as soon as $\alpha \geq \sqrt{\frac{3\gamma}{L}}$. In the case $\mu = \gamma$, we determine the conditions under which the second term of the numerator of Equation 27 is positive, that is:

$$\begin{aligned} &\alpha^2(L(-2\mu^2) + 2\mu(L^2 - L\mu + \mu^2)) + \mu(2\mu^2 - 2\mu(2L - \mu)) > 0 \\ &\equiv \alpha^2(L(-2\mu^2) + 2\mu(L^2 - L\mu + \mu^2)) > \mu(-2\mu^2 + 2\mu(2L - \mu)) \end{aligned} \tag{28}$$

First one can see that:

$$\begin{aligned} &(L(-2\mu^2) + 2\mu(L^2 - L\mu + \mu^2)) > 0, \\ &\mu(-2\mu^2 + 2\mu(2L - \mu)) > 0 \end{aligned} \tag{29}$$

hold as soon as $L > \mu > 0$ which is satisfied by hypothesis. Therefore, the second term of the numerator of Equation 27 is positive as soon as

$$\alpha > \sqrt{\frac{\mu(-2\mu^2 + 2\mu(2L - \mu))}{(L(-2\mu^2) + 2\mu(L^2 - L\mu + \mu^2))}} = \sqrt{\frac{2\mu}{L - \mu}} \tag{30}$$

which exists since $L > \mu > 0$ by hypothesis (the second root of equation 29 being negative). Finally, since $\alpha \in [\alpha_c, \infty]$ by hypothesis, $\frac{dt_2(\alpha)}{d\alpha}$ is positive as soon as:

$$\begin{aligned} \alpha_c &> \sqrt{\frac{3\mu}{L}} \\ \alpha_c &> \sqrt{\frac{2\mu}{L - \mu}} \end{aligned} \tag{31}$$

hold with $\alpha_c = \frac{2\mu + 2\sqrt{\mu L}}{L - \mu}$. One can easily show that both inequalities hold as soon as $L > \mu > 0$ which is satisfied by the hypothesis. This concludes the proof of the strict increasing monotonicity of $\rho(E(\alpha))$ w.r.t. α for $\alpha \in [\alpha_c, \infty]$ assuming $L > \mu > 0$ and $\gamma = \mu$.

□

A.1.2 Covariance analysis

In this section, we study the contribution to the computation of maximum step size for the NAG-GS method through the analysis of the covariance matrix at stationarity. Let us start by computing the covariance matrix C obtained at iteration $k+1$ from Algorithm 1:

$$C_{k+1} = \mathbb{E}(y_{k+1}y_{k+1}^T) \tag{32}$$

By denoting $\xi_k = \begin{bmatrix} 0 \\ \sigma \frac{\sqrt{\alpha}}{1+\tau}\eta_k \end{bmatrix}$, let us replace y_{k+1} by its expression given in equation 18, equation 32 writes:

$$\begin{aligned} C_{k+1} &= \mathbb{E}(y_{k+1}y_{k+1}^T) \\ &= \mathbb{E}\left((Ey_k + \xi_k)(Ey_k + \xi_k)^T\right) \\ &= \mathbb{E}\left(Ey_k y_k^T E^T\right) + \mathbb{E}\left(\xi_k \xi_k^T\right) \end{aligned} \tag{33}$$

which holds since expectation operator $\mathbb{E}(.)$ is a linear operator and by assuming statistical independence between ξ_k and Ey_k. On the one hand, by using again the properties of linearity of $\mathbb{E}$ and since E is seen as a constant by $\mathbb{E}(.)$, one can show that $\mathbb{E}\left(Ey_k y_k^T E^T\right) = EC_kE^T$. On the other hand, since $\eta_k \sim \mathcal{N}(0,1)$, then Equation equation 33 becomes:

$$C_{k+1} = EC_kE^T + Q \tag{34}$$

where $Q = \begin{bmatrix} 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & \frac{\alpha_k\sigma^2}{(1+\tau_k)^2} I_{2\times2} \end{bmatrix}$. Let us now look at the limiting behavior of Equation equation 34, that is $\lim_{k\to\infty} C_k$. Let be $C = \lim_{k\to\infty} C_k$ the covariance matrix reached in the asymptotic regime, also referred to as stationary regime. Applying the limit on both sides of Equation equation 34, C then satisfies

$$C = ECE^T + Q \tag{35}$$

Hence equation 35 is a particular case of discrete Lyapunov equation. For solving such equation, the vectorization operator denoted $\vec{.}$is applied on both sides on equation 35, this amounts to solve the following linear system:

$$(I_{4^2\times4^2} - E\otimes E)\vec{C} = \vec{Q} \tag{36}$$

where $A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}$ stands for the Kronecker product. The solution is given by:

$$C = \overleftarrow{(I_{4^2\times4^2} - E\otimes E)^{-1}\vec{Q}} \tag{37}$$

where $\overleftarrow{a}$ stands for the un-vectorized operator.

Let us note that, even for the 2-dimensional case considered in this section, the dimension of matrix C rapidly growth and cannot be written in plain within this paper. For the following, we will keep its symbolic expression. The stationary matrix C quantifies the spreading of the limit of the sequence $\{y_k\}$, as a direct consequence of the Brownian motion effect. Now we look at the directions that maximize the scattering of the points, in other words, we are looking for the eigenvectors and the associated eigenvalues of C. Actually, the required information for the analysis of the step size is contained within the expression of the eigenvalues $\lambda_i(C)$. The obtained eigenvalues are rationale functions w.r.t. the parameters of the schemes, while their numerator brings less interest for us (supported further), we will focus on their denominator. We obtained the following expressions:

$$\begin{aligned} \lambda_1(C) &= \frac{N_1(\alpha,\mu,L,\gamma,\sigma)}{D_1(\alpha,\mu,L,\gamma,\sigma)}, \\ \text{s.t. } D_1(\alpha,\mu,L,\gamma,\sigma) &= -L^2\alpha^3\mu - L^2\alpha^2\mu - \gamma L^2\alpha^2 + 2L\alpha^3\mu^2 + 4L\alpha^2\mu^2 + \\ &\quad 4\gamma L\alpha^2\mu + 2L\alpha\mu^2 + 8\gamma L\alpha\mu + 2\gamma^2 L\alpha + 4\gamma L\mu + 4\gamma^2 L \end{aligned} \tag{38}$$

$$\begin{aligned} \lambda_2(C) &= \frac{N_2(\alpha,\mu,L,\gamma,\sigma)}{D_2(\alpha,\mu,L,\gamma,\sigma)}, \\ \text{s.t. } D_2(\alpha,\mu,L,\gamma,\sigma) &= \alpha^3\mu^3 + 3\alpha^2\mu^3 + 3\gamma\alpha^2\mu^2 + 2\alpha\mu^3 + \\ &\quad 8\gamma\alpha\mu^2 + 2\gamma^2\alpha\mu + 4\gamma\mu^2 + 4\gamma^2\mu \end{aligned} \tag{39}$$

$$\begin{aligned}\lambda_3(C) &= \frac{N_3(\alpha,\mu,L,\gamma,\sigma)}{D_3(\alpha,\mu,L,\gamma,\sigma)}, \\ \text{s.t. } D_3(\alpha,\mu,L,\gamma,\sigma) &= \alpha^3\mu^3 + 3\alpha^2\mu^3 + 3\gamma\alpha^2\mu^2 + 2\alpha\mu^3 + \\ &\quad 8\gamma\alpha\mu^2 + 2\gamma^2\alpha\mu + 4\gamma\mu^2 + 4\gamma^2\mu\end{aligned} \tag{40}$$

$$\begin{aligned}\lambda_4(C) &= \frac{N_4(\alpha,\mu,L,\gamma,\sigma)}{D_4(\alpha,\mu,L,\gamma,\sigma)}, \\ \text{s.t. } D_4(\alpha,\mu,L,\gamma,\sigma) &= -L^2\alpha^3\mu - L^2\alpha^2\mu - \gamma L^2\alpha^2 + 2L\alpha^3\mu^2 + 4L\alpha^2\mu^2 + \\ &\quad 4\gamma L\alpha^2\mu + 2L\alpha\mu^2 + 8\gamma L\alpha\mu + 2\gamma^2 L\alpha + 4\gamma L\mu + 4\gamma^2 L\end{aligned} \tag{41}$$

One can observe that:

1. Given α, L, μ, γ positive, the denominators of eigenvalues λ_2 and λ_3 are positive as well, unlike eigenvalues λ_1 and λ_4 for which some vertical asymptotes may appear. The latter will be studied in more detail further. Note that, even if some eigenvalues share the same denominator, it is not the case for the numerator. This will be illustrated later in Figures 8 and 9 to ease the analysis.
2. Interestingly, the volatility of the noise defined by the parameter σ does not appear within the expressions of the denominators. It gives us a hint that these vertical asymptotes are due to the fact that spectral radius is getting close to 1 (discussed further in Appendix A.1.3). Moreover, the parameter σ appears only within the numerators and based on intensive numerical tests, this parameter has a pure scaling effect onto the eigenvalues $\lambda_i(C)$ when studied w.r.t. α without modifying the trends of the curves.

Let us now study in more details the denominator of λ_1 and λ_4 and seek for critical step size as a function of γ, μ and L at which a vertical asymptote may appear by solving:

$$\begin{aligned}&-L^2\alpha^3\mu - L^2\alpha^2\mu - \gamma L^2\alpha^2 + 2L\alpha^3\mu^2 + 4L\alpha^2\mu^2 + \\ &4\gamma L\alpha^2\mu + 2L\alpha\mu^2 + 8\gamma L\alpha\mu + 2\gamma^2 L\alpha + 4\gamma L\mu + 4\gamma^2 L = 0 \\ &\equiv \mu(2\mu - L)\alpha^3 + (\mu+\gamma)(4\mu - L)\alpha^2 + (2\mu^2 + 8\gamma\mu + 2\gamma^2)\alpha + 4\gamma(\mu+\gamma) = 0\end{aligned} \tag{42}$$

This polynomial equation in α has three roots:

$$\begin{aligned}\alpha_1 &= \frac{-\gamma-\mu}{\mu}, \\ \alpha_2 &= \frac{\mu+\gamma-\sqrt{\gamma^2 - 6\gamma\mu + \mu^2 + 4\gamma L}}{L-2\mu}, \\ \alpha_3 &= \frac{\mu+\gamma+\sqrt{\gamma^2 - 6\gamma\mu + \mu^2 + 4\gamma L}}{L-2\mu}.\end{aligned} \tag{43}$$

First, it is obvious that the first root α_1 is negative given γ, μ assumed nonnegative and therefore can be disregarded. Concerning α_2 and α_3, those are real roots as soon as:

$$\begin{aligned}&\gamma^2 - 6\gamma\mu + \mu^2 + 4\gamma L \geq 0 \\ \equiv&(\gamma-\mu)^2 - 4\gamma\mu + 4\gamma L \geq 0 \\ \equiv&(\gamma-\mu)^2 \geq 4\gamma(\mu - L)\end{aligned} \tag{44}$$

which is always satisfied since $\gamma > 0$ and $0 < \mu < L$ by hypothesis.

Further, it is obvious that the study must include three scenarios:

1. Scenario 1: $L - 2\mu < 0$, or equivalently $\mu > L/2$. Given μ and γ positive by hypothesis, it implies that α_3 is negative and hence can be disregarded. It remains to check if α_2 can be positive, it amounts to verifying if

$$\begin{aligned}&\mu+\gamma-\sqrt{\gamma^2 - 6\gamma\mu + \mu^2 + 4\gamma L} < 0 \\ \equiv&(\mu+\gamma)^2 < \gamma^2 - 6\gamma\mu + \mu^2 + 4\gamma L \\ \equiv&\mu < \frac{L}{2}\end{aligned}$$

which never holds by hypothesis. Therefore, for the first scenario, there is no positive critical step size at which a vertical asymptote for the eigenvalues may appear.

2. Scenario 2: $L - 2\mu > 0$, or equivalently $\mu < L/2$. Obviously, α_3 is positive and hence shall be considered for the analysis of maximum step size for our NAG-GS method. It remains to check if α_2 is positive, that is to verify if the numerator can be negative. We have seen in the first scenario that α_2 is negative as soon as $\mu < \frac{L}{2}$ which is verified by hypothesis. Therefore, only α_3 is positive.
3. Scenario 3: $L - 2\mu = 0$. For such a situation, the critical step size is located at ∞ and can be disregarded as a potential limitation in our study.

In summary, a potentially critical and limiting step size only exists in the case $\mu < L/2$, or equivalently if $\kappa(A) > 2$. In this setting, the critical step size is positive and is equal to $\alpha_{\text{crit}} = \frac{\mu+\gamma+\sqrt{\gamma^2-6\gamma\mu+\mu^2+4\gamma L}}{L-2\mu}$. Figures 6 to 7 display the evolution of the eigenvalues $\lambda_i(C)$ for $1 \leq i \leq 4$ w.r.t. to α for the two first scenarios, that are for $\mu > L/2$ and $\mu < L/2$. For the first scenario, the parameters σ, γ, μ and L have been respectively set to $\{1, 3/2, 1, 3/2\}$. For the second scenario, σ, γ, μ and L have been respectively set to $\{1, 3/2, 1, 3\}$. As expected, one can observe in Figure 6 that no vertical asymptote is present. Furthermore, one can observe $\lambda_i(C)$ seem to converge to some limit point when $\alpha \to \infty$, numerically we report that this limit point is zero, for all the values of γ and σ considered.

Finally, again as expected by the results presented in this section, Figure 7 shows the presence of two vertical asymptotes for the eigenvalues λ_1 and λ_4, and none for λ_2 and λ_3. Moreover, the critical step size is approximately located at $\alpha = 6$ which aligns with analytical expression $\alpha_{\text{crit}} = \frac{\mu+\gamma+\sqrt{\gamma^2-6\gamma\mu+\mu^2+4\gamma L}}{L-2\mu}$. Finally, one can observe that, after the vertical asymptotes, all the eigenvalues converge to some limit points, again numerically we report that this limit point is zero, for all the values of γ and σ considered.

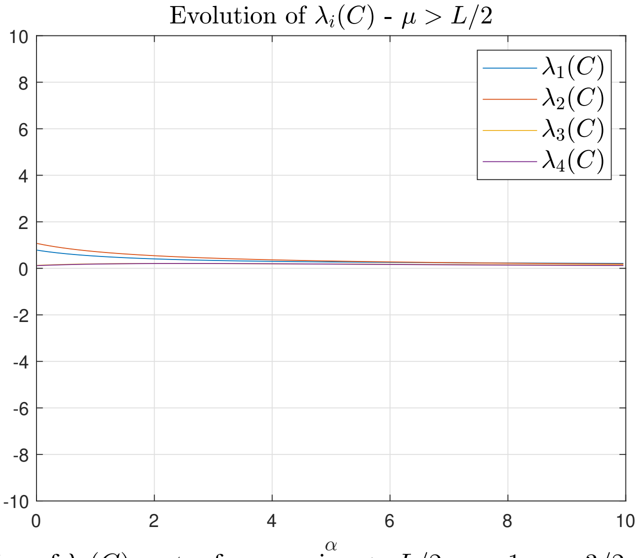


Figure 6: Evolution of $\lambda_i(C)$ w.r.t α for scenario $\mu > L/2$; $\sigma = 1$, $\gamma = 3/2$, $\mu = 1$, $L = 3/2$.

A.1.3 A CONCLUSION FOR THE 2-DIMENSIONAL CASE

In Appendix A.1.1 and Appendix A.1.2, several theoretical results have been derived for coming up with appropriate choices of constant step size for Algorithm 1. Key insights and interesting values for the step size have been discussed from the study of the spectral radius of iteration matrix E and through the analysis of the covariance matrix in the asymptotic regime. Let us summarize the theoretical results obtained:

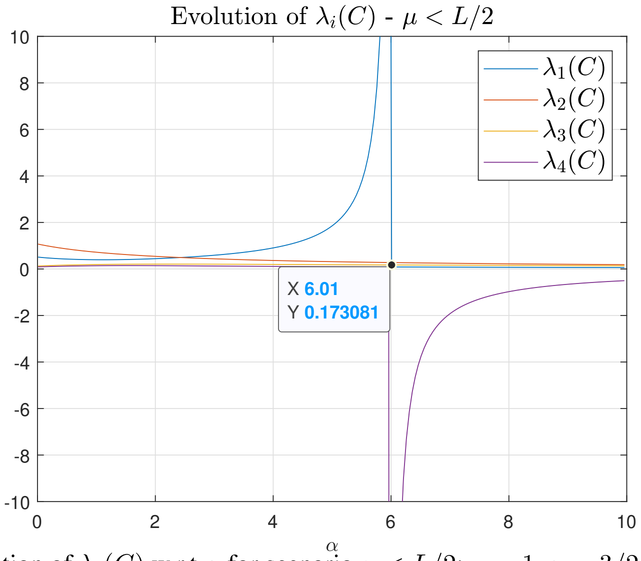


Figure 7: Evolution of $\lambda_i(C)$ w.r.t α for scenario $\mu < L/2$; $\sigma = 1$, $\gamma = 3/2$, $\mu = 1$, $L = 3$.

- from the spectral radius analysis of iteration matrix E; two scenarios have been highlighted, that are:
  1. case $\gamma = \mu$: the step size α that minimizes the spectral radius of matrix E is $\alpha = \frac{2\mu+2\sqrt{\mu L}}{L-\mu}$,
  2. case $\gamma > \mu$: the step size α that minimizes the spectral radius of matrix E is $\alpha = \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}$.
- from the analysis of covariance matrix C at stationarity: in the case $L - 2\mu > 0$, or equivalently $\mu < L/2$, we have seen that there is a vertical asymptote for two eigenvalues of C at $\alpha_{\text{crit}} = \frac{\mu+\gamma+\sqrt{\gamma^2-6\gamma\mu+\mu^2+4\gamma L}}{L-2\mu}$, leading to an intractable scattering of the limit points $\{y_k\}_{k\to\infty}$ generated by Algorithm 1. In the case $\mu > L/2$, there is no positive critical step size at which a vertical asymptote for the eigenvalues may appear.

Therefore, for quadratic functions such that $\mu > L/2$, we can safely choose either $\alpha = \frac{2\mu+2\sqrt{\mu L}}{L-\mu}$ when $\gamma = \mu$ either $\alpha = \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}$ when $\gamma > \mu$ to get the minimal spectral radius for iteration matrix E and hence the highest contraction rate for the NAG-GS method.

For quadratic functions such that $\mu < L/2$, we must show that the NAG-GS method is stable for both step sizes. Let us denote by $\alpha_c = \{\frac{2\mu+2\sqrt{\mu L}}{L-\mu}, \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}\}$, two values of step size for the two scenarios $\gamma = \mu$ and $\gamma > \mu$. In Lemma 3, we show that NAG-GS is asymptotically convergent, or stable, for the 2-dimensional case under mild assumptions in the case $\mu < L/2$.

Lemma 3. *Given $\gamma > 0$, and assuming $0 < \mu < L/2$, then for $\gamma = \mu$ and $\gamma > \mu$ the following inequalities respectively hold:*

$$\begin{aligned} \frac{\mu+\gamma+\sqrt{\gamma^2-6\gamma\mu+\mu^2+4\gamma L}}{L-2\mu} &> \frac{2\mu+2\sqrt{\mu L}}{L-\mu} \\ \frac{\mu+\gamma+\sqrt{\gamma^2-6\gamma\mu+\mu^2+4\gamma L}}{L-2\mu} &> \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu} \end{aligned} \tag{45}$$

Thus, in the 2-dimensional case, NAG-GS is asymptotically convergent (or stable) when choosing $\alpha_c = \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}$ or $\alpha_c = \frac{2\mu+2\sqrt{\mu L}}{L-\mu}$ respectively for the cases $\gamma > \mu$ and $\gamma = \mu$.

Proof. In order to prove the asymptotic stability or convergence of NAG-GS for the 2-dimensional case within the set of assumptions detailed above, one must show that $\rho(E(\alpha_c)) < 1$ for the two choices of α_c.

Let us start by computing α such that $\rho(E(\alpha)) = 1$. As proved in Lemma 2, for $\alpha \in [\alpha_c, \infty]$, $\rho(E(\alpha)) = -\lambda_4$ with λ_4 given in equation 20, we then have to compute α such that:

$$-\lambda_4 = -\frac{2\gamma + \alpha\gamma + \alpha\mu - L\alpha^2 + \alpha^2\mu}{2(\gamma + \alpha\gamma + \alpha\mu + \alpha^2\mu)} + \frac{\alpha(L^2\alpha^2 - 2L\alpha^2\mu - 2L\alpha\mu - 2\gamma L\alpha - 4\gamma L + \alpha^2\mu^2 + 2\alpha\mu^2 + 2\gamma\alpha\mu + \mu^2 + 2\gamma\mu + \gamma^2)^{1/2}}{2(\gamma + \alpha\gamma + \alpha\mu + \alpha^2\mu)} = 1.$$

This leads to computing the roots of a quadratic polynomial equation in α, the positive root is:

$$\alpha = \frac{\gamma + \mu + \sqrt{4L\gamma + \gamma^2 - 6\gamma\mu + \mu^2}}{L - 2\mu}, \tag{46}$$

which not surprisingly identifies to α_{crit} from the covariance matrix analysis [2].

Furthermore, as per Lemma 2, $\rho(E(\alpha))$ is strictly monotonically increasing function over the interval $[\alpha_c, \infty]$. Therefore, showing that $\rho(E(\alpha_c)) < 1$ is equivalent to show that α_c is strictly lower than $\alpha_{\text{crit}} := \frac{\gamma+\mu+\sqrt{4L\gamma+\gamma^2-6\gamma\mu+\mu^2}}{L-2\mu}$.

Let us focus on the case $\gamma > \mu$; since $0 < \mu < L/2$ by hypothesis, the second inequality from equation 45 can be written as:

$$\begin{aligned}
&(L-\mu)(\gamma + \mu + \sqrt{(\gamma-\mu)^2 + 4\gamma(L-\mu)}) - (L-2\mu)(\gamma + \mu + \sqrt{(\gamma-\mu)^2 + 4\gamma L}) > 0 \\
\equiv&\gamma\mu + \mu^2 + (L-\mu)\sqrt{\gamma^2 + \mu^2 + \gamma(4L - 6\mu)} + (2\mu - L)\sqrt{(\gamma-\mu)^2 + 4\gamma L} > 0
\end{aligned}$$

Given $\gamma, \mu > 0$, it remains to show that:

$$(L-\mu)\sqrt{\gamma^2 + \mu^2 + \gamma(4L-6\mu)} + (2\mu - L)\sqrt{(\gamma-\mu)^2 + 4\gamma L} > 0 \tag{47}$$

In order to show this, we study the conditions for γ such that the left-hand side of equation 47 is positive. With simple manipulations, one can show that canceling the left-hand side of equation 47 boils down to canceling the following quadratic polynomial:

$$\begin{aligned}
(L-\mu)\sqrt{\gamma^2 + \mu^2 + \gamma(4L-6\mu)} + (2\mu - L)\sqrt{(\gamma-\mu)^2 + 4\gamma L} = 0 \\
\equiv (-3\mu + 2L)\gamma^2 + (2\mu^2 - 8L\mu + 4L^2)\gamma + 2L\mu^2 - 3\mu^3 = 0
\end{aligned}$$

The two roots are:

$$\begin{aligned}
\gamma_1 &= \frac{-\mu^2 - 2L^2 - 2\sqrt{-2\mu^4 + L^4 - 4\mu L^3 + 4\mu^2L^2 + \mu^3 L} + 4\mu L}{2L - 3\mu} \\
\gamma_2 &= \frac{-\mu^2 - 2L^2 + 2\sqrt{-2\mu^4 + L^4 - 4\mu L^3 + 4\mu^2L^2 + \mu^3 L} + 4\mu L}{2L - 3\mu},
\end{aligned}$$

which are real and distinct as soon as:

$$\begin{aligned}
&-2\mu^4 + L^4 - 4\mu L^3 + 4\mu^2L^2 + \mu^3L > 0 \\
\equiv&(L-2\mu)(L-\mu)(-\mu^2 + L^2 - \mu L) > 0,
\end{aligned}$$

which holds since $0 < \mu < L/2$ by hypothesis (one can easily show that $-\mu^2 + L^2 - \mu L$ is positive in such setting). Moreover, the denominator $2L - 3\mu$ is strictly positive since $0 < \mu < L/2$. One can check that γ_1 is negative for all $\gamma, L > 0$ and $0 < \mu < L/2$ (simply show that $-\mu^2 - 2L^2 + 4\mu L$ is negative) and can be disregarded since γ is positive by hypothesis. Therefore, proving that equation 47 holds is equivalent to show that:

$$\gamma > \frac{-\mu^2 - 2L^2 + 2\sqrt{(L-2\mu)(L-\mu)(-\mu^2 + L^2 - \mu L)} + 4\mu L}{2L - 3\mu} \tag{48}$$

[2] It explains why the critical α does not include σ, this singularity is due to the spectral radius reaching the value 1.

To achieve this, let us first show that

$$\begin{aligned}
&\mu > \frac{-\mu^2 - 2L^2 + 2\sqrt{(L-2\mu)(L-\mu)(-\mu^2+L^2-\mu L)} + 4\mu L}{2L-3\mu} \\
\equiv& 0 > \mu^2 + \sqrt{(L-2\mu)(L-\mu)(-\mu^2+L^2-\mu L)} - L^2 + \mu L \\
\equiv& -\mu^2 + L^2 - \mu L > (L-2\mu)(L-\mu) \\
\equiv& \mu < \frac{2}{3}L,
\end{aligned}$$

which holds by hypothesis. Since $\gamma > \mu$ by hypothesis, inequality equation 48 holds for any μ and L positive as well, conditions satisfied by hypothesis.

Finally, since $\frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu} > \frac{\mu+\gamma+2\sqrt{\gamma L}}{L-\mu}$ for any $\gamma, \mu, L > 0$, then first inequality in equation 45 holds as well. This concludes the proof. □

We conclude this section by discussing several important insights:

- Except for α_{crit}, we do not report significant information coming from the analysis of $\lambda_i(C)$ for the computation of the step size and the validity of the candidates for α that are from $\left\{\frac{2\mu+2\sqrt{\mu L}}{L-\mu}, \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}\right\}$ respectively for the cases $\gamma = \mu$ and $\gamma > \mu$.
- Concerning the effect of the volatility σ of the noise, we have mentioned earlier that the parameter σ appears only within the numerators $\lambda_i(C)$ and based on intensive numerical tests, this parameter has a pure scaling effect onto the eigenvalues $\lambda_i(C)$ when studied w.r.t. α without modifying the trends of the curves. For compliance purpose, Figures 8 and 9 respectively show the evolution of the numerators $N_i(\alpha, \mu, L, \gamma, \sigma)$ of eigenvalues expressions of C given in Equations equation 38 to equation 41 w.r.t. σ, for both scenarios $\mu < L/2$ and $\mu > L/2$. One can observe monotonic polynomial increasing behavior of $N_i(\alpha, \mu, L, \gamma, \sigma)$ w.r.t σ for all $1 \leq i \leq 4$.
- The theoretical analysis summarized in this section is valid for the 2-dimensional case, we show in Appendix A.1.4 how to generalize our results for the n-dimensional case. This has no impact on our results.

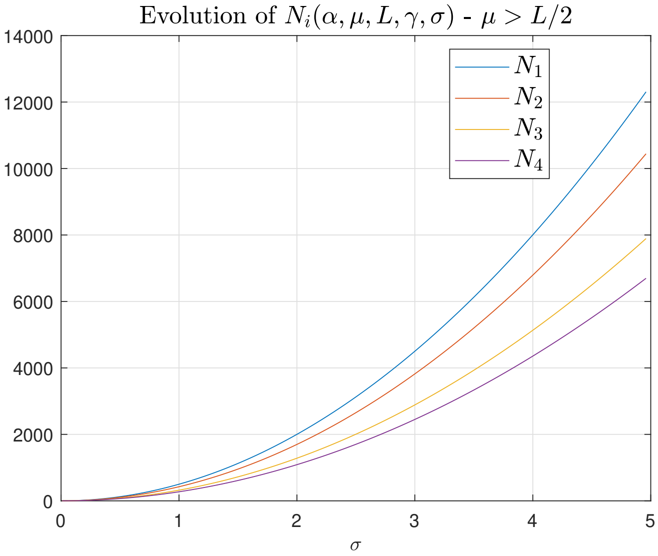


Figure 8: Evolution of $N_i(\alpha, \mu, L, \gamma, \sigma)$ w.r.t σ for scenario $\mu > L/2$; $\gamma = 3/2$, $\mu = 1$, $L = 3/2$, $\alpha = \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}$.

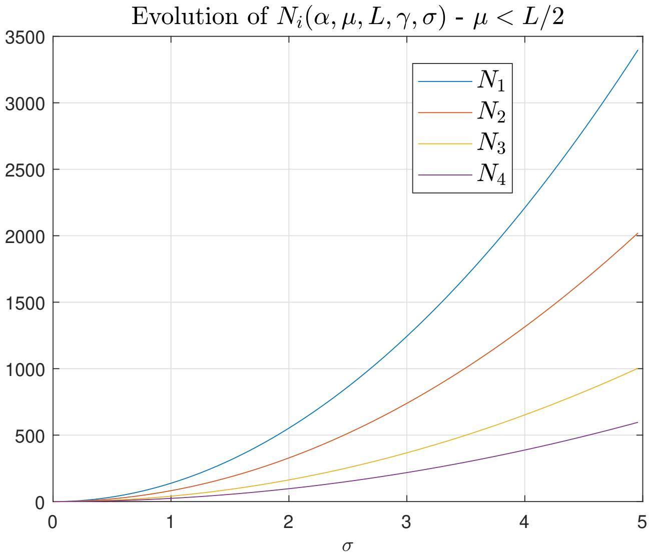


Figure 9: Evolution of $N_i(\alpha, \mu, L, \gamma, \sigma)$ w.r.t σ for scenario $\mu < L/2$; $\gamma = 3/2$, $\mu = 1$, $L = 3$, $\alpha = \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}$.

A.1.4 Extension to n-dimensional case

In this section, we show that we can easily extend the results obtained for the 2-dimensional case in Appendix A.1.1, Appendix A.1.2 and Appendix A.1.3 to the n-dimensional case with $n > 2$. Let us start by recalling that for NAG transformation (7), the general SDE's system to solve for the quadratic case is:

$$\dot{y}(t) = \begin{bmatrix} -I_{n\times n} & I_{n\times n} \\ 1/\gamma(\mu I_{n\times n} - A) & -\mu/\gamma I_{n\times n} \end{bmatrix} y(t) + \begin{bmatrix} 0_{n\times 1} \\ \frac{dZ}{dt} \end{bmatrix}, \quad t > 0. \tag{49}$$

Let recall that $y = (x, v)$ with $x, v \in \mathbb{R}^n$, let n be even and let consider the permutation matrix P associated to permutation indicator π given here-under in two-line form: $\pi = \left[\begin{array}{cccc|ccc} (1 \quad 2) & (3 \quad 4) & \cdots & (n-1 \quad n) & (n+1 \quad n+2) & \cdots & (2n-1 \quad 2n) \\ (2*1-1 \quad 2*1) & (2*3-1 \quad 2*3) & \cdots & (2n-3 \quad 2n-2) & (3 \quad 4) & \cdots & (2n-1 \quad 2n) \end{array} \right]$ where the bottom second-half part of π corresponds to the complementary of the bottom first half w.r.t. to the set $\{1, 2, ..., 2n\}$ in the increasing order. For avoiding ambiguities, the ones element of P are at indices $(\pi(1, j), \pi(2, j))$ for $1 \le j \le 2n$. For such convention and since permutation matrix P associated to indicator π is orthogonal matrix, equation 49 can be equivalently written as follows:

$$\begin{aligned} \dot{y}(t) &= PP^T \begin{bmatrix} -I_{n\times n} & I_{n\times n} \\ 1/\gamma(\mu I_{n\times n} - A) & -\mu/\gamma I_{n\times n} \end{bmatrix} PP^T y(t) + \begin{bmatrix} 0_{n\times 1} \\ \dot{Z} \end{bmatrix}, \\ \equiv P^T \dot{y}(t) &= P^T \begin{bmatrix} -I_{n\times n} & I_{n\times n} \\ 1/\gamma(\mu I_{n\times n} - A) & -\mu/\gamma I_{n\times n} \end{bmatrix} PP^T y(t) + P^T \begin{bmatrix} 0_{n\times 1} \\ \dot{Z} \end{bmatrix}, \end{aligned} \tag{50}$$

Since we assumed w.l.o.g. $A = \text{diag}(\lambda_1, \ldots, \lambda_n)$ with $\mu = \lambda_1 \leq \ldots \leq \lambda_n = L$, one can easily see that Equation equation 50 has the structure:

$$\left[\begin{array}{c} \dot{x}_1 \\ \dot{x}_2 \\ \dot{v}_1 \\ \dot{v}_2 \\ \hline \vdots \\ \hline \dot{x}_{2i-1} \\ \dot{x}_{2i} \\ \dot{v}_{2i-1} \\ \dot{v}_{2i} \\ \hline \vdots \\ \hline \dot{x}_{n-1} \\ \dot{x}_n \\ \dot{v}_{n-1} \\ \dot{v}_n \end{array}\right] = \left[\begin{array}{cc|c|cc|c|cc} I_2 & -I_2 & & & & & & \\ 1/\gamma(\mu I_2 - A_1) & -\mu/\gamma I_2 & 0 & \multicolumn{2}{c|}{0} & 0 & \multicolumn{2}{c}{0} \\ \hline \multicolumn{2}{c|}{0} & \ddots & \multicolumn{2}{c|}{0} & 0 & \multicolumn{2}{c}{0} \\ \hline \multicolumn{2}{c|}{0} & 0 & I_2 & -I_2 & 0 & \multicolumn{2}{c}{0} \\ & & & 1/\gamma(\mu I_2 - A_i) & -\mu/\gamma I_2 & & & \\ \hline \multicolumn{2}{c|}{0} & 0 & \multicolumn{2}{c|}{0} & \ddots & \multicolumn{2}{c}{0} \\ \hline \multicolumn{2}{c|}{0} & 0 & \multicolumn{2}{c|}{0} & 0 & I_2 & -I_2 \\ & & & & & & 1/\gamma(\mu I_2 - A_m) & -\mu/\gamma I_2 \end{array}\right] \cdot \left[\begin{array}{c} x_1 \\ x_2 \\ v_1 \\ v_2 \\ \hline \vdots \\ \hline x_{2i-1} \\ x_{2i} \\ v_{2i-1} \\ v_{2i} \\ \hline \vdots \\ \hline x_{n-1} \\ x_n \\ v_{n-1} \\ v_n \end{array}\right] + \left[\begin{array}{c} 0 \\ 0 \\ \dot{Z}_1 \\ \dot{Z}_2 \\ \hline \vdots \\ \hline 0 \\ 0 \\ \dot{Z}_{2i-1} \\ \dot{Z}_{2i} \\ \hline \vdots \\ \hline 0 \\ 0 \\ \dot{Z}_{n-1} \\ \dot{Z}_n \end{array}\right] \tag{51}$$

which boils down to $m = \frac{n}{2}$ independent 2-dimensional SDE's systems where $A_i = \text{diag}(\lambda_{2i-1}, \lambda_{2i})$ with $1 \leq i \leq m$ such that $\lambda_1 = \mu$ and $\lambda_n = L$.

Therefore, the m SDE's systems can be studied and theoretically solved independently with the schemes and the associated step sizes presented in previous sections. However, in practice, we will use a unique and general step size α to tackle the full SDE's system 49.

Let now use the "decoupled" structure given in equation 51 to come up with a general step size that will ensure the convergence of each system and hence the convergence of the full original system given in equation 49. Let us denote by α_i the step size for the i-th SDE's system with $1 \leq i \leq m = n/2$ minimizing the spectral radius of the system at hand. For convenience, let us consider the case $\gamma > \mu$, we apply the same method as detailed in Appendix A.1.1 and Appendix A.1.2 to compute the expression of α_i that minimizes $\rho(E_i(\alpha))$, we obtain:

$$\alpha_i = \frac{\mu + \gamma + \sqrt{(\mu - \gamma)^2 + 4\gamma\lambda_{2i}}}{\lambda_{2i} - \mu} \tag{52}$$

Finally, in Theorem 1, we show that choosing $\alpha_c := \alpha = \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}$ ensures the convergence of NAG-GS method used to solve the SDE's system 49 in the n-dimensional case for $n > 2$. Theorem 1 is enunciated in Section 2.3 in the main text and the proof is given here-under.

Proof. First, we recall that Lemma 3 in Section A.1.3 provides the proof for the asymptotic convergence of NAG-GS method for $n = 2$ when choosing $\alpha := \alpha_c = \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}$ for the case $\gamma > \mu$. In particular, it is shown that the spectral radius of the iteration matrix $\rho(E(\alpha_c))$ is strictly lower than 1 under consistent assumptions with the ones of Theorem 1 (see Lemma 3 for more details). The following steps of the proof show that choosing α_c also leads to the asymptotic convergence of NAG-GS method for $n > 2$.

To do so, let us start by considering, w.l.o.g., the SDE's system in the form given by equation 51 and let $\alpha_i = \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma\lambda_{2i}}}{\lambda_{2i}-\mu}$ be the step size (given in Equation (52)) selected for solving the i-th SDE's system with $1 \leq i \leq m = n/2$, minimizing $\rho(E_i(\alpha))$, that is the spectral radius of the associated iteration matrix E_i. The result of Lemma 3 can be directly extended for each independent 2-dimensional SDE's system, in particular showing that $\rho(E_i(\alpha_i)) < 1$ for $1 \leq i \leq m = n/2$.

Therefore, to prove the convergence of the NAG-GS method by choosing a single step size α such that $0 < \alpha \leq \frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}$, it suffices to show that:

$$\alpha = \frac{\mu + \gamma + \sqrt{(\mu - \gamma)^2 + 4\gamma L}}{L - \mu} \leq \min_{1 \leq i \leq m = n/2} \alpha_i \tag{53}$$

For proving that equation 53 holds, it sufficient to show that for any λ such that $0 < \mu \leq \lambda \leq L < \infty$ we have:

$$\frac{\mu + \gamma + \sqrt{(\mu - \gamma)^2 + 4\gamma L}}{L - \mu} \leq \frac{\mu + \gamma + \sqrt{(\mu - \gamma)^2 + 4\gamma\lambda}}{\lambda - \mu}. \tag{54}$$

which is equivalent to showing:

$$\begin{aligned}
&\frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}-\frac{\mu+\gamma+\sqrt{(\mu-\gamma)^2+4\gamma\lambda}}{\lambda-\mu}\leq 0\\
\equiv&\gamma(\frac{1}{L-\mu}-\frac{1}{\lambda-\mu})+\mu(\frac{1}{L-\mu}-\frac{1}{\lambda-\mu})+\frac{\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}-\frac{\sqrt{(\mu-\gamma)^2+4\gamma\lambda}}{\lambda-\mu}\leq 0
\end{aligned} \tag{55}$$

Since $0<\mu\leq\lambda\leq L<\infty$ by hypothesis, one can easily show that first two terms of the last inequality are negative. It remains to show that:

$$\begin{aligned}
&\frac{\sqrt{(\mu-\gamma)^2+4\gamma L}}{L-\mu}-\frac{\sqrt{(\mu-\gamma)^2+4\gamma\lambda}}{\lambda-\mu}\leq 0\\
\equiv&(-\gamma^2-4\gamma\lambda+2\gamma\mu-\mu^2)L^2+(4\gamma\lambda^2+2\gamma^2\mu+2\mu^3)L+\\
&\gamma^2\lambda^2-2\gamma^2\lambda\mu-2\gamma\lambda^2\mu+\lambda^2\mu^2-2\lambda\mu^3\leq 0
\end{aligned} \tag{56}$$

Note that we can easily show that the coefficient of L^2 is negative, hence last inequality is satisfied as soon as $L\leq\frac{-\gamma^2\lambda+2\gamma^2\mu+2\gamma\lambda\mu-\lambda\mu^2+2\mu^3}{\gamma^2+4\gamma\lambda-2\gamma\mu+\mu^2}$ or $L\geq\lambda$. The latter condition is satisfied by hypothesis, this concludes the proof.

Note that one can check that $\frac{-\gamma^2\lambda+2\gamma^2\mu+2\gamma\lambda\mu-\lambda\mu^2+2\mu^3}{\gamma^2+4\gamma\lambda-2\gamma\mu+\mu^2}\leq\lambda$. □

The theoretical results derived in these sections, along with the key insights, are validated in Appendix A.1.5 through numerical experiments conducted for the NAG-GS method in the quadratic case.

A.1.5 Numerical tests for quadratic case

In this section, we report some simple numerical tests for the NAG-GS method (Algorithm 1) used to tackle the accelerated SDE's system given in (11) where:

- the objective function is $f(x)=(x-ce)^TA(x-ce)$ with $A\in\mathbb{S}^3_+$, e a all-ones vector of dimension 3 and c a positive scalar. For such a strongly convex setting, since the feasible set is $V=\mathbb{R}^3$, the minimizer $\arg\min f$ uniquely exists and is simply equal to ce; it will be denoted further by $x^\star$. The matrix A is generated as follows: $A=QAQ^{-1}$ where matrix D is a diagonal matrix of size 3 and Q is a random orthogonal matrix. This test procedure allows us to specify the minimum and maximum eigenvalues of A that are respectively μ and L and hence it allows us to consider the two scenarios discussed in Appendix A.1.1, that are $\mu>L/2$ and $\mu<L/2$.
- The noise volatility σ is set to 1, we report that this corresponds to a significant level of noise.
- Initial parameter γ_0 is set to μ.
- Different values for the step size α will be considered in order to empirically demonstrate the optimal choice α_c in terms of contraction rate, but also validate the critical values for step size in the case $\mu<L/2$ and, finally, highlight the effect of the step size in terms of scattering of the final iterates generated by NAG-GS around the minimizer of f.

From a practical point of view, we consider $m=200000$ points. For each of them, the NAG-GS method is run for a maximum number of iterations to reach the stationarity, and the initial state x_0 is generated using normal Gaussian distribution. Since $f(x)$ is a quadratic function, the points are expected to converge to some Gaussian distribution around the minimizer $x^\star=ce$. Furthermore, since the initial distribution is also Gaussian, the intermediate distributions (at each iteration of the NAG-GS method) are expected to be Gaussian as well. Therefore, to quantify the rate of convergence of the NAG-GS method for different values of step size, we will monitor $\|\bar{x}^k-x^\star\|$, that is the distance between the empirical mean of the distribution at iteration k and the minimizer $x^\star$ of f.

Figures 10 and 11 respectively show the evolution of $\|\bar{x}^k-x^\star\|$ along iteration and the final distribution of points obtained by NAG-GS at stationarity for the scenario $\mu>L/2$, for the latter the points are projected onto the three planes to have a full visualization. As expected by the theory presented in Appendix A.1.3, there is no critical α, hence one may choose arbitrary large values for step size while the NAG-GS method still converges. Moreover, the choice of $\alpha = \alpha_c$ gives the highest rate of convergence. Finally, one can observe that the distribution of limit points tightens more and more around the minimizer $x^\star$ of f as the chosen step increases, as expected by the analysis of Figure 6. Hence, one may choose a very large step size α so that the limit points converge to $x^\star$ almost surely but at the cost of a (much) slower convergence rate. Here comes the tradeoff between the convergence rate and the limit points scattering.

Finally, Figures 12 and 13 provide similar results for the scenario $\mu < L/2$. The theory outlined in Appendix A.1.3 and Appendix A.1.4 predicts a critical value of α that indicates when the convergence of NAG-GS is destroyed in such a scenario. In order to illustrate this gradually, different values of α have been chosen within the set $\{\alpha_c, \alpha_c/2, (\alpha_c + \alpha_{\text{crit}})/2, 0.98\alpha_{\text{crit}}\}$. First, one can observe that the choice of $\alpha = \alpha_c$ gives again the highest rate of convergence; see Figure 12. Moreover, one can see that for $\alpha \to \alpha_{\text{crit}}$, the convergence starts to fail, and the spreading of the limit points tends to infinity. We report that for $\alpha = \alpha_{\text{crit}}$, NAG-GS method diverges. Again, the theory derived in previous sections fully predicted these numerical results.

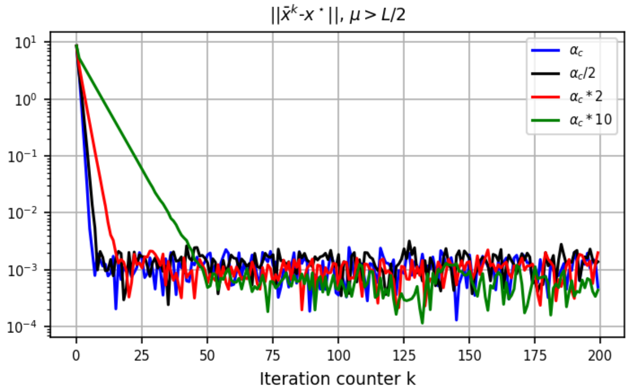


Figure 10: Evolution of $\|\bar{x}^k - x^\star\|$ along iteration for the scenario $\mu > L/2$; $c = 5$, $\gamma = \mu = 1$, $L = 1.9$ and $\sigma = 1$ for $\alpha \in \{\alpha_c, \alpha_c/2, 2\alpha_c, 10\alpha_c\}$ with $\alpha_c = \frac{2\mu+2\sqrt{\mu L}}{L-\mu} = 5.29$.

A.2 Fully-implicit scheme

In this section, we present an iterative method based on the NAG transformation G_{NAG} (7) along with a fully implicit discretization to tackle (4) in the stochastic setting, the resulting method shall be referred to as "NAG-FI" method. We propose the following discretization for (6) perturbated with noise; given step size $\alpha_k > 0$:

$$\begin{aligned} \frac{x_{k+1} - x_k}{\alpha_k} &= v_{k+1} - x_{k+1}, \\ \frac{v_{k+1} - v_k}{\alpha_k} &= \frac{\mu}{\gamma_k}(x_{k+1} - v_{k+1}) - \frac{1}{\gamma_k} A x_{k+1} + \sigma \frac{W_{k+1} - W_k}{\alpha_k}. \end{aligned} \tag{57}$$

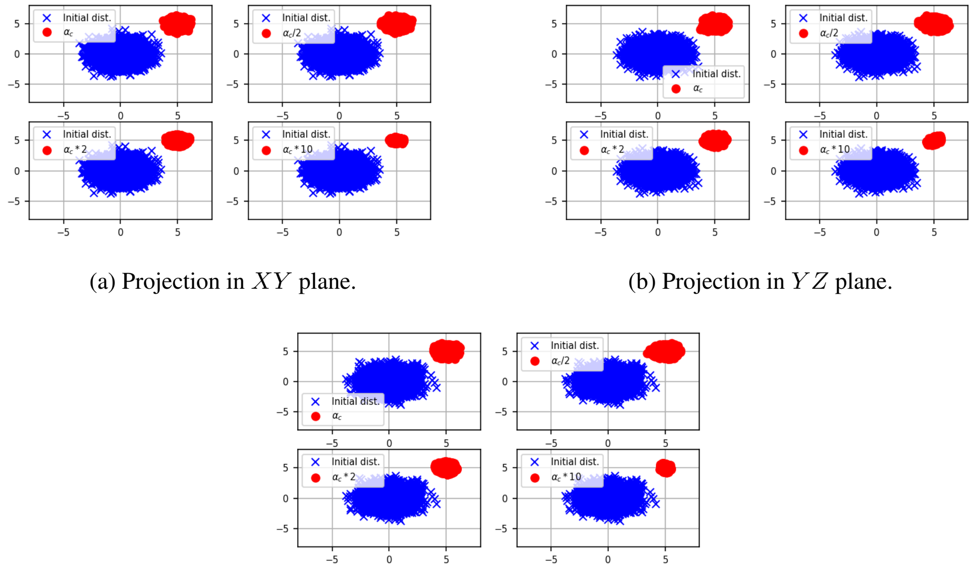
As done for the NAG-GS method, from a practical point of view, we will use $W_{k+1} - W_k = \Delta W_k = \sqrt{\alpha_k}\eta_k$ where $\eta_k \sim \mathcal{N}(0, 1)$, by the properties of the Brownian motion.

In the quadratic case, that is $f(x) = \frac{1}{2}x^\top Ax$, solving equation 57 is equivalent to solve:

$$\begin{bmatrix} x_k \\ v_k + \sigma\sqrt{\alpha_k}\eta_k \end{bmatrix} = \begin{bmatrix} (1+\alpha_k)I & -\alpha_k I \\ \frac{\alpha_k}{\gamma_k}(A - \mu I) & (1 + \frac{\alpha_k \mu}{\gamma_k})I \end{bmatrix} \begin{bmatrix} x_{k+1} \\ v_{k+1} \end{bmatrix} \tag{58}$$

where $\eta_k \sim \mathcal{N}(0, 1)$. Furthermore, ODE (8) from the main text is again discretized implicitly:

$$\frac{\gamma_{k+1} - \gamma_k}{\alpha_k} = \mu - \gamma_{k+1}, \quad \gamma_0 > 0. \tag{59}$$



(a) Projection in XY plane.

(b) Projection in YZ plane.

(c) Projection in XZ plane.

Figure 11: Initial (blue crosses) and final (red circles) distributions of points generated by the NAG-GS method for the scenario $\mu > L/2$; $c = 5$, $\gamma = \mu = 1$, $L = 1.9$ and $\sigma = 1$ for $\alpha \in \{\alpha_c, \alpha_c/2, 2\alpha_c, 10\alpha_c\}$ with $\alpha_c = \frac{2\mu+2\sqrt{\mu L}}{L-\mu} = 5.29$.

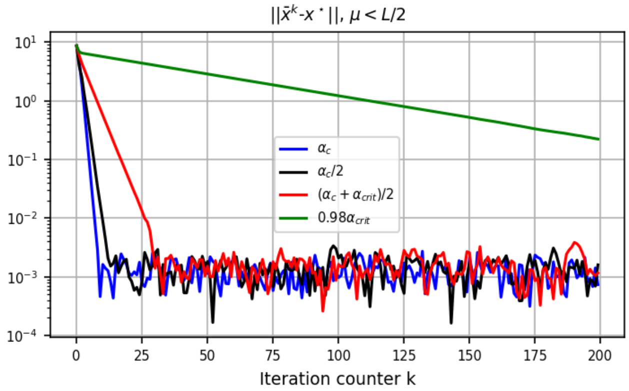
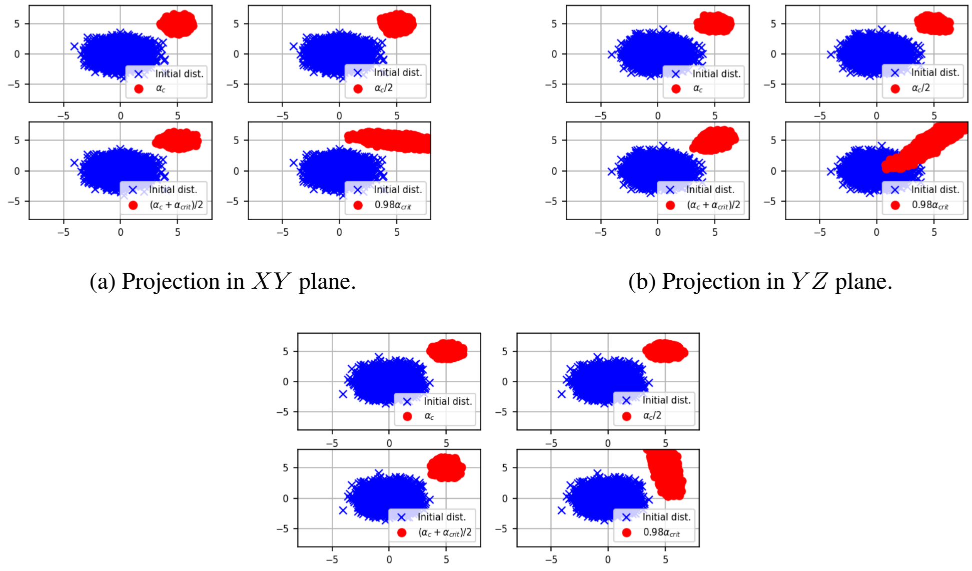


Figure 12: Evolution of $\|\bar{x}^k - x^\star\|$ along iteration for the scenario $\mu < L/2$; $c = 5$, $\gamma = \mu = 1$, $L = 3$ and $\sigma = 1$ for $\alpha \in \{\alpha_c, \alpha_c/2, (\alpha_c + \alpha_{\text{crit}})/2, 0.98\alpha_{\text{crit}}\}$ with $\alpha_c = \frac{2\mu+2\sqrt{\mu L}}{L-\mu} = 2.73$ and $\alpha_{\text{crit}} = \frac{\mu+\gamma+\sqrt{\gamma^2-6\gamma\mu+\mu^2+4\gamma L}}{L-2\mu} = 4.83$.

As done for NAG-GS method, heuristically, for general $f \in \mathcal{S}^{1,1}_{L,\mu}$ with $\mu \geq 0$, we just replace Ax_{k+1} in equation 57 with $\nabla f(x_{k+1})$ and obtain the following NAG-FI scheme:

$$\begin{aligned}
\frac{x_{k+1} - x_k}{\alpha_k} &= v_{k+1} - x_{k+1}, \\
\frac{v_{k+1} - v_k}{\alpha_k} &= \frac{\mu}{\gamma_k}(x_{k+1} - v_{k+1}) - \frac{1}{\gamma_k}\nabla f(x_{k+1}) + \sigma \frac{W_{k+1} - W_k}{\alpha_k}.
\end{aligned} \tag{60}$$



(a) Projection in XY plane.

(b) Projection in YZ plane.

(c) Projection in XZ plane.

Figure 13: Initial (blue crosses) and final (red circles) distributions of points generated by the NAG-GS method for scenario $\mu < L/2$; $c = 5$, $\gamma = \mu = 1$, $L = 3$ and $\sigma = 1$ for $\alpha \in \{\alpha_c, \alpha_c/2, (\alpha_c + \alpha_{\text{crit}})/2, 0.98\alpha_{\text{crit}}\}$ with $\alpha_c = \frac{2\mu+2\sqrt{\mu L}}{L-\mu} = 2.73$ and $\alpha_{\text{crit}} = \frac{\mu+\gamma+\sqrt{\gamma^2-6\gamma\mu+\mu^2+4\gamma L}}{L-2\mu} = 4.83$.

From the first equation, we get $v_{k+1} = \frac{x_{k+1}-x_k}{\alpha_k} + x_{k+1}$ that we substitute within the second equation, we obtain:

$$x_{k+1} = \frac{v_k + \tau_k x_k - \frac{\alpha_k}{\gamma_k}\nabla f(x_{k+1}) + \sigma\sqrt{\alpha_k}\eta_k}{1+\tau_k} \tag{61}$$

with $\tau_k = 1/\alpha_k + \mu/\gamma_k$.

Computing x_{k+1} is equivalent to computing a fixed point of the operator given by the right-hand side of equation 61. Hence, it is also equivalent to finding the root of the function:

$$g(u) = u - \left(\frac{v_k + \tau_k x_k - \frac{\alpha_k}{\gamma_k}\nabla f(u) + \sigma\sqrt{\alpha_k}\eta_k}{1+\tau_k}\right) \tag{62}$$

with $g : \mathbb{R}^n \to \mathbb{R}^n$. In order to compute the root of this function, we consider a classical Newton-Raphson procedure detailed in Algorithm 2. In Algorithm 2, $J_g(.)$ denotes the Jacobian operator of function g equation 62 w.r.t. u, I_n denotes the identity matrix of size n and $\nabla^2 f$ denotes the Hessian matrix of objective function f. Please note that the iterative method outlined in Algorithm 2 exhibits a connection to the family of second-order methods called the Levenberg-Marquardt algorithm Levenberg (1944); Marquardt (1963) applied to the unconstrained minimization problem $\min_{x\in\mathbb{R}^n} f(x)$ for a twice-differentiable function f. Finally, Algorithm 3 summarizes the NAG-FI method.

Algorithm 2 Newton-Raphson method

```
Input: Choose the point $u_0 \in \mathbb{R}^n$, some $\alpha_k, \gamma_k, \tau_k > 0$.
  for $i = 0, 1, \dots$ do
    Compute $J_g(u_i) = I_n + \frac{\alpha_k}{\gamma_k(1+\tau_k)}\nabla^2 f(u_i)$
    Compute $g(u_i)$ using equation 62
    Set $u_{i+1} = u_i - [J_g(u_i)]^{-1} g(u_i)$
  end for
```

Algorithm 3 NAG-FI Method

```
Input: Choose the point $x_0 \in \mathbb{R}^n$, set $v_0 = x_0$, some $\sigma \geq 0$, $\mu \geq 0, \gamma_0 > 0$.
  for $k = 0, 1, \dots$ do
    Sample $\eta_k \sim \mathcal{N}(0, 1)$
    Choose $\alpha_k > 0$
    Set $\gamma_{k+1} := \frac{\gamma_k + \alpha_k \mu}{1 + \alpha_k}$
    Set $\tau_{k+1} = 1/\alpha_k + \mu/\gamma_{k+1}$
    Compute the root $u$ of equation 62 by using Algorithm 2
    Set $x_{k+1} = u$
  end for
```

By following a similar stability analysis as the one performed for NAG-GS, one can show that this method is unconditionally A-stable as expected by the theory of implicit schemes. In particular, one can show that eigenvalues of the iterations matrix are positive decreasing functions w.r.t. step size α, allowing then the choice of any positive value for α. Similarly, one can show that the eigenvalues of the covariance matrix at stationarity associated with the NAG-FI method are decreasing functions w.r.t. α that tend to 0 as soon as $\alpha \to \infty$. It implies that Algorithm 3 is theoretically able to generate iterates that converge to $\arg\min f$ almost surely, even in the stochastic setting with the potentially quadratic rate of converge. This theoretical result is quickly highlighted in Figure 14 that shows the final distribution of points generated by NAG-FI once used in test setup detailed in Appendix A.1.5, in the most interesting and critical scenario $\mu < L/2$. As expected, α can be chosen as large as desired, we choose here $\alpha = 1000\alpha_c$. Moreover, for increasing α, the final distributions of points are more and more concentrated around x^*.

Therefore, the NAG-FI method constitutes a good basis for deriving efficient second-order methods for tackling stochastic optimization problems, which is hard to find in the current SOTA. Indeed, second-order methods and more generally some variants of preconditioned gradient methods have recently been proposed and used in the deep learning community for the training of NN for instance. However, it appears that there is limited empirical success for such methods when used for training NN when compared to well-tuned Stochastic Gradient Descent schemes, see for instance Botev et al. (2017); Zeiler (2012). To the best of our knowledge, no theoretical explanations have been brought to formally support these empirical observations. This will be part of our future research directions.

Besides these nice preliminary theoretical results and numerical observations for small dimension problems, there is a limitation of the NAG-FI method that comes from the numerical feasibility for computing the root of the non-linear function equation 62 that can be very challenging in practice. We will try to address this issue in future works.

A.3 Additional insights about the notion of A-stability

In this section, we recall the concept of A-stability of ODE solvers, which is the classical notion of "negative real part" by Dahlquist (1963). First, we note that the discussion about A-stability of solver for general ODE in the form $\dot{x}(t) = f\big(t, x(t)\big)$ with $x(0) = x_0, \Re(\lambda) < 0 \forall \lambda \in \sigma(J_f)$ can be long and tedious. Hence, we consider a simple linear ODE of the form

$$\dot{x}(t) \,=\, Gx(t), \quad x(0) \,=\, x_0 \quad \text{with} \quad \Re(\lambda) < 0 \;\; \forall \lambda \in \sigma(G). \tag{63}$$

A one-step method for solving ODE (63) with step size $\alpha > 0$ can be written as $x_{k+1} = E(G, \alpha)x_k$. The numerical scheme is called absolute stable or A-stable if $\rho(E(G, \alpha)) < 1$ (from which the asymptotic convergence $x_k \to 0$ follows). If $\rho(E(G, \alpha)) < 1$ holds for all $\alpha > 0$, then the scheme is called unconditionally A-stable, and if $\rho(E(G, \alpha)) < 1$ holds for some $\alpha \in I$, where I denotes an interval of the positive half line, then the scheme is conditionally A-stable. In the next subsection, we consider two popular schemes from the A-stability point of view.

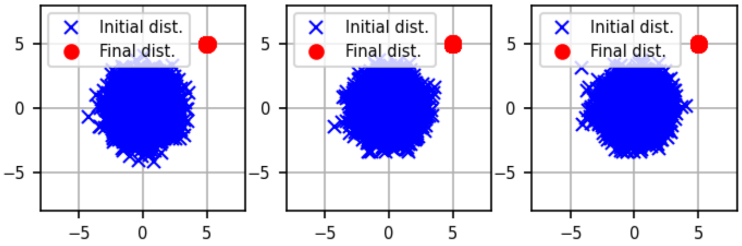


Figure 14: Projection onto XY, YZ, and XZ planes (from left to right) of initial (blue crosses) and final (red circles) distributions of points generated by NAG-FI method - scenario $\mu < L/2$; $c = 5$, $\gamma = \mu = 1$, $L = 3$ and $\sigma = 1$ for $\alpha = 1000\alpha_c$ with $\alpha_c = \frac{2\mu+2\sqrt{\mu L}}{L-\mu} = 2.73$.

A.3.1 Explicit and implicit Euler schemes

Here, we review the stability of the explicit and implicit Euler schemes for solving (63). The analytical solution for (63) with a constant discretization step α generates the iterates:

$$x_{k+1} = x_k + \int_{t_k}^{t_{k+1}} Gx(s)ds, \quad k = 0, \ldots, M-1.$$

For $G = -A$, the explicit Euler method approximates the integral by the area of a rectangle with width α and height $-Ax_k$. This leads to the iterates $x_{k+1} = (I - \alpha A)x_k$ which corresponds to the GD scheme for minimizing $\Phi(x) = \frac{1}{2}x^\top Ax$. The explicit Euler method is A-stable if the spectral radius of $I - \alpha A$ is strictly less than 1, i.e., if $\rho(I - \alpha A) = \max_{\lambda \in \sigma(I-\alpha A)} |\lambda| < 1$. We can easily show that $\rho(I - \alpha A) = \max(|1 - \alpha\mu|, |1 - \alpha L|)$, where μ and L respectively denote the smallest and largest eigenvalue of A. Therefore, the explicit Euler method is A-stable if $0 < \alpha < 2/L$. Additionally, we can determine the optimal α that minimizes the spectral radius: $\min_{\alpha>0} \rho(I - \alpha A)$, which gives $\alpha^\star = 2/(\mu + L)$, resulting in $\rho(I - \alpha^\star A) = (Q_f - 1)/(Q_f + 1)$. Assuming $0 < \mu \le L < \infty$, we have $0 < \alpha^\star = 2/(\mu + L) < 2/L$. Hence, the explicit Euler method is A-stable, and the norm convergence with a linear rate follows as in (64).

$$\|x_k - x^\star\|_2 \le \left(\frac{Q_f - 1}{Q_f + 1}\right)^k \|x_0 - x^\star\|_2 \quad \text{and} \quad \Phi(x_k) - \Phi^\star \le \frac{L}{2}\left(\frac{Q_f - 1}{Q_f + 1}\right)^{2k} \|x_0 - x^\star\|_2^2. \tag{64}$$

On the other hand, the implicit Euler scheme approximates the integral by the area of a rectangle with a height of $-Ax_{k+1}$, leading to the iterates $x_{k+1} = (I + \alpha A)^{-1}x_k$. The term $\rho(I + \alpha A)^{-1}$ can be expressed as $\max\left(|\frac{1}{1+\alpha\mu}|, |\frac{1}{1+\alpha L}|\right)$. This implies that the stability condition $\rho(I + \alpha A)^{-1} < 1$ holds true for all $\alpha > 0$, making the implicit Euler scheme unconditionally A-stable. Moreover, the implicit Euler method can achieve a faster convergence rate by time rescaling, as any constraints on the step size do not limit it. This is equivalent to opting for a larger step size.

B Convergence to the stationary distribution

Another way to study the convergence of the proposed algorithms is to consider the Fokker-Planck equation for the density function $\rho(t, x)$. We will consider the simple case of the scalar SDE for the stochastic gradient flow (similarly as in (11)). Here $f : \mathbb{R} \to \mathbb{R}$:

$$dx = -\nabla f(x)dt + dZ = -\nabla f(x)dt + \sigma dW, \quad x(0) \sim \rho(0, x).$$

It is well known, that the density function for $x(t) \sim \rho(t, x)$ satisfies the corresponding Fokker-Planck equation:

$$\frac{\partial \rho(t, x)}{\partial t} = \nabla \left(\rho(t, x) \nabla f(x) \right) + \frac{\sigma^2}{2} \Delta \rho(t, x) \tag{65}$$

For the equation 65 one could write down the stationary (with $t \to \infty$) distribution

$$\rho^*(x) = \lim_{t \to \infty} \rho(t, x) = \frac{1}{Z} \exp\left(-\frac{2}{\sigma^2} f(x) \right), \quad Z = \int\limits_{x \in V} \exp\left(-\frac{2}{\sigma^2} f(x) \right) dx. \tag{66}$$

It is useful to compare different optimization algorithms in terms of convergence in the probability space because it allows us to study the methods in the non-convex setting. We have to address two problems with this approach. Firstly, we need to specify some distance functional between current distribution $\rho_t = \rho(t, x)$ and stationary distribution $\rho^* = \rho^*(x)$. Secondly, we do not need to have access to the densities ρ_t, ρ^* themselves.

For the first problem, we will consider the following distance functionals between probability distributions in the scalar case:

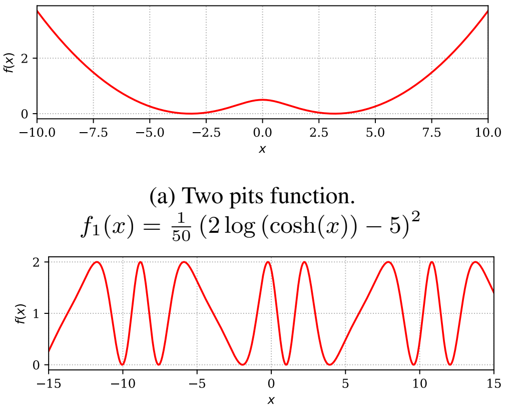
- **Kullback-Leibler divergence.** Several studies dedicated to convergence in probability space are available Arnold et al. (2001); Chewi et al. (2020); Lambert et al. (2022). We used the approach proposed in Pérez-Cruz (2008) to estimate KL divergence between continuous distributions based on their samples.

- **Wasserstein distance.** Wasserstein distance is relatively easy to compute for scalar densities. Also, it was shown, that the stochastic gradient process with a constant learning rate is exponentially ergodic in the Wasserstein sense Latz (2021).

- **Kolmogorov-Smirnov statistics.** We used the two-sample Kolmogorov-Smirnov test for goodness of fit.

To the best of our knowledge, the explicit formula for the stationary distribution of Fokker-Planck equations for the ASG SDE (11) remains unknown. That is why we have decided to get samples from the empirical stationary distributions using Euler-Maruyama integration Maruyama (1955) with a small enough step size of corresponding SDE with a bunch of different independent initializations.

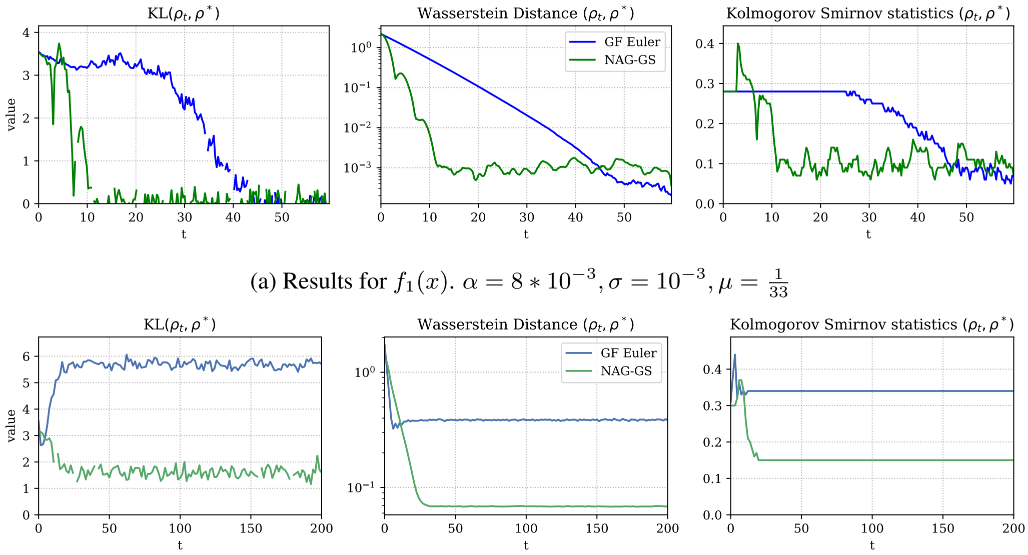
We tested two functions, which are presented in Figure 15. We initially generated 100 points uniformly in the function domain. Then we independently solved the initial value problem (9) for each of them with Maruyama (1955). Results of the integration are presented in Figure 16. One can see, that in the relatively easy case (Figure 15a), NAG-GS converges faster, than gradient flow to its stationary distribution, see Figure 16a. At the same time, in the hard case (Figure 15b), NAG-GS is more robust to the large step size, see Figure 16b.



(a) Two pits function.
$f_1(x) = \frac{1}{50}\left(2\log\left(\cosh(x)\right) - 5\right)^2$

(b) Frequently modulated sin function.
$f_2(x) = \cos\left(1.6x + \frac{5}{3}\sin(0.64x) - \pi\right)$

Figure 15: Non convex scalar functions to test



(a) Results for $f_1(x)$. $\alpha = 8 * 10^{-3}, \sigma = 10^{-3}, \mu = \frac{1}{33}$

(b) Results for $f_2(x)$. $\alpha = 1.5, \sigma = 10^{-2}, \mu = 1$

Figure 16: Convergence in probabilities of Euler integration of Gradient Flow (GF Euler) and NAG-GS for the non-convex scalar problems.

C ADDITIONAL INSIGHTS FROM EXPERIMENTAL EVALUATION

In this section, we provide additional experimental details. In particular, we discuss our experimental setup a little more and give some insights about NAG-GS. Our computational resources are limited to a single Nvidia DGX-1 with 8 GPUs V100. Almost all experiments were carried out on a single GPU. The only exception is for the training of ResNet50 on ImageNet, which used 8 GPUs.

C.1 PHASE DIAGRAMS

In Appendix C.4 we mentioned that the lowest eigenvalues μ of approximated Hessian matrices evaluated during the training of the ResNet-20 model were negative. Furthermore, our theoretical analysis of NAG-GS in the convex case includes some conditions on the optimizer parameters α, γ, and μ. In particular, it is required that $\mu > 0$ and $\gamma \geq \mu$. In order to bring some insights about these remarks in the non-convex setting and inspired by Velikanov et al. (2022), we experimentally study the convergence regions of NAG-GS and sketch out the phase diagrams of convergence for different projection planes, see Figure 17.

We consider the same setup as in Section 3.4 in the main text, a paragraph about the ResNet-20 model, and use hyper-optimization library OPTUNA Akiba et al. (2019). Our preliminary experiments on RoBERTa show that α should be of magnitude 10^{-1}. With the estimate of the Hessian spectrum of ResNet-20, we define the following search space

$$\alpha \sim \text{LogUniform}(10^{-2}, 10^2), \quad \gamma \sim \text{LogUniform}(10^{-2}, 10^2), \quad \mu \sim \text{Uniform}(-10, 100).$$

We sample a fixed number of triples and train the ResNet-20 model on CIFAR-10. The objective function is a top-1 classification error.

We report that there is a convergence almost everywhere within the projected search space onto α-γ plane (see Figure 17). The analysis of projections onto α-μ and γ-μ planes brings different conclusions: there are regions of convergence for negative μ for some $\alpha < \alpha_{th}$ and $\gamma > \gamma_{th}$. Also, there is a subdomain of negative μ comparable to a domain of positive μ in the sense of the target metrics. Moreover, the majority of sampled points are located in the vicinity of the band $\lambda_{\min} < \mu < \lambda_{\max}$.

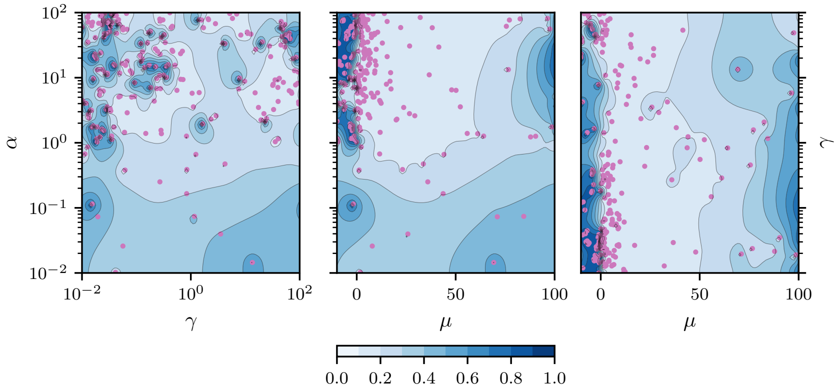


Figure 17: Landscapes of classification error for ResNet-20 model trained on CIFAR-10 with NAG-GS after projections onto $\alpha - \gamma$, $\alpha - \mu$ and $\gamma - \mu$ planes (from left to right). Hyperparameter optimization algorithm samples learning rate α from $[10^{-2}, 10^{2}]$, factor γ from $[10^{-2}, 10^{2}]$, and factor μ from $[-10, 90]$. Hyperparameters α and γ are sampled from log-uniform distribution, and hyperparameter μ is sampled from a uniform distribution.

Table 5: The comparison of a single step duration for different optimizers on RESNET-20 on CIFAR-10. ADAM-like optimizers have in twice larger state than SGD with momentum or NAG-GS.

OPTIMIZER	MEAN, S	VARIANCE, S	REL. MEAN	REL. VARIANCE
SGD	0.458	0.008	1.0	1.0
NAG-GS	1.648	0.045	3.6	5.5
SGD-M	3.374	0.042	7.4	5.2
SGD-MW	3.512	0.037	17.7	4.7
ADAMW	5.208	0.102	11.4	12.6
ADAM	7.919	0.169	17.3	20.8

C.2 IMPLEMENTATION DETAILS

In our work, we implemented NAG-GS in PyTorch Paszke et al. (2017) and JAX Bradbury et al. (2018); Babuschkin et al. (2020). Both implementations are used in our experiments and available online[3]. According to Algorithm 1, the size of the NAG-GS state equals to number of optimization parameters which makes NAG-GS comparable to SGD with momentum. It is worth noting that Adam-like optimizers have a twice larger state than NAG-GS. The arithmetic complexity of NAG-GS is linear $O(n)$ in the number of parameters. Table 5 shows a comparison of the computational efficiency of common optimizers used in practice. Although forward pass and gradient computations usually give the main contribution to the training step, there is a setting where the efficiency of gradient updates is important (e.g. batch size or a number of intermediate activations are small with respect to a number of parameters).

C.3 UPDATABLE SCALING FACTOR γ

According to the theory of NAG-GS optimizer presented in Section 2, the scaling factor γ decays exponentially fast to μ and, in the case $\gamma_0 = \mu$, γ remains constant along iterations. So, a natural question arises: is the update on γ necessary? Our experiments confirm that scaling factor γ should be updated accordingly to Algorithm 1, even in this highly non-convex setting, to get better metrics on test sets.

We use an experimental setup for ResNet-20 from Section 3.4 in the main text and search for hyperparameters for NAG-GS with updatable γ and with constant one. Common hyper-optimization

[3] https://github.com/skolai/nag-gs

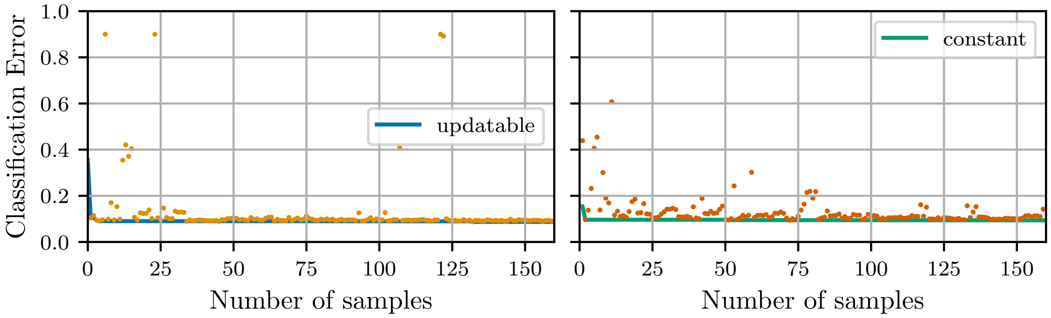


Figure 18: The best acc@1 on test set for updatable and fixed scaling factor γ during hyperoptimization. NAG-GS with updatable γ gives more frequently better results than the ones obtained with constant γ.

library OPTUNA Akiba et al. (2019) is used with a budget of 160 iterations to sample NAG-GS parameters. Figure 18 plots the evolution of the best score value along optimization time.

C.4 NON-CONVEXITY AND HESSIAN SPECTRUM

Theoretical analysis of NAG-GS highlights the importance of the smallest eigenvalue of the Hessian matrix for convex and strongly convex functions. Unfortunately, the objective functions usually considered for the training of neural networks are not convex. In this section, we try to address this issue. ResNet-20 is the smallest model in our experimental setup. However, we cannot afford to compute exactly the Hessian matrix since ResNet-20 has almost 300k parameters. Instead, we use Hessian-vector product (HVP) $H(x)$ and apply matrix-free algorithms for finding the extreme eigenvalues. We estimate the extreme eigenvalues of the Hessian spectrum with power iterations (PI) along with Rayleigh quotient (RQ) (Golub & van Loan, 2013). PI is used to get a good initial vector which is used later in the optimization of RQ. In order to get a more useful initial vector for the estimation of the smallest eigenvalue, we apply the spectral shift $H(x) - \lambda_{\max} x$ and use the corresponding eigenvector.

Figure 19 shows the extreme eigenvalues of ResNet-20 Hessian at the end of each epoch for the batch size 256 in the same setup as in Section 3.4 in the main text. The largest eigenvalue is strictly positive, while the smallest one is negative and usually oscillates around -1. It turns out that there is an island of hyperparameters near that μ. We report that training ResNet-20 with hyperparameters from this island gives good target metrics. The negative momenta domain is non-conventional and poorly understood, to the best of our knowledge. Moreover, NAG-GS has no theoretical guarantees in the non-convex case and negative μ. However, Velikanov et al. (2022) reports the existence of regions of convergence for SGD with negative momentum, which supports our observations. The theoretical aspects of these observations will be studied in future work.

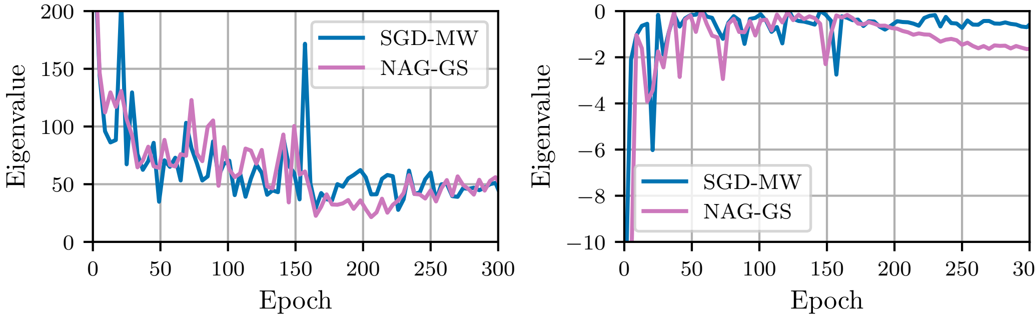


Figure 19: Evolution of the extreme eigenvalues (the largest and the smallest ones) during training RESNET-20 on CIFAR-10 with the NAG-GS optimizer.