On the Global Convergence of the Regularized Generalized Gauss-Newton Algorithm

Vincent Roulet VROULET@UW.EDU
Department of Statistics, University of Washington, Seattle, WA, USA
Maryam Fazel MFAZEL@ECE.UW.EDU
Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, USA
Siddhartha Srinivasa SIDDH@CS.UW.EDU
Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA, USA
Zaid Harchaoui ZAID@UW.EDU
Department of Statistics, University of Washington, Seattle, WA, USA

Abstract

We detail the global convergence rates of a regularized generalized Gauss-Newton algorithm applied to compositional problems with surjective inner Jacobian mappings. Our analysis uncovers several convergence phases and identifies key condition numbers governing the complexity of the algorithm. We present an implementation with a line-search adaptive to unknown constants.

Keywords: Generalized Gauss-Newton, Global Convergence

1. Introduction

We consider compositional optimization problems of the form

$$\min_{w \in \mathbb{R}^p} \left\{ f(w) := h(g(w)) \right\}, \tag{1}$$

with $h : \mathbb{R}^p \to \mathbb{R}^q$ strongly convex, $g : \mathbb{R}^q \to \mathbb{R}$ nonlinear and $\arg\min_{w \in \mathbb{R}^p} f(w) \neq \emptyset$. Such problems arise in numerous applications such as in nonlinear control [2, 14, 25], in deep learning applications, see, e.g., [28] and references therein, or in any nonlinear least-squares problems, such as phase retrieval [12], see [3] for an overview. For example, a nonlinear control problem can take the form

$$\min_{\substack{u_0, \ldots, u_{\tau-1} \in \mathbb{R}^{d_u} \\ x_0, \ldots, x_\tau \in \mathbb{R}^{d_x}}} \textstyle\sum_{t=1}^{\tau} h_t(x_t) \qquad \text{subject to} \quad x_{t+1} = \phi_t(x_t, u_t), \ \text{for } t \in \{0, \ldots, \tau-1\}, \quad x_0 = \bar{x}_0,$$

where, at time t, x_t is the state of the system, u_t is the control applied to the system, ϕ_t is the discrete time dynamic, h_t is the cost on the state and $\bar{x}_0$ is a fixed initial point. The problem is entirely characterized by the choice of the control variables and can then be reformulated as

$$\min_{w = (u_0; \ldots; u_{\tau-1}) \in \mathbb{R}^{\tau d_u}} h(g(w)) \quad \begin{array}{c} \text{with } g(w) = (x_1; \ldots; x_\tau) \\ \text{s.t. } x_{t+1} = \phi_t(x_t, u_t) \end{array} \qquad h(x) = \textstyle\sum_{t=1}^{\tau} h_t(x_t),$$

where h encapsulates the total cost on the trajectory $x = (x_1; \ldots; x_\tau)$ and g is the function that, at a given set of control variables $w = (u_0; \ldots; u_{\tau-1})$, associates the corresponding trajectory.

Similarly, a feed-forward deep network of τ layers applied to an input x such as an image reads

$$\psi(x, w) = z_\tau \quad \text{s.t.} \quad z_t = \phi_t(z_{t-1}, w_t) \text{ for } t \in \{1, \ldots, \tau\},\ w = (w_1; \ldots; w_\tau) \quad z_0 = x,$$

where w_t are the parameters of the t$^{\text{th}}$ layer ϕ_t that consist, for example, for a multi-layer perception, a nonlinear function composed with an affine function. Given a dataset of input-output pairs $(x_i, y_i)_{i=1}^n$ and a loss ℓ measuring the error of predicting $\hat{y}$ instead of y as $\ell(\hat{y}, y)$, training a deep network consists in minimizing the loss incurred by predicting the outputs through the deep network, i.e., solving $\min_{w\in\mathbb{R}^p} \sum_{i=1}^n \ell(\phi(x_i, w), y_i)$, which can be rewritten as

$$\min_{w\in\mathbb{R}^p} h(g(w)) \quad \text{with } g(w) = (\phi(x_1, w); \ldots; \phi(x_n, w)), \quad h(z) = \textstyle\sum_{i=1}^n \ell(z_i, y_i).$$

Provided that g, h are differentiable, the objective in (1) can be tackled by standard first order methods such as a gradient descent [18, 28]. Here we rather consider taking advantage of the compositional structure of the problem to apply a Gauss-Newton type algorithm provided that the outer function h is also twice differentiable [19]. Namely, we consider updating iterates by minimizing a quadratic approximation of the outer function h on top of a linear approximation of g around the current iterate with an additional regularization, that is, we consider Regularized Generalized Gauss-Newton (RGGN) iterates of the form

$$\begin{aligned} w_{k+1} &= \operatorname*{arg\,min}_{w\in\mathbb{R}^p} \left\{ q_h^{g(w_k)}(\ell_g^{w_k}(w)) + \frac{\nu_k}{2}\|w - w_k\|_2^2 \right\} \\ &= w_k - (\nabla g(w_k)\nabla^2 h(g(w_k))\nabla g(w_k)^\top + \nu_k \mathrm{I})^{-1}\nabla g(w_k)\nabla h(g(w_k)), \end{aligned} \tag{2}$$

for $\nu_k \geq 0$ a regularization that may depend on the current iterate, where for a function f we denoted by $\ell_f^x(y) = f(x) + \nabla f(x)^\top(y - x)$ and $q_f^x(y) = f(x) + \nabla f(x)^\top(y - x) + (y - x)^\top\nabla^2 f(x)(y - x)/2$ the linear and quadratic approximations of f around x. Updates of the form (2) generalize the classical Gauss-Newton algorithm [6, 17, 19] by considering a generic outer function h rather than the usual Euclidean squared norm [9] and by adding a regularization scheme in the spirit of Levenberg-Marquardt methods [16].

We consider deriving global convergence rates under the following additional assumption

$$\exists \sigma_g > 0 \text{ s.t. } \forall w \in \mathbb{R}^p,\ \sigma_{\min}(\nabla g(w)) := \inf_{\lambda\in\mathbb{R}^q} \frac{\|\nabla g(w)\lambda\|_2}{\|\lambda\|_2} \geq \sigma_g > 0, \tag{3}$$

where $\nabla g(w) = (\partial_{w_i} g_j(w))_{1\leq i\leq p, 1\leq j\leq q} \in \mathbb{R}^{p\times q}$ denotes the gradient of g, i.e., the transpose of the Jacobian of g. Assumption (3) ensures that the Jacobian mapping $v \to \nabla g(w)^\top v$ of the inner function g is surjective such that any problem of the form $\nabla g(w)^\top v = z$ for $w \in \mathbb{R}^p$ and $z \in \mathbb{R}^q$ can be solved in terms of $v \in \mathbb{R}^p$. Assumption (1) has been previously studied for degenerate nonlinear systems of equations [1, 17] or nonlinear control problems with some adequate controllability conditions [23].

In this work, we show that given condition (3) and suitable smoothness assumptions, by taking regularization terms proportional to the norm of the outer function at the current iterate, an RGGN algorithm converges globally with a local quadratic rate. In Sec. 2, we first analyze the RGGN algorithm for an ideal choice of regularization given the knowledge of all constants governing the problem and then propose an implementation with a line-search on the regularization parameter. Comparisons with previous work are presented in Sec. 3 after having detailed our results. Additional global convergence results for convex outer functions satisfying a generic Polyak-Łojasiewicz inequality and local convergence proofs for self-concordant outer functions are presented in Theorem 2 and Theorem 5 respectively in Appendix B.

2. Convergence Analysis

RGGN algorithm with ideal regularization choice. Our main theorem is presented in Theorem 1 with detailed assumptions and constants. We first present the rationale behind our results.

First, note that since g is not linear, the objective $f = h \circ g$ is a priori not convex even if h is convex. Yet, global convergence of, e.g., first order methods can be ensured if the objective satisfies a gradient dominating property, i.e., if there exists $m > 0$ such that $\|\nabla f(w)\|_2^2 \geq m(f(w) - \min_{v \in \mathbb{R}^p} f(v))$ [4, 15, 20]. By considering a μ_h-strongly convex outer function and Assumption (3), we have that the objective satisfies such gradient dominating property as we have for any $w \in \mathbb{R}^p$, denoting $x = g(w)$,

$$\|\nabla f(w)\|_2^2 = \|\nabla g(w) \nabla h(x)\|_2^2 \geq \sigma_g^2 \|\nabla h(x)\|_2^2 \geq \sigma_g^2 \mu_h \left(h(x) - \min_{y \in \mathbb{R}^q} h(y) \right). \tag{4}$$

Since the set $\{w : \nabla f(w) = 0\}$ is not empty as we assumed that the problem had a minimizer, we conclude that $\min_{v \in \mathbb{R}^p} f(v) = \min_{y \in \mathbb{R}^q} h(y)$ and the objective satisfies a gradient dominating property.

By choosing a large enough regularization, the updates of the RGGN algorithm approach the ones of a gradient descent as we have for $\nu_k \gg 1$, $w_{k+1} \approx w_k - \nu_k^{-1} \nabla f(w_k)$, which suggests that the RGGN algorithm can converge globally at a linear rate just as a gradient descent given (4) [4, 20]. Formally, we consider taking a regularization ν_k that may depend on the current $w_k \in \mathbb{R}^p$, s.t.

$$\begin{aligned} f(w_{k+1}) &\leq \min_{w \in \mathbb{R}^p} \left\{ q_h^{g(w_k)}(\ell_g^{w_k}(w)) + \frac{\nu_k}{2} \|w - w_k\|_2^2 \right\} \\ &= f(w_k) - \frac{1}{2} \nabla f(w_k)^\top (\nabla g(w_k) \nabla^2 h(g(w_k)) \nabla g(w_k)^\top + \nu_k \mathrm{I})^{-1} \nabla f(w_k). \end{aligned} \tag{5}$$

The above condition ensures that $f(w_{k+1}) - f(w_k) \leq -\alpha_k \|\nabla f(w_k)\|_2^2$, where α_k depends on the regularization ν_k and the properties of the objective. Hence, using (4), by taking a constant regularization ensuring (5), we get a global linear convergence rate.

The global rate of convergence sketched above can be further improved by analyzing the local behavior of the algorithm around a solution. Namely, if g satisfies (3), then the matrix $\nabla g(w)^\top \nabla g(w)$ is invertible for any $w \in \mathbb{R}^p$. Denoting $x_k = g(w_k)$, $G = \nabla g(w_k)$ and $H = \nabla^2 h(x_k)$, we then have by standard linear algebra, that $w_{k+1} - w_k = -G(G^\top G)^{-1}(H + \nu_k (G^\top G)^{-1})^{-1} \nabla h(x_k)$. Consider then the variables $x_k = g(w_k)$ associated to a single step of the RGGN algorithm,

$$\begin{aligned} x_{k+1} = g(w_{k+1}) &\approx g(w_k) + \nabla g(w_k)^\top (w_{k+1} - w_k) \\ &= x_k - (\nabla^2 h(x_k) + \nu_k (\nabla g(w_k)^\top \nabla g(w_k))^{-1})^{-1} \nabla h(x_k). \end{aligned}$$

For $\nu \ll 1$, the difference $x_{k+1} - x_k$ is then close to a Newton direction on the outer function h. The RGGN algorithm can then be analyzed as an approximate Newton method on the outer function, which suggests that it can have a local quadratic convergence rate if ν_k decreases fast enough.

To blend global convergence and local quadratic convergence, we observe that to satisfy condition (5), the regularization can be chosen to be proportional to the norm of the gradient of the outer function at the current iterate, i.e., $\nu_k = \bar{\nu}_k \|\nabla h(g(w_k))\|_2$ for $\bar{\nu}_k$ bounded above by a constant which ensures that ν_k tends to 0 with the iterations k. Theorem 1 presents then the total complexity of an RGGN algorithm for strongly convex outer functions h with an ideal choice of regularization.

Theorem 1 *Consider problems of the form* (1) *for an outer function h μ_h-strongly convex with L_h-Lipschitz continuous gradients, M_h-Lipschitz-continuous Hessians and an inner function g ℓ_g-Lipschitz-continuous with L_g-Lipschitz-continuous gradients satisfying* (3)*.*

The number of iterations of an RGGN algorithm (2)*, with regularizations*

$$\nu_k = \left(1 + \frac{\alpha}{2(1+\theta_g\|\nabla h(g(w_k))\|_2/(\sqrt{\mu_h}\rho_g))}\right) L_g\|\nabla h(g(w_k))\|_2, \tag{6}$$

needed to reach an accuracy ε is at most

$$k(\delta_0, \varepsilon) := 4\theta_g\left(\sqrt{\delta_0} - \sqrt{\varepsilon}\right) + 2\rho_h \ln\left(\frac{\delta_0}{\varepsilon}\right) + 2\alpha \ln\left(\frac{\theta_g\sqrt{\delta_0} + \rho_g}{\theta_g\sqrt{\varepsilon} + \rho_g}\right), \tag{7}$$

where $\rho_h = L_h/\mu_h$, $\rho_g = \ell_g/\sigma_g$, $\theta_g = L_g/(\sigma_g^2\sqrt{\mu_h})$, $\theta_h = M_h/(2\mu_h^{3/2})$, $\alpha = 4\rho_g^2\rho_h(\beta+1)$, $\beta = M_h\ell_g^2/(3L_gL_h)$ and $\delta_0 = f(w^{(0)}) - \min_{w\in\mathbb{R}^p} f(w)$.

If the desired target accuracy ε is smaller than a gap $\mathfrak{d} = 1/(32\rho_h(\theta_h(1+\sqrt{\rho_h}\rho_g^3/3) + \sqrt{\rho_h}\theta_g(1+\rho_g\rho_h))^2)$ which determines a quadratic convergence phase, the number of iterations of an RGGN algorithm, with regularization ν_k defined above, needed to reach the accuracy ε is at most $k(\delta_0, \mathfrak{d}) + \ln\ln(\varepsilon^{-1})$.

The constants appearing in the complexity bound are (i) the condition number $\rho_h = L_h/\mu_h$ of the outer function h, (ii) the condition number $\rho_g = \ell_g/\sigma_g$ of the gradient of g, (iii) a constant $\theta_h = M_h/(2\mu_h^{3/2})$ that is a bound on the self-concordance parameter of the outer function h [18, Section 5], (iv) a constant $\theta_g = L_g/(\sigma_g^2\sqrt{\mu_h})$ whose dimension is the same as θ_h, i.e., the inverse of the squared root of the objective. Finally the terms α and β are additional dimension independent constants that act as additional condition numbers.

After k iterations of the RGGN algorithm with regularizations defined as in (6), the number of remaining iterations to reach an accuracy $\varepsilon \geq \mathfrak{d}$ is bounded as $k(\delta_k, \varepsilon)$, that is dominated by the term $4\theta_g\sqrt{\delta_k}$ as long as $\delta_k\theta_g^2 \geq C(\delta_k, \varepsilon)^2(\alpha+\rho_h)^2$, where $C(\delta_k, \varepsilon)$ entails logarithmic terms in δ_k, ε. Hence, $R = 1/(\theta_g(\alpha+\rho_h))^2$ acts as a radius of linear convergence, since for $\delta_k \leq R$, the dominating terms in the complexity bounds are only logarithmic in the relative gap δ_k/ε.

The complexity bound presented in Theorem 1 reveals then three phases of convergence: (i) the number of iterations to reach some linear convergence determined by the first term in the complexity bound (7), (ii) the number of iterations to reach the quadratic convergence rate that is captured by the logarithmic terms in the complexity bound (7), (iii) the quadratic convergence phase once δ_k is smaller than the gap of local quadratic convergence $\mathfrak{d}$.

RGGN Algorithm with Line-Search Procedure. Theorem 1 presents an ideal implementation of the RGGN algorithm given the knowledge of all constants to define the regularizations. We can modify the implementation of the RGGN step to select regularizations of the form $\nu_k = \bar{\nu}_k\|\nabla h(g(w_k))\|_2$ that satisfy a sufficient decrease condition by searching over the scaled regularization $\bar{\nu}_k$. Namely, given an iterate w_k and a past scaled regularization $\bar{\nu}_{k-1}$, we can compute the next iterate by searching over $t = 0, 1, \ldots,$ such that the iterate w_{k+1} defined in (2) with $\nu_k = 2^t\bar{\nu}_{k-1}\|\nabla h(g(w_k))\|$ satisfies (5). The chosen scaled regularization $\bar{\nu}_k = 2^{t^*}\bar{\nu}_{k-1}$ for t^* the minimal integer ensuring (5) is kept to initialize the search for the next iterate. The overall algorithm is presented in Algo. 1 in Appendix B.

The proposed search on the scaled regularization parameter preserves the complexity bounds given in Theorem 1 up to a logarithmic factor, both in terms of global and local rates, as shown in Corollary 8 in Appendix B.

Textbook implementations of Gauss-Newton methods methods consider a line-search of the form $w_{k+1} = w_k - \gamma v_k$ for $v_k = -(\nabla g(w_k)\nabla^2 h(g(w_k))\nabla g(w_k)^\top)^{-1}\nabla g(w_k)\nabla h(g(w_k))$ for γ a tunable parameter. Provided that v_k is well-defined, it can be shown to be a descent direction [19], which ensures the termination of such line-search. Here we rather consider using a regularization ν which ensures that the steps of the RGGN algorithm are always defined. By varying the regularization parameter, our method is similar to trust-region methods as done by, e.g., [1].

We present numerical illustrations of the convergence of these algorithms on some nonlinear control problems in Appendix D, where we observe both a global convergence and a fast local convergence of the RGGN algorithm towards an optimal objective.

3. Comparisons to Previous Work

Levenberg-Marquardt methods. Convergence of regularized Gauss-Newton algorithms, i.e., Levenberg-Marquardt methods [16], was studied by e.g. [1, 8, 11, 27, 29]. [1] summarizes the convergence results up to date. [1, Theorem 3.1] shows global convergence rates to stationary points at a polynomial rate. Though these easily translate to global convergence guarantees provided that the non-linear mappings have surjective Jacobians as shown, e.g., by [26, Corrolary 2.1], our results improve on the resulting polynomial rate by detailing the computational complexity of the algorithm as the sum of a constant term depending on the initial gap, a logarithmic term to reach the region of quadratic convergence and a term depending on the logarithm of the logarithm of the target auccracy. [1] and references therein provide local quadratic convergence rates under an error bound condition [1, Assumption 4.2], [27, Eq. (1.6)]. Our assumption on the surjectivity of the Jacobians is stronger than an error bound but it allows a simple treatment of the regularized *generalized* Gauss-Newton algorithms. Moreover, we are able to characterize precisely the region of quadratic convergence for a simple regularized Gauss-Newton algorithm using a bound on the smallest singular value of the transpose of the Jacobian (by considering the case $M_h{=}\theta_h{=}0$, $\rho_h{=}1$).

Modified Gauss-Newton method, a.k.a. prox-linear algorithm. Closer to our approach is the work of [17] where the assumption of surjective Jacobians was analyzed to provide global convergence guarantees of a *modified* Gauss-Newton method. In that purpose, [17] considers minimizing the norm and not the squared norm of non-linear residuals. The generalization of this method to compositional problems with generic Lipschitz-continuous outer cost on top of the non-linear mappings was studied as the prox-linear algorithm [10]. [17] points out that by considering minimizing the norm instead of the squared norms of the residuals, the resulting problem enjoys a better condition number. However, [17] does not take into account the cost of the line-search which consists of solving a one dimensional problem by a trust-region method whose computational complexity is unclear. Similar issues appear when considering the prox-linear algorithm which require an inner convex solver to compute each iteration, without taking into account the potential additional burden of a line-search. In comparison, the implementation of a Levenberg-Marquardt method or a regularized generalized Gauss-Newton method (depending on whether the costs are quadratic or not) can be implemented efficiently in some applications such as nonlinear control [23]. Finally, recently, [9] presented sufficient conditions for local convergence of generalized Gauss-Newton methods. Our work differs by considering the additional Assumption 3 to provide global convergence rates.

References

[1] El Houcine Bergou, Youssef Diouane, and Vyacheslav Kungurtsev. Convergence and complexity analysis of a Levenberg-Marquardt algorithm for inverse problems. *Journal of Optimization Theory and Applications*, 185(3):927–944, 2020.

[2] John Betts. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.

[3] Åke Björck. *Numerical methods for least squares problems*. SIAM, 1996.

[4] Jérôme Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce W Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2):471–507, 2017.

[5] Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical Gauss-Newton optimisation for deep learning. In *International Conference on Machine Learning*, pages 557–565, 2017.

[6] James V Burke and Michael C Ferris. A Gauss-Newton method for convex composite optimization. *Mathematical Programming*, 71(2):179–194, 1995.

[7] Andrzej Cichocki and Shun-ichi Amari. *Adaptive blind signal and image processing: learning algorithms and applications*. John Wiley & Sons, 2002.

[8] Hiroshige Dan, Nobuo Yamashita, and Masao Fukushima. Convergence properties of the inexact Levenberg-Marquardt method under local error bound conditions. *Optimization methods and software*, 17(4):605–626, 2002.

[9] Moritz Diehl and Florian Messerer. Local convergence of generalized Gauss-Newton and sequential convex programming. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3942–3947, 2019.

[10] Dmitriy Drusvyatskiy and Courtney Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, 178(1):503–558, 2019.

[11] Jin-Yan Fan and Ya-Xiang Yuan. On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption. *Computing*, 74(1):23–39, 2005.

[12] James Lincoln Herring, James Nagy, and Lars Ruthotto. Gauss–Newton optimization for phase recovery from the bispectrum. *IEEE Transactions on Computational Imaging*, 6:235–247, 2019.

[13] Kejun Huang and Xiao Fu. Low-complexity proximal Gauss-Newton algorithm for nonnegative matrix factorization. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5, 2019.

[14] Weiwei Li and Emanuel Todorov. Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system. *International Journal of Control*, 80 (9):1439–1453, 2007.

[15] Stanislaw Łojasiewicz. A topological property of real analytic subsets. *Coll. du CNRS, Les équations aux dérivées partielles*, 117(87-89):2, 1963.

[16] Jorge Moré. The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[17] Yurii Nesterov. Modified Gauss-Newton scheme with worst case guarantees for global performance. *Optimization methods and software*, 22(3):469–483, 2007.

[18] Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

[19] James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

[20] Boris Polyak. Some methods of speeding up the convergence of iteration methods. *USSR computational mathematics and mathematical physics*, 4(5):1–17, 1964.

[21] Audrey Repetti, Emilie Chouzenoux, and Jean-Christophe Pesquet. A nonconvex regularized approach for phase retrieval. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 1753–1757, 2014.

[22] Vincent Roulet, Siddhartha Srinivasa, Dmitriy Drusvyatskiy, and Zaid Harchaoui. Iterative linearized control: stable algorithms and complexity guarantees. In *International Conference on Machine Learning*, pages 5518–5527, 2019.

[23] Vincent Roulet, Siddhartha Srinivasa, Maryam Fazel, and Zaid Harchaoui. Complexity bounds of iterative linear quadratic optimization algorithms for discrete time nonlinear control. *arXiv preprint arXiv:2204.02322*, 2022.

[24] Vincent Roulet, Siddhartha Srinivasa, Maryam Fazel, and Zaid Harchaoui. Iterative linear quadratic optimization for nonlinear control: Differentiable programming algorithmic templates. *arXiv preprint arXiv:2207.06362*, 2022.

[25] Athanasios Sideris and James Bobrow. An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems. In *Proceedings of the 2005 American Control Conference*, 2005.

[26] Kenji Ueda and Nobuo Yamashita. On a global complexity bound of the Levenberg-Marquardt method. *Journal of optimization theory and applications*, 147(3):443–453, 2010.

[27] Nobuo Yamashita and Masao Fukushima. On the rate of convergence of the Levenberg-Marquardt method. In *Topics in numerical analysis*, pages 239–249. Springer, 2001.

[28] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

[29] Ruixue Zhao and Jinyan Fan. Global complexity bound of the Levenberg-Marquardt method. *Optimization Methods and Software*, 31(4):805–814, 2016.

Appendix

The Appendix is structured as follows.

1. Appendix A present applications whose problem can be formulated as in (1)
2. Appendix B present the proof of the main theorem. It is decomposed as follows.
   (a) Appendix B.1 presents global convergence results of RGGN algorithms for problems of the form (1) for a convex outer function h satisfying a generic Łojasiewicz inequality given that the inner function satisfies condition (3) and suitable smoothness assumptions.
   (b) Appendix B.2 presents local convergence results of RGGN algorithms for problems of the form (1) for a convex self-concordant outer function h given that the inner function satisfies condition (3) and suitable smoothness assumptions.
   (c) Appendix B.3 details the proofs of Theorem 1 and Corollary 8 for strongly convex outer functions h given that the inner function satisfies condition (3) and suitable smoothness assumptions.
3. Appendix C presents the lemmas used for the convergence analysis.
4. Appendix D presents numerical illustrations of the performance of the RGGN algorithm on some nonlinear control problems.

Appendix A. Problem formulations

In this section, we present how classical problems such as nonlinear control and deep learning can be cast as (1). We denote by semi-columns the concatenation of vectors such that for $x_1, \ldots, x_n \in \mathbb{R}^d$, we have $(x_1; \ldots; x_n) \in \mathbb{R}^{nd}$.

A.1. Nonlinear Control

A discrete time nonlinear control problem with state cost only takes the form

$$\begin{aligned} \min_{\substack{u_0,\ldots,u_{\tau-1}\in\mathbb{R}^{d_u}\\ x_0,\ldots,x_\tau\in\mathbb{R}^{d_x}}} \quad & \sum_{t=1}^{\tau} h_t(x_t) \\ \text{subject to} \quad & x_{t+1} = \phi_t(x_t, u_t), \ \text{for } t \in \{0, \ldots, \tau-1\}, \quad x_0 = \bar{x}_0, \end{aligned} \tag{8}$$

where, at time t, x_t is the state of the system, u_t is the control applied to the system, ϕ_t is the discrete time dynamic, h_t is the cost on the state and $\bar{x}_0$ is a fixed initial point. The problem is entirely characterized by the choice of the control variables and can then be reformulated as

$$\min_{w=(u_0;\ldots;u_{\tau-1})\in\mathbb{R}^{\tau d_u}} h(g(w)) \quad \begin{array}{c}\text{with } g(w) = (x_1; \ldots; x_\tau) \\ \text{s.t. } x_{t+1} = \phi_t(x_t, u_t),\end{array} \qquad h(x) = \sum_{t=1}^{\tau} h_t(x_t)$$

where h encapsulates the total cost on the trajectory $x = (x_1; \ldots; x_\tau)$ and g is the function that, at a given set of control variables $w = (u_0; \ldots; u_{\tau-1})$, associates the corresponding trajectory $x = (x_1; \ldots; x_\tau)$.

Gauss-Newton algorithms for nonlinear control problems have been developed by [14, 25] which use the dynamical structure of the problem to implement Gauss-Newton updates by dynamic programming at a linear computational cost with respect to the horizon τ, namely, the cost of an update is $O(\tau(d_x + d_u)^3)$. Convergence to stationary points of regularized Gauss-Newton algorithms for nonlinear control were studied by [22].

A.2. Deep Learning

A feed-forward deep network of τ layers applied to an input x such as an image can be written as

$$\begin{aligned}
&\psi(x, w) = z_\tau \\
&\quad \text{s.t. } z_t = \phi_t(z_{t-1}, w_t) \quad \text{for } t \in \{1, \ldots, \tau\} \; w = (w_1; \ldots; w_\tau) \quad z_0 = x,
\end{aligned}$$

where w_t are the parameters of the t$^{\text{th}}$ layer ϕ_t that consist, for example, for a multi-layer perception, in a nonlinear function composed with an affine function, i.e., $\phi_t(z_{t-1}, w_t) = a(W_t z_{t-1} + b_t)$ where a is the element-wise application of a nonlinear function such as the sigmoid function and $w_t = (W_t, b_t)$ consists in a matrix of weights W_t and a set of offsets b_t.

Given a dataset of input-output pairs $(x_i, y_i)_{i=1}^n$ and a loss ℓ measuring the error of predicting $\hat{y}$ instead of y as $\ell(\hat{y}, y)$, training a deep network consists in minimizing the loss incurred by predicting the outputs through the deep network, i.e.,

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^n \ell(\phi(x_i, w), y_i).$$

By defining

$$g(w) = (\phi(x_1, w), \ldots, \phi(x_n, w)), \quad h(z) = \sum_{i=1} \ell(z_i, y_i),$$

the training of a deep network can be formulated as in (1).

Here the inner mapping g maps vectors of size $\sum_{t=1}^\tau p_t$ for p_t the dimension of the t^{th} parameter of the deep network to vectors of size $d_\tau n$ for d_τ the dimension of the last layer such as $d_\tau = 1$ for prediction tasks and $d_\tau = k$ for a classification in k classes. As for nonlinear control problems, the dynamical structure of the deep network can be exploited to reduce the computational complexity of an RGGN algorithm from a cubic complexity w.r.t. the depth τ to a linear complexity w.r.t. τ by implementing the oracle with dynamic programming. However, the complexity may remain cubic in the number of parameters, the number of samples and the dimension of the layers.

Stochastic variants of Gauss-Newton algorithms have been developed, see, e.g. [5] and references therein. Our convergence results may help getting convergence rates for such implementations of Gauss-Newton methods in an overparameterized regime such that condition (3) is satisfied.

A.3. Nonlinear Least-Square Problems

In numerous applications such as phase retrieval [7, 12, 21] or non-negative matrix factorization [13], the problem consists in solving a system of nonlinear equations of the form $r_i(w) = 0$ for q nonlinear functions r_i such as $r_i(w) = (y_i - (x_i^\top w)^2)^2$ for phase retrieval problems. One possible approach is then to solve a nonlinear least squares problem of the form

$$\min_{w \in \mathbb{R}^p} \|r(w)\|_2^2 = \sum_{i=1}^q \|r_i(w)\|_2^2$$

where $r(w) = (r_1(w); \ldots ; r_q(w))$, which is clearly of the form (1). Here an RGNN method presented in (2) amounts simply to to a Levenberg-Marquardt method [16] with varying regularizations. Similar algorithms using trust region methods instead of varying regularizations have been developed by, e.g., [1].

Appendix B. Convergence Analysis Proofs

For all results, we define the oracle used by a RGGN algorithm (2) for problem (1) as, for $w \in \mathbb{R}^p$, $\nu > 0$,

$$\begin{aligned}\mathrm{RGGN}_\nu(w) &= \operatorname*{arg\,min}_{v\in\mathbb{R}^p} \left\{ q_h^{g(w)}(\ell_g^w(w+v)) + \frac{\nu}{2}\|v\|_2^2 \right\} \\ &= -(\nabla g(w)\nabla^2 h(g(w))\nabla g(w)^\top + \nu\,\mathrm{I})^{-1}\nabla g(w)\nabla h(g(w)),\end{aligned}$$

such that the iterates of the RGGN algorithm read

$$w_{k+1} = w_k + \mathrm{RGGN}_{\nu_k}(w_k),$$

where we recall that for a function f we denoted by $\ell_f^x(y) = f(x) + \nabla f(x)^\top(y-x)$ and $q_f^x(y) = f(x) + \nabla f(x)^\top(y-x) + (y-x)^\top \nabla^2 f(x)(y-x)/2$ the linear and quadratic approximations of f around x.

B.1. Global Convergence for Gradient Dominating Outer Functions

The global convergence result relies on ensuring a sufficient decrease of the objective at each iteration. Namely, for a given iterate w, we need to select ν that may depend on w such that the next iterate, $w + v$ with $v = \mathrm{RGGN}_\nu(w)$ satisfies

$$\begin{aligned}f(w+v) &\leq \min_{v'\in\mathbb{R}^p} \left\{ q_h^{g(w)}(\ell_g^w(w+v')) + \frac{\nu}{2}\|v'\|_2^2 \right\} \qquad (9)\\ &= q_h^{g(w)}(\ell_g^w(w+v)) + \frac{\nu}{2}\|v\|_2^2 \\ &= f(w) - \frac{1}{2}\nabla f(w)^\top(\nabla g(w)\nabla^2 h(g(w))\nabla g(w)^\top + \nu\,\mathrm{I})^{-1}\nabla f(w).\end{aligned}$$

Theorem 2 provides such regularization, which, combined with an assumption of gradient dominance on the outer function, gives the total complexity in this case.

Theorem 2 *Consider problems of the form* (1) *for a convex outer function h with L_h-Lipschitz-continuous gradients, M_h-Lipschitz-continuous Hessians satisfying a gradient dominating property with parameters $\mu_h > 0, r \in [1/2, 1)$, i.e.,*

$$\forall x \in \mathbb{R}^q,\ \|\nabla h(x)\|_2^2 \geq \mu_h^r \left(h(x) - \min_{y\,\in\mathbb{R}^q} h(x) \right)^r, \qquad (10)$$

and an inner function g ℓ_g-Lipschitz-continuous with L_g-Lipschitz-continuous gradients satisfying that there exists $\sigma_g > 0$, such that

$$\forall w \in \mathbb{R}^p,\ \sigma_{\min}(\nabla g(w)) := \inf_{\lambda\in\mathbb{R}^q} \frac{\|\nabla g(w)\lambda\|_2}{\|\lambda\|_2} \geq \sigma_g > 0. \qquad (11)$$

The sufficient decrease condition (9) *is satisfied for a regularization*

$$\nu(w) = \frac{L_g\|\nabla h(g(w))\|_2}{2}\gamma\left(\frac{L_g\|\nabla h(g(w))\|_2}{4\ell_g^2 L_h(1+\beta)}\right),$$

where $\gamma(x) = 1+\sqrt{1+1/x}$ *and* $\beta = M_h\ell_g^2/(3L_gL_h)$ *is a dimension independent constant.*

If $r = 1/2$*, the number of iterations of the RGGN algorithm* (2) *to converge to an accuracy* ε *for problem* (1) *given regularizations* $\nu_k = \nu(w_k)$ *is at most*

$$k \le 4\theta_g\sqrt{\delta_0}\gamma\left(\frac{\theta_g\sqrt{\delta_0}}{\alpha}\right) + 2\rho_h\ln\left(\frac{\delta_0}{\varepsilon}\right),$$

and, if $1/2 < r < 1$*, the number of iterations to converge to an accuracy* ε *is at most*

$$k \le \frac{2}{2r-1}\frac{\rho_h}{\varepsilon^{2r-1}} + \frac{2}{1-r}\theta_g\delta_0^{1-r} + \sqrt{2\theta_g\alpha}\frac{1}{1-3r/2}\left(\varepsilon^{1-3r/2} - \left(\frac{\alpha}{\theta_g}\right)^{1/r-3/2}\right),$$

with $\rho_h{=}L_h/\mu_h^{2r}$*,* $\rho_g{=}\ell_g/\sigma_g$*,* $\theta_h{=}M_h/(2\mu_h^{3r})$*,* $\theta_g{=}L_g/(\sigma_g^2\mu_h^r)$*,* $\alpha{=}4\rho_g^2(2\rho_g^2\theta_h/(3\theta_g)+\rho_h)$*,* $\delta_0 = f(w^{(0)}) - \min_{v\in\mathbb{R}^p} f(v)$ *and the case* $r = 2/3$ *is to be understood limit-wise.*

Remark 3 *For* $L_g = 0$*, the terms depending uniquely on* δ_0 *vanish since* $\theta_g = 0$ *in this case. We then get the classical rates when minimizing a function* h *that satisfies* (10) *with a first-order method. The rates can be improved by analyzing the local behavior of the algorithm to take advantage of the quadratic approximations taken on the outer function* h*.*

Proof [Proof of Theorem 2] First, note that if h satisfies (10) and g satisfies (11), then for any $w \in \mathbb{R}^p$, we have $\|\nabla(h\circ g)(w)\|_2 \ge \sigma_g\mu_h^r(h(g(w)) - \min_{y\in\mathbb{R}^q} h(y))^r$. Hence for $w^* \in \arg\min f(w)$ with $f = h\circ g$, we get $0 = \|\nabla f(w^*)\|_2 \ge \sigma_g\mu_h^r(h(g(w^*)) - \min_{y\in\mathbb{R}^q} h(y))^r \ge 0$, so we have that $\min_{v\in\mathbb{R}^p} f(v) = \min_{y\in\mathbb{R}^q} h(y)$.

We have from Lemma 9 that for any $w, v \in \mathbb{R}^p$, denoting $a_0 = M_h\ell_g^3/3{+}L_gL_h\ell_g$,

$$|(h\circ g)(w+v){-}q_h^{g(w+v)}\circ\ell_g^w(w+v)| \le \frac{L_g\|\nabla h(g(w))\|_2{+}a_0\|v\|_2}{2}\|v\|_2^2.$$

Since $\|\operatorname{RGGN}_\nu(f)(w)\|_2 \le \ell_g\|\nabla h(g(w))\|_2/\nu$, condition (9) is satisfied for $\nu > 0$ s.t. $a_1 + a_2/\nu \le \nu$, where $a_1 = L_g\|\nabla h(g(w))\|_2$ $a_2 = a_0\ell_g\|\nabla h(g(w))\|_2$. Therefore, denoting $\gamma(x) = 1+\sqrt{1+1/x}$, condition (9) is satisfied for any

$$\nu \ge \nu(w) = \frac{a_1{+}\sqrt{a_1^2{+}4a_2}}{2} = \frac{L_g\|\nabla h(g(w))\|_2}{2}\gamma\left(\frac{L_g^2\|\nabla h(g(w))\|_2}{4a_0\ell_g}\right).$$

We have then for $v = \operatorname{RGGN}_{\nu(w)}(f)(w)$, $G = \nabla g(w)$, $H = \nabla^2 h(g(w))$, since condition (9) is satisfied,

$$\begin{aligned} f(w+v) - f(w) &\le -\frac{1}{2}\nabla h(g(w))^\top G^\top(GHG^\top + \nu(w)\,\mathrm{I})^{-1}G\nabla h(g(w)) \\ &= -\frac{1}{2}\nabla h(g(w))^\top(H + \nu(w)(G^\top G)^{-1})^{-1}\nabla h(g(w)) \\ &\le -\frac{1}{2}\frac{\sigma_g^2}{\sigma_g^2 L_h + \nu(w)}\|\nabla h(g(w))\|_2^2 \le -\frac{b_1x^2}{\sqrt{b_2x^2+b_3x}+b_4x+b_5}, \end{aligned} \tag{12}$$

where $x = \|\nabla h(g(w))\|_2$, $b_1 = \sigma_g^2$, $b_2 = L_g^2$, $b_3 = 4a_0\ell_g$, $b_4 = L_g$, $b_5 = 2\sigma_g^2 L_h$.

The function $f_1 : x \to b_1 x^2/(\sqrt{b_2 x^2 + b_3 x} + b_4 x + b_5)$ is increasing for $x \geq 0$. Hence, denoting $\delta = h(g(w)) - \min_{y\in\mathbb{R}^q} h(y) = f(w) - \min_{v\in\mathbb{R}^p} f(v)$, we have $f_1(x) \geq f_1((\mu_h\delta)^r)$ by assumption (10). Denoting $\delta_k = f(w_k) - \min_{v\in\mathbb{R}^p} f(v)$ for the k^{th} iteration of the ILQR algorithm, we then have $f_2'(\delta_k)(\delta_{k+1} - \delta_k) \leq -1$, with

$$f_2'(\delta) = \frac{1}{f_1((\mu_h\delta)^r)} = \frac{2\rho_h}{\delta^{2r}} + \frac{\theta_g}{\delta^r} + \frac{\theta_g\sqrt{\delta^{2r} + \alpha\delta^r/\theta_g}}{\delta^{2r}} = \frac{2\rho_h}{\delta^{2r}} + \frac{\theta_g\gamma(\theta_g\delta^r/\alpha)}{\delta^r},$$

with $\rho_h{=}L_h/\mu_h^{2r}$, $\rho_g{=}\ell_g/\sigma_g$, $\theta_h{=}M_h/(2\mu_h^{3r})$, $\theta_g{=}L_g/(\sigma_g^2\mu_h^r)$, $\alpha{=}4\rho_g^2(2\rho_g^2\theta_h/(3\theta_g){+}\rho_h)$.

Since f_2 is concave on $\mathbb{R}^+$, we deduce that $f_2(\delta_{k+1}) - f_2(\delta_k) \leq -1$ and so $f_2(\delta_k) \leq -k + f_2(\delta_0)$. Note that f_2 is strictly decreasing, so we get that, for the algorithm to reach an accuracy ε, we need at most $k \leq f_2(\delta_0) - f_2(\varepsilon)$ iterations.

If $r = 1/2$, one can verify that $\delta \to a\ln(2a\sqrt{\delta}\gamma(\sqrt{\delta}/a) + a^2) + 2\sqrt{\delta}\gamma(\sqrt{\delta}/a)$ is an antiderivative of $\delta \to \gamma(\sqrt{\delta}/a)/\sqrt{\delta}$ for any $a > 0$. Hence, for $r = 1/2$, the number of iterations to converge to an accuracy ε is at most

$$\begin{aligned}
k \leq{}& 2\rho_h \ln\left(\frac{\delta_0}{\varepsilon}\right) {+}2\theta_g\left(\sqrt{\delta_0}\gamma\left(\frac{\theta_g\sqrt{\delta_0}}{\alpha}\right) {-}\sqrt{\varepsilon}\gamma\left(\frac{\theta_g\sqrt{\varepsilon}}{\alpha}\right)\right) \\
&{+}\alpha\ln\left(\frac{2\theta_g\sqrt{\delta_0}\gamma(\theta_g\sqrt{\delta_0}/\alpha){+}\alpha}{2\theta_g\sqrt{\varepsilon}\gamma(\theta_g\sqrt{\varepsilon}/\alpha){+}\alpha}\right) \\
\leq{}& 2\rho_h \ln\left(\frac{\delta_0}{\varepsilon}\right) + 2\theta_g\sqrt{\delta_0}\gamma\left(\frac{\theta_g\sqrt{\delta_0}}{\alpha}\right) + \alpha\ln\left(1{+}2\frac{\theta_g\sqrt{\delta_0}}{\alpha}\gamma\left(\frac{\theta_g\sqrt{\delta_0}}{\alpha}\right)\right).
\end{aligned}$$

By using that $\ln(1 + x) \leq x$ for $x > -1$, we get the claimed bound in this case.

If $1/2 < r < 1$, by integrating f_2, the number of iterations to converge to an accuracy ε is at most

$$k \leq \frac{2\rho_h}{2r-1}\left(\frac{1}{\varepsilon^{2r-1}} - \frac{1}{\delta_0^{2r-1}}\right) + \frac{\theta_g}{(1-r)}\left(\delta_0^{1-r} - \varepsilon^{1-r}\right) + \int_\varepsilon^{\delta_0} \frac{\theta_g\sqrt{x^{2r} + \alpha x^r/\theta_g}}{x^{2r}} dx.$$

The bound follows in this case by using that, for $1/2 < r < 1$, and $a > 0$,

$$\int_\varepsilon^{\delta_0} \frac{\sqrt{x^{2r} + a x^r}}{x^{2r}} dx \leq \int_\varepsilon^{a^{1/r}} \frac{\sqrt{2a}}{x^{3r/2}} dx + \int_{a^{1/r}}^{\delta_0} \frac{1}{x^r} dx.$$

■

B.2. Local Convergence under Self-Concordance Conditions

As we analyze the RGGN algorithm locally as an approximate Newton method on the outer function, we use the notations and assumptions used to analyze a Newton method. Namely, we assume the outer function h strictly convex and we define the norm induced by the Hessian of h at a point $x \in \mathbb{R}^q$ and its dual norm as, respectively, for $y \in \mathbb{R}^q$,

$$\|y\|_x = \sqrt{y^\top \nabla^2 h(x) y}, \quad \|y\|_x^* = \sqrt{y^\top \nabla^2 h(x)^{-1} y}.$$

For a matrix $A \in \mathbb{R}^{q \times p}$, we denote $\|A\|_x = \|\nabla^2 h(x)^{1/2} A\|_2$ the norm induced by the local geometry of h w.r.t. the Euclidean norm. Finally we denote the Newton decrement of the cost function, as, for $x \in \mathbb{R}^p$,

$$\lambda_h(x) = \sqrt{\nabla h(x)^\top \nabla^2 h(x)^{-1} \nabla h(x)}.$$

To analyze the local convergence of the RGGN algorithm we consider the outer function to be self-concordant [18, Section 5]. In addition, we consider smoothness properties of the inner function g with respect to the geometry induced by the Hessian of the outer function h as presented in the assumptions below.

Assumption 4 *We consider that the outer function h is strictly convex and that the following constants are finite*

$$\ell = \sup_{\substack{w,v \in \mathbb{R}^p \\ v \neq 0}} \frac{\|g(w+v) - g(w)\|_{g(w)}}{\|v\|_2}, \quad L = \sup_{\substack{w,v \in \mathbb{R}^p \\ v \neq 0}} \frac{\|\nabla g(w+v)^\top - \nabla g(w)^\top\|_{g(w)}}{\|v\|_2}$$

$$\vartheta_h = \sup_{\substack{x, y_1, y_2, y_3 \in \mathbb{R}^q \\ y_1 \neq 0, y_2 \neq 0, y_3 \neq 0}} \frac{|\nabla^3 h(x)[y_1, y_2, y_3]|}{2\|y_1\|_x \|y_2\|_x \|y_3\|_x}, \quad \sigma = \inf_{\substack{w \in \mathbb{R}^p, \mu \in \mathbb{R}^q \\ \mu \neq 0}} \frac{\|\nabla g(w)\mu\|_2}{\|\mu\|^*_{g(w)}}.$$

In consequence, h is ϑ_h-self concordant [18, Definition 5.1.1, Lemma 5.1.2] and we have that $\sigma \leq \sigma_{\min}(\nabla g(w) \nabla^2 h(g(w))^{1/2})$, $\sigma_{\max}(\nabla g(w) \nabla^2 h(g(w))^{1/2}) \leq \ell$, for any $w \in \mathbb{R}^p$.

Assumption 4 is satisfied if the outer function h is μ_h-strongly convex with L_h-Lipschitz continuous gradients, M_h-Lipschitz continuous Hessians and the outer function is ℓ_g-Lipschitz continuous, with L_g-Lipschitz continuous gradients and satisfies that $\sigma_{\min}(\nabla g(w)) \geq \sigma_g$ for all $w \in \mathbb{R}^p$. In that case, we have

$$\ell \leq \sqrt{L_h} \ell_g, \quad L \leq \sqrt{L_h} L_g, \quad 2\vartheta_h \leq M_h / \mu_h^{3/2}, \quad \sigma \geq \sqrt{\mu_h} \sigma_g. \tag{13}$$

Equipped with a stepsize proportional to the Newton decrement, we can show a local quadratic convergence rate of the RGGN algorithm given Assumption 4.

Theorem 5 *Given Assumption 4, consider the RGGN algorithm (2) for problem (1) with regularizations of the form $\nu_k = \bar{\nu} \lambda_h(g(w_k))$ for some $\bar{\nu} \geq 0$. For $k \geq 0$ such that*

$$\lambda_h(g(w_k)) < \lambda = \frac{1}{\max\{4\vartheta_h + 3\vartheta_g + 2\bar{\nu}/\sigma^2, 2\varrho\vartheta_h\}}, \tag{14}$$

where $\varrho = \ell/\sigma$ and $\vartheta_g = L/\sigma^2$, we have $\lambda_h(g(w_{k+1})) \leq \lambda^{-1} \lambda_h(g(w_k))^2$, and the RGGN algorithm converges quadratically to the minimum value of problem (1).

Remark 6 *If h is a quadratic, such that the algorithm reduces to a regularized Gauss-Newton algorithm, a.k.a. Levenberg-Marquardt method, and $\vartheta_h = 0$, the radius of quadratic convergence reduces to $\lambda = 1/(3\vartheta_g + 2\bar{\nu})$. If in addition, no regularization is in effect such that the algorithm reduces simply to a Gauss-Newton algorithm, the radius of convergence reduces to $\lambda = 1/3\vartheta_g$, which can be expressed as $1/(3\theta_g \sqrt{\rho_h})$ if the total cost is μ_h strongly convex with θ_g, ρ_h defined as in Theorem 2 and σ, L expressed using Eq. (13). So up to $3\sqrt{\rho_h}$, the parameter $1/\theta_g$ acts again as a radius of fast convergence as in Theorem 2.*

Remark 7 *For better readability, we simplified the expression of the radius of convergence. A closer look at the proof shows that a positive regularization may lead to a larger radius of convergence than no regularization.*

Proof [Proof of Theorem 5] Let $w \in \mathbb{R}^p$ $G{=}\nabla g(w)$, $H{=}\nabla^2 h(g(w))$, $v{=}\mathrm{RGGN}_\nu(f)(w)$ with $\nu = \bar{\nu}\lambda_h(g(w))$. Assume that

$$\lambda_h(g(w)) \leq 1/\max\{\sqrt{2\vartheta_h\vartheta_g}c_1, 2\vartheta_h\varrho c_2, 2\vartheta_h c_2\},$$

where $c_1{=}\max\{1{-}\bar{\nu}/(\sqrt{2\vartheta_h L\ell}), 0\}$, $c_2{=}\max\{1{-}\bar{\nu}/(2\ell^2\vartheta_h), 0\}$, $\varrho{=}\ell/\sigma$, $\vartheta_g{=}L/\sigma^2$. We have

$$\lambda_h(g(w{+}v)) {\leq} \underbrace{\|\nabla h(g(w{+}v)){-}\nabla h(g(w){+}G^\top v)\|^*_{g(w+v)}}_{A} + \underbrace{\|\nabla h(g(w){+}G^\top v)\|^*_{g(w+v)}}_{B}. \qquad (15)$$

Bounding A in (15). By definition of ℓ in Assumption 4 and Lemma 12, we have

$$\|g(w+v) - g(w)\|_{g(w)} \leq \ell\|v\|_2, \qquad \|v\|_2 \leq \frac{\ell\lambda_h(g(w))}{\ell\sigma + \bar{\nu}\lambda_h(g(w))}. \qquad (16)$$

One easily verifies that $x/(1+ax) \leq c$ if $0 \leq x \leq c/\max\{1-ca, 0\}$ for any $a, c > 0$. So for $\lambda_h(g(w))) \leq 1/(2\vartheta_h\varrho c_2)$, we have $\|g(w+v) - g(w)\|_{g(w)} \leq 1/(2\vartheta_h)$. Hence, using that h is ϑ_h-self-concordant, [18, Theorem 5.1.7] applies and by using the definition of L in Assumption 4, we have

$$\begin{aligned}\|g(w{+}v){-}g(w){-}G^\top v\|_{g(w+v)} &\leq \frac{1}{1{-}\vartheta_h\|g(w{+}v){-}g(w)\|_{g(w)}}\|g(w{+}v){-}g(w){-}G^\top v\|_{g(w)} \\ &\leq 2\left\|\int_0^1 \nabla g(w{+}tv)^\top v dt - \nabla g(w)^\top v\right\|_{g(w)} = L\|v\|_2^2.\end{aligned}$$

Using (16), for $\lambda_h(g(w)) \leq 1/(\sqrt{2\vartheta_h\vartheta_g}c_1)$, we get $\|g(w{+}v){-}g(w){-}G^\top v\|_{g(w+v)}{\leq}1/(2\vartheta_h)$. Since the total cost h is ϑ_h-self-concordant, we can then use Lemma 13 to obtain

$$\begin{aligned}A &\leq \frac{1}{1 - \vartheta_h\|g(w+v) - g(w) - G^\top v\|_{g(w+v)}}\|g(w+v) - g(w) - G^\top v\|_{g(w+v)} \\ &\leq \frac{2L\ell^2\lambda_h(g(w))^2}{(\ell\sigma + \bar{\nu}\lambda_h(g(w)))^2}.\end{aligned} \qquad (17)$$

Bounding B in (15). Recall that for $\lambda_h(g(w))) \leq 1/(2\vartheta_h\varrho c_2)$, we have $\|g(w{+}v){-}g(w)\|_{g(w)} \leq 1/(2\vartheta_h)$. Since h is ϑ_h-self-concordant, we have then [18, Theorem 5.1.7],

$$B \leq \frac{1}{1{-}\vartheta_h\|g(w{+}v){-}g(w)\|_{g(w)}}\|\nabla h(g(w){+}G^\top v)\|^*_{g(w)} \leq 2\|\nabla h(g(w){+}G^\top v)\|^*_{g(w)}. \qquad (18)$$

Denote $\nu = \bar{\nu}\lambda_h(g(w))$ and define $n = -(H + \nu(G^\top G)^{-1})^{-1}\nabla h(g(w))$. Using that

$$v = -G(G^\top G)^{-1}(H + \nu(G^\top G)^{-1})^{-1}\nabla h(g(w)),$$

and denoting $x = g(w)$, we have then

$$\begin{aligned}\|\nabla h(g(w) + G^\top v)\|_{g(w)}^* &= \|\nabla h(x+n) - \nabla h(x) - (H + \nu(G^\top G)^{-1})n\|_x^* \\ &\leq \|\nabla h(x+n) - \nabla h(x) - Hn\|_x^* + \nu\|(G^\top G)^{-1}n\|_x^*. \qquad (19)\end{aligned}$$

The first term can be bounded as in the proof of local convergence of a Newton method [18, Theorem 5.2.2]. Namely, we have

$$\|\nabla h(x+n) - \nabla h(x) - Hn\|_x^* = \|\int_0^1 (\nabla^2 h(x+tn) - \nabla^2 h(x))n dt\|_x^*.$$

Since $\sigma_{\max}(\nabla g(w)\nabla^2 h(g(w))^{1/2}) \leq \ell$, we have

$$\|n\|_x = \|(\mathrm{I} + \nu H^{-1/2}(G^\top G)^{-1}H^{-1/2})^{-1}H^{-1/2}\nabla h(g(w))\|_2 \leq \frac{\lambda_h(g(w))}{1 + \bar{\nu}\ell^{-2}\lambda_h(g(w))}.$$

So if $\lambda_h(g(w)) \leq 1/(2\vartheta_h c_2)$, we get $\|n\|_x \leq 1/(2\vartheta_h)$ and, since h is self-concordant, by [18, Corollary 5.1.5], we have, denoting $J = \int_0^1(\nabla^2 h(x+tn) - \nabla^2 h(x))dt$,

$$(-\|n\|_x\vartheta_h + \|n\|_x^2\vartheta_h^2/3)\nabla^2 h(x) \preceq J \preceq \frac{\|n\|_x\vartheta_h}{1 - \|n\|_x\vartheta_h}\nabla^2 h(x).$$

Moreover, since $\|n\|_x < 1/(2\vartheta_h)$, we have $\|n\|_x\vartheta_h - \|n\|_x^2\vartheta_h^2/3 \leq \frac{\|n\|_x\vartheta_h}{1-\|n\|_x\vartheta_h}$. Hence we get

$$\|\nabla h(x+n) - \nabla h(x) - Hn\|_x^* \leq \frac{\|n\|_x^2\vartheta_h}{1 - \|n\|_x\vartheta_h} \leq \frac{2\lambda_h(g(w))^2\vartheta_h}{(1 + \bar{\nu}\ell^{-2}\lambda_h(g(w)))^2}. \qquad (20)$$

On the other hand, since $\sigma \leq \sigma_{\min}(\nabla g(w)\nabla^2 h(g(w))^{1/2})$, we have

$$\|(G^\top G)^{-1}n\|_{g(w)}^* = \|(H^{1/2}G^\top G H^{1/2} + \nu\,\mathrm{I})^{-1}H^{-1/2}\nabla h(g(w))\|_2 \leq \frac{\lambda_h(g(w))}{\sigma^2 + \bar{\nu}\lambda_h(g(w))}. \qquad (21)$$

So combining (21) and (20) into (19) and then (18) we get

$$B \leq 2\left(\frac{2\vartheta_h}{(1 + \bar{\nu}\ell^{-2}\lambda_h(g(w)))^2} + \frac{\bar{\nu}}{\sigma^2 + \bar{\nu}\lambda_h(g(w))}\right)\lambda_h(g(w))^2. \qquad (22)$$

Local quadratic convergence rate. So combining (17) and (22) into (15), we get, as long as $\lambda_h(g(w)) \leq 1/\max\{\sqrt{2\vartheta_h\vartheta_g}c_1, 2\vartheta_h\varrho c_2, 2\vartheta_h c_2\}$,

$$\lambda_h(g(w+v)) \leq \left(\frac{2L\ell^2}{(\ell\sigma + \bar{\nu}\lambda_h(g(w)))^2} + \frac{4\vartheta_h}{(1+\bar{\nu}\ell^{-2}\lambda_h(g(w)))^2} + \frac{2\bar{\nu}}{\sigma^2 + \bar{\nu}\lambda_h(g(w))}\right)\lambda_h(g(w))^2.$$

Note that $c_1, c_2 \leq 1$ and that $2\vartheta_g + 4\vartheta_h + 2\bar{\nu}/\sigma^2 \geq \max\{2\vartheta_h, \sqrt{2\vartheta_h\vartheta_g}\}$, using the arithmetic-geometric mean inequality. Hence, for

$$\lambda_h(g(w)) < \lambda = 1/\max\{2\vartheta_g + 4\vartheta_h + 2\bar{\nu}/\sigma^2, 2\vartheta_h\varrho\},$$

we get $\lambda_h(g(w+v)) \leq \bar{\lambda}^{-1}\lambda_h(g(w))^2 < \lambda_h(g(w))$, that is, we reach the region of quadratic convergence for $g(w)$. ■

B.3. Total Complexity for Strongly Convex Outer Functions

If the inner function and the outer function of problem (1) satisfy the smoothness assumptions of Theorem 2 and condition (3), then strong convexity of the outer function ensures both a gradient dominating property as in (10) and self-concordance assumptions as presented in Assumption 4. We get then both global convergence and local quadratic convergence as stated in Theorem 1 whose statement and proof are presented below.

Theorem 1 *Consider problems of the form (1) for an outer function h μ_h-strongly convex with L_h-Lipschitz continuous gradients, M_h-Lipschitz-continuous Hessians and an inner function g ℓ_g-Lipschitz-continuous with L_g-Lipschitz-continuous gradients satisfying (3).*

The number of iterations of an RGGN algorithm (2), with regularizations

$$\nu_k = \left(1 + \frac{\alpha}{2(1+\theta_g\|\nabla h(g(w_k))\|_2/(\sqrt{\mu_h}\rho_g))}\right) L_g\|\nabla h(g(w_k))\|_2, \tag{6}$$

needed to reach an accuracy ε is at most

$$k(\delta_0, \varepsilon) := 4\theta_g\left(\sqrt{\delta_0} - \sqrt{\varepsilon}\right) + 2\rho_h \ln\left(\frac{\delta_0}{\varepsilon}\right) + 2\alpha \ln\left(\frac{\theta_g\sqrt{\delta_0}+\rho_g}{\theta_g\sqrt{\varepsilon}+\rho_g}\right), \tag{7}$$

where $\rho_h = L_h/\mu_h$, $\rho_g = \ell_g/\sigma_g$, $\theta_g = L_g/(\sigma_g^2\sqrt{\mu_h})$, $\theta_h = M_h/(2\mu_h^{3/2})$, $\alpha = 4\rho_g^2\rho_h(\beta+1)$, $\beta = M_h\ell_g^2/(3L_gL_h)$ and $\delta_0 = f(w^{(0)}) - \min_{w\in\mathbb{R}^p} f(w)$.

If the desired target accuracy ε is smaller than a gap $\delta = 1/(32\rho_h(\theta_h(1+\sqrt{\rho_h}\rho_g^3/3) + \sqrt{\rho_h}\theta_g(1+\rho_g\rho_h))^2)$ which determines a quadratic convergence phase, the number of iterations of an RGGN algorithm, with regularization ν_k defined above, needed to reach the accuracy ε is at most $k(\delta_0, \delta) + \ln\ln(\varepsilon^{-1})$.

Proof By using the strong convexity of the costs h, we can refine the choice of the regularization to ensure (9). The validity of the proposed regularization to ensure condition (9) is shown in Lemma 10. With the proposed regularization, as shown in Lemma 11, following the same reasoning as in the proof of Theorem 2, we have that the number of iterations of the RGGN algorithm needed to reach an accuracy ε is bounded by

$$k \leq 2\rho_h \ln\left(\frac{\delta_0}{\varepsilon}\right) + 4\theta_g\left(\sqrt{\delta_0} - \sqrt{\varepsilon}\right) + 2\alpha \ln\left(\frac{\theta_g\sqrt{\delta_0}+\rho_g}{\theta_g\sqrt{\varepsilon}+\rho_g}\right), \tag{23}$$

with ρ_h, ρ_g, θ_h, θ_g, α defined as in the claim.

For the local convergence, the constants in Theorem 5 can be expressed in terms of the constants in Theorem 2 as $\sigma = \sqrt{\mu_h}\sigma_g, \vartheta_h = \theta_h, \vartheta_g = \sqrt{\rho_h}\theta_g, \varrho = \sqrt{\rho_h}\rho_g$. From the proof of Theorem 5, if $\lambda_h(g(w_k)) \leq 1/\max\{\sqrt{2\vartheta_h\vartheta_g}, 2\vartheta_h\varrho, 2\vartheta_h\}$, then

$$\lambda_h(g(w_{k+1})) \leq \left(2\vartheta_g + 4\vartheta_h + \frac{2\nu_k}{\sigma^2\lambda_h(g(w_k))}\right)\lambda_h(g(w_k))^2,$$

where $\nu_k/\lambda_h(g(w_k)) \leq \sqrt{L_h}(L_g + 2\ell_g(M_h\ell_g^2/3{+}L_gL_h)/(\sigma_g\mu_h))$. Define then

$$\lambda = \frac{1}{4(\theta_h(1+\sqrt{\rho_h}\rho_g^3/3) + \sqrt{\rho_h}\theta_g(1+\rho_g\rho_h))}.$$

We have that $\lambda \leq 1/\max\{\sqrt{2\vartheta_h\vartheta_g}, 2\vartheta_h\varrho, 2\vartheta_h\}$. So, if $\lambda_h(g(w_k)) \leq \lambda$, quadratic convergence is ensured.

It remains to link the objective gap to the Newton decrement. By considering a gradient step with step-size $1/L_h$, we have $\|\nabla h(x)\|^2 \leq 2L_h(h(x) - h^*)$ for any x, hence $\lambda_h(x) \leq \sqrt{2\rho_h(h(x) - h^*)}$. So, the number of iterations to reach quadratic convergence is bounded by the number of iterations to get an accuracy $\delta = \lambda^2/(2\rho_h)$. Once quadratic convergence is reached the remaining number of iterations is of the order of $O(\ln\ln\varepsilon^{-1})$. ■

The proof of Theorem 1 reveals that as long as ν_k is chosen to ensure the sufficient decrease condition (9) while being proportional to the norm of the gradient of the outer function, the results still hold. Hence the line-search procedure presented in Algo. 1 is a practical implementation of the RGGN algorithm that keeps its convergence behavior as stated in Corollary 8.

Algorithm 1 Regularized Generalized Gauss-Newton Algorithm with Line-Search

```
Inputs: Initial point $w_0$, initial scaled regularization $\bar{\nu}_{-1} > 0$, twice differentiable outer function $h$, differentiable inner function $g$
for $k = 0, \ldots$ do
    Set $\bar{\nu}_k = \bar{\nu}_{k-1}$, $\nu_k = \bar{\nu}_k\|\nabla h(g(w_k))\|_2$
    Compute $w_{k+1} = w_k - (\nabla g(w_k)\nabla^2 h(g(w_k))\nabla g(w_k)^\top + \nu_k \mathrm{I})^{-1}\nabla g(w_k)\nabla h(g(w_k))$
    while $f(w_{k+1}) > f(w_k) - \frac{1}{2}\nabla f(w_k)^\top(\nabla g(w_k)\nabla^2 h(g(w_k))\nabla g(w_k)^\top + \nu_k \mathrm{I})^{-1}\nabla f(w_k)$ do
        Set $\bar{\nu}_k \leftarrow 2\bar{\nu}_k$, $\nu_k \leftarrow \bar{\nu}_k\|\nabla h(g(w_k))\|_2$
        Set $w_{k+1} \leftarrow w_k - (\nabla g(w_k)\nabla^2 h(g(w_k))\nabla g(w_k)^\top + \nu_k \mathrm{I})^{-1}\nabla g(w_k)\nabla h(g(w_k))$
    end
end
```

Corollary 8 *Consider the assumptions and notations of Theorem 1 on problem* (1) *and Algo. 1 with an initial scaled regularization guess $\bar{\nu}_{-1} \leq \left(1 + \alpha/(2 + 2\theta_g\sqrt{\delta_0}/\rho_g)\right) L_g$. The total number of calls to oracles of the form $w, \nu \to -(\nabla g(w)\nabla^2 h(g(w))\nabla g(w)^\top + \nu\,\mathrm{I})^{-1}\nabla g(w)\nabla h(g(w))$ of Algo. 1 to reach an accuracy $\varepsilon \leq \delta'$ is at most $2k(\delta_0, \delta') + \ln\ln(\varepsilon^{-1}) + \lceil\log_2((1 + \alpha/2)L_g/\bar{\nu}_{-1})\rceil$, where $k(\delta_0, \delta')$ is defined as in Theorem 1 and $\delta' = 1/(32\rho_h(\theta_h(1 + 2\sqrt{\rho_h}\rho_g^3/3) + \sqrt{\rho_h}\theta_g(1 + 2\rho_g\rho_h))^2)$ is a gap of quadratic convergence for Algo. 1.*

Proof Define for $w \in \mathbb{R}^p$,

$$\bar{\nu}(w) = \left(1 + \frac{\alpha}{2(1 + \theta_g\sqrt{f(w) - \min_{v\in\mathbb{R}^p} f(v)}/\rho_g))}\right) L_g$$

Since h is strongly convex, we have that $\|\nabla h(g(w))\|_2 \geq \sqrt{\mu_h}(h(g(w)) - \min_{y\in\mathbb{R}^q} h(y)) = \sqrt{\mu_h}(f(w) - \min_{v\in\mathbb{R}^q} f(v))$, where we recall that $\min_{y\in\mathbb{R}^q} h(y) = \min_{v\in\mathbb{R}^q} f(v)$ as shown in Theorem 2. Hence we have that $\bar{\nu}(w)\|\nabla h(g(w))\|_2 \geq \nu(w)$ for $\nu(w)$ defined in Lemma 10. Therefore by Lemma 10, the line-search procedure of Algo. 1 at the k^{th} iteration necessarily terminates with a scaled regularization $\bar{\nu}_k \leq 2\bar{\nu}(w_k)$ since we chose $\bar{\nu}_{-1} \leq \bar{\nu}(w_0)$ and since $\bar{\nu}(w_k)$ necessarily increases over the iterations as $f(w_k)$ decreases when the decrease condition (9) is satisfied.

Moreover, since $\bar{\nu}(w)$ is upper bounded by $(1 + \alpha/2)L_g$ the total number of calls to oracles made by the line-search inner loop to satisfy the decrease condition after k iterations is at most

$$k + \left\lceil \log_2\left(\frac{(1 + \alpha/2)L_g}{\bar{\nu}_{-1}}\right)\right\rceil.$$

Since the line-search ensures the decrease condition (9), we have, as in Lemma 11 that for $\nu_k = \bar{\nu}_k \|\nabla h(g(w_k))\|_2$,

$$
\begin{aligned}
f(w_{k+1}) - f(w_k) &\leq -\frac{1}{2}\frac{\sigma_g^2}{\sigma_g^2 L_h + \nu_k}\|\nabla h(g(w))\|_2^2 \\
&\leq -\frac{1}{2}\frac{\sigma_g^2}{\sigma_g^2 L_h + 2\bar{\nu}(w_k)\|\nabla h(g(w))\|_2}\|\nabla h(g(w))\|_2^2 \\
&\leq -\frac{1}{4}\frac{\sigma_g^2}{\sigma_g^2 L_h + \bar{\nu}(w_k)\|\nabla h(g(w))\|_2}\|\nabla h(g(w))\|_2^2.
\end{aligned}
$$

The rest of the proof of Lemma 11 follows and we get that the number of iterations of Algo. 1 to reach an accuracy ε is at most $2k(\delta_0, \varepsilon)$ for $k(\delta_0, \varepsilon)$ defined as in Theorem 1.

For the quadratic convergence rate, we have, with the notations of the proof of Theorem 1, that $\nu_k/\lambda_h(g(w_k)) \leq 2\bar{\nu}(w_k)\|\nabla h(g(w_k))\|_2^2/\lambda_h(g(w_k)) \leq 2\sqrt{L_h}(L_g + 2\ell_g(M_h\ell_g^2/3 + L_g L_h)/(\sigma_g\mu_h))$. The rest of the proof follows with a slightly modified quadratic convergence gap. ■

Appendix C. Helper Lemmas

C.1. Global Convergence Analysis Lemmas

Lemma 9 states that a linear quadratic approximation of the compositional objective in (1) approximates the objective up to a cubic error. Recall that for a function f we denote by $\ell_f^x(y) = f(x) + \nabla f(x)^\top(y-x)$ and $q_f^x(y) = f(x) + \nabla f(x)^\top(y-x) + (y-x)^\top\nabla^2 f(x)(y-x)/2$ the linear and quadratic approximations of f around x.

Lemma 9 *For $h : \mathbb{R}^q \to \mathbb{R}$ with L_h-Lipschitz continuous gradients and M_h-Lipschitz continuous Hessians, and $g : \mathbb{R}^p \to \mathbb{R}^q$ ℓ_g-Lipschitz continuous with L_g-Lipschitz continuous gradients, we have for any $w, v \in \mathbb{R}^p$,*

$$
|(h\circ g)(w+v) - q_h^{g(w)}(\ell_g^w(w+v))| \leq \frac{L_g\|\nabla h(g(w))\|_2 + (M_h\ell_g^3/3 + L_g L_h \ell_g)\|v\|_2}{2}\|v\|_2^2.
$$

Proof In the proof, for a function f, we denote by $\bar{\ell}_f^x(y) = \nabla f(x)^\top y$ and $\bar{q}_f^x(y) = \nabla f(x)^\top y + y^\top \nabla^2 f(x) y/2$ the linear and quadratic expansions of f around x such that the linear quadratic approximation of the composite objective $f = h\circ g$ around w is given as $f(w+v) \approx q_h^{g(w)}(\ell_g^w(w+v)) = f(w) + \bar{q}_h^{g(w)}(\bar{\ell}_g^w(v))$. We have for any $w, v \in \mathbb{R}^p$,

$$
\begin{aligned}
|h(g(w+v)) - h(g(w)) - \bar{q}_h^{g(w)}(\bar{\ell}_g^w(v))| \leq\ & |h(g(w+v)) - h(g(w)) - \bar{q}_h^{g(w)}(g(w+v) - g(w))| \\
&+ |\bar{q}_h^{g(w)}(g(w+v) - g(w)) - \bar{q}_h^{g(w)}(\bar{\ell}_g^w(v))|.
\end{aligned}
$$

On one hand, we have, by Taylor-Lagrange inequality,

$$
|h(g(w+v)) - h(g(w)) - \bar{q}_h^{g(w)}(g(w+v) - g(w))| \leq \frac{M_h}{6}\|g(w+v) - g(w)\|_2^3 \leq \frac{M_h\ell_g^3}{6}\|v\|_2^3.
$$

On the other hand, we have,

$$\begin{aligned}|\bar{q}_h^{g(w)}(g(w{+}v){-}g(w)){-}\bar{q}_h^{g(w)}(\bar{\ell}_g^w(v))| &= \Big|(g(w{+}v){-}g(w){-}\nabla g(w)^\top v)^\top \nabla h(g(w)) \\ &\quad + \frac{1}{2}(g(w{+}v){-}g(w){-}\nabla g(w)^\top v)^\top \nabla^2 h(g(w))(g(w{+}v){-}g(w){+}\nabla g(w)^\top v)\Big| \\ &\le \frac{L_g \|\nabla h(g(w))\|_2}{2}\|v\|_2^2 + \frac{L_h L_g \ell_g}{2}\|v\|_2^3.\end{aligned}$$

■

Lemma 10 refines the regularization choice of Theorem 2 by exploiting an additional assumption of strong convexity of the outer function.

Lemma 10 *For $h : \mathbb{R}^q \to \mathbb{R}$ μ_h-strongly convex, L_h-smooth with M_h-smooth Hessians, and $g : \mathbb{R}^p \to \mathbb{R}^q$ ℓ_g-Lipschitz continuous and L_g-smooth such that $\sigma_{\min}(\nabla g(w)) \geq \sigma_g > 0$ for all $w \in \mathbb{R}^p$, condition (9) is satisfied by choosing a regularization*

$$\nu \geq \nu(w) = \left(1 + \frac{\alpha}{2(1+\theta_g\|\nabla h(g(w))\|_2/(\sqrt{\mu_h}\rho_g))}\right) L_g \|\nabla h(g(w))\|_2.$$

where $\rho_h = L_h/\mu_h$, $\rho_g = \ell_g/\sigma_g$, $\theta_g = L_g/(\sigma_g^2\sqrt{\mu_h})$, $\theta_h = M_h/(2\mu_h^{3/2})$, $\alpha = 4\rho_g^2(2\rho_g^2\theta_h/(3\theta_g) + \rho_h)$.

Proof Let $w \in \mathbb{R}^p$, $G = \nabla g(w)$, $H = \nabla^2 h(g(w))$. We have

$$\begin{aligned}\mathrm{RGGN}_\nu(f)(w) &= -G(G^\top G)^{-1}(H + \nu(G^\top G)^{-1})^{-1}\nabla h(g(w)) \\ &= -G(G^\top G)^{-1/2}((G^\top G)^{1/2}H(G^\top G)^{1/2} + \nu\,\mathrm{I})^{-1}(G^\top G)^{1/2}\nabla h(g(w)).\end{aligned}$$

We have then

$$\begin{aligned}\|\,\mathrm{RGGN}_\nu(f)(w)\|_2/\|\nabla h(g(w))\|_2 &\leq \min\{\ell_g^2/(\mu_h\sigma_g\ell_g^2 + \nu\sigma_g), \ell_g/(\nu + \mu_h\sigma_g^2)\} \\ &\leq 2\ell_g/(\nu(1+\sigma_g/\ell_g) + \mu_h\sigma_g(\sigma_g+\ell_g)) \\ &\leq 2\ell_g/(\nu + \mu_h\sigma_g\ell_g),\end{aligned}$$

where we used that $\min\{a,b\} \leq 2/(1/a+1/b)$. Hence condition (5) is satisfied if ν satisfies $a_1 + a_2/(a_3+\nu) \leq \nu$ with $a_1 = L_g\|\nabla h(g(w))\|_2$, $a_2 = 2a_0\ell_g\|\nabla h(g(w))\|_2$, $a_3 = \sigma_g\ell_g\mu_h$, $a_0 = M_h\ell_g^3/3{+}L_gL_h\ell_g$. Hence condition (5) is satisfied for $\nu \geq \nu_0 = (a_1 - a_3 + (a_1 + a_3)\sqrt{1+4a_2(a_1+a_3)^{-2}})/2$. The result follows using that since $\sqrt{1+2x} \leq 1+x$, we have $\nu_0 \leq a_1 + a_2/(a_1+a_3)$. Hence it suffices to select a regularization larger than

$$\begin{aligned}\nu(w) &= L_g\|\nabla h(g(w))\|_2 + \frac{2\ell_g^2(M_h\ell_g^2/3{+}L_gL_h)\|\nabla h(g(w))\|_2}{L_g\|\nabla h(g(w))\|_2 + \sigma_g\ell_g\mu_h} \\ &= \left(1 + \frac{2\rho_g(2\theta_h\rho_g^2/(3\theta_g) + \rho_h)}{1+\theta_g\|\nabla h(g(w))\|_2/(\sqrt{\mu_h}\rho_g)}\right) L_g\|\nabla h(g(w))\|_2.\end{aligned}$$

■

Lemma 11 details the computations of the complexity bounds of the RGGN algorithm in the case of a strongly convex outer function used in Eq. (23) before taking into account the local quadratic convergence.

Lemma 11 *Consider problems of the form (1) for an outer function h μ_h-strongly convex with L_h-Lipschitz continuous gradients and M_h-Lipschitz continuous Hessians, and $g : \mathbb{R}^p \to \mathbb{R}^q$ ℓ_g-Lipschitz continuous with L_g-Lipschitz continuous gradients satisfying that $\sigma_{\min}(\nabla g(w)) \geq \sigma_g > 0$ for all $w \in \mathbb{R}^p$, The number of iterations of an RGGN algorithm (2), with regularizations*

$$\nu_k = \left(1 + \frac{\alpha}{2(1 + \theta_g \|\nabla h(g(w_k))\|_2/(\sqrt{\mu_h}\rho_g))}\right) L_g \|\nabla h(g(w_k))\|_2,$$

needed to reach an accuracy ε is at most

$$k \leq 2\rho_h \ln\left(\frac{\delta_0}{\varepsilon}\right) + 4\theta_g \left(\sqrt{\delta_0} - \sqrt{\varepsilon}\right) + 2\alpha \ln\left(\frac{\theta_g\sqrt{\delta_0} + \rho_g}{\theta_g\sqrt{\varepsilon} + \rho_g}\right),$$

where $\rho_h = L_h/\mu_h$, $\rho_g = \ell_g/\sigma_g$, $\theta_g = L_g/(\sigma_g^2\sqrt{\mu_h})$, $\theta_h = M_h/(2\mu_h^{3/2})$, $\alpha = 4\rho_g^2(2\rho_g^2\theta_h/(3\theta_g) + \rho_h)$.

Proof Let $w \in \mathbb{R}^p$ and $v = \mathrm{RGGN}_{\nu(w)}(f)(w)$ for

$$\nu(w) = \left(1 + \frac{\alpha}{2(1 + \theta_g \|\nabla h(g(w))\|_2/(\sqrt{\mu_h}\rho_g))}\right) L_g \|\nabla h(g(w))\|_2.$$

As shown in Lemma 10, the chosen regularization ensures the sufficient decrease (9). As in Eq. (12), in the proof of Theorem 2, we get that

$$f(w+v) - f(w) \leq -\frac{1}{2}\frac{\sigma_g^2}{\sigma_g^2 L_h + \nu(w)} \|\nabla h(g(w))\|_2^2 = -\frac{b_1 x^3 + b_2 x^2}{b_3 x^2 + b_4 x + 1},$$

where $x = \|\nabla h(g(w))\|_2$, $b_1 = L_g/(2\ell_g\mu_h L_h \sigma_g)$, $b_2 = 1/(2L_h)$, $b_3 = L_g^2/(\sigma_g^3 \ell_g \mu_h L_h)$, $b_4 = L_g/(\sigma_g \ell_g \mu_h) + L_g/(\sigma_g^2 L_h) + 2a_0/(\sigma_g^3 \mu_h L_h)$. The function $f_1(x) = (b_1x^3 + b_2x^2)/(b_3x^2 + b_4x + 1)$ is increasing and since h is strongly convex, we have that $\|\nabla h(g(w))\|_2^2 \geq \mu_h(h(g(w)) - h^*) = \mu_h\delta$ for $\delta = f(w) - f^*$. Hence, as in the proof of Theorem 2, we get that the total number of iterations to reach an accuracy ε is at most $k \leq f_2(\delta_0) - f_2(\varepsilon)$ where

$$f_2'(\delta) = \frac{1}{f_1(\sqrt{\mu_h\delta})} = \frac{1 + c_1\delta^{1/2} + c_2\delta}{c_3\delta + c_4\delta^{3/2}},$$

where $c_1 = \theta_g(\rho_g^{-1} + 2\rho_g + \rho_h^{-1}) + 4\rho_g^3\theta_h/(3\rho_h)$, $c_2 = \theta_g^2/(\rho_g\rho_h)$, $c_3 = 1/(2\rho_h)$, $c_4 = \theta_g/(2\rho_g\rho_h)$. By standard integration, we have that an antiderivative of f_2' is

$$\begin{aligned} f_2(x) &= \frac{\ln(\delta)}{c_3} + \frac{2c_2}{c_4}\sqrt{\delta} - 2\frac{(c_2c_3^2 - c_4c_1c_3 + c_4^2)}{c_3c_4^2} \ln(c_4\sqrt{\delta} + c_3) \\ &= 2\rho_h \ln(\delta) + 4\theta_g\sqrt{\delta} + 8\rho_g^2(\rho_h + 2\rho_g^2\theta_h/(3\theta_g)) \ln(\theta_g\sqrt{\delta}/(2\rho_h\rho_g) + 1/(2\rho_h)). \end{aligned}$$

The result follows. ■

C.2. Local Convergence Analysis Lemmas

Lemma 12 provides a bound on the oracle returned by an RGGN algorithm in terms of the constants introduced in Assumption 4.

Lemma 12 *Given Assumption 4 on problem (1), we have for any $w \in \mathbb{R}^p$, $\nu \geq 0$,*

$$\| \operatorname{RGGN}_\nu(f)(w)\|_2 \leq \frac{\ell}{\ell\sigma + \nu} \|\nabla h(g(w))\|_{g(w)}^*.$$

Proof For $w \in \mathbb{R}^p$, $\nu \geq 0$, denoting $\nabla^2 h(g(w)) = H$, $\nabla g(w) = G$, we have

$$\operatorname{RGGN}_\nu(f)(w) = -GH^{1/2}(H^{1/2}G^\top GH^{1/2} + \nu \operatorname{I})^{-1} H^{-1/2} \nabla h(g(w)).$$

Recall that by definition of σ and ℓ, we have $\sigma \leq \sigma_{\min}(GH^{1/2}), \sigma_{\max}(GH^{1/2}) \leq \ell$. By considering the singular value decomposition of $GH^{1/2}$, we then have

$$\|GH^{1/2}(H^{1/2}G^\top GH^{1/2} + \nu \operatorname{I})^{-1}\|_2 \leq \max_{x \in [\sigma, \ell]} \frac{x}{\nu + x^2} = \begin{cases} \frac{\sigma}{\sigma^2+\nu} & \text{if } \nu \leq \sigma^2 \\ \frac{1}{2\sqrt{\nu}} & \text{if } \sigma^2 \leq \nu \leq \ell^2 \\ \frac{\ell}{\ell^2+\nu} & \text{if } \nu \geq \ell^2 \end{cases}.$$

By analyzing each case, we get the claimed inequality. ■

Lemma 13 provides a bound on the differences of gradients of a self-concordant function. It replaces the classical bound we can have for Lipschitz-continuous gradients.

Lemma 13 *For a ϑ_h-self-concordant strictly convex function h [18, Definition 5.1.1] and y, x such that $\|y - x\|_x < 1/\vartheta_h$, we have,*

$$\|\nabla h(y) - \nabla h(x)\|_x^* \leq \frac{1}{1 - \vartheta_h \|y - x\|_x} \|y - x\|_x.$$

Proof Denote $J = \int_0^1 \nabla^2 h(x + t(y - x))dt$ and $H = \nabla^2 h(x)$, we have $\|\nabla h(y) - \nabla h(x)\|_x^* = \|J(y - x)\|_x^* = \|H^{-1/2} J H^{-1/2}\|_2 \|y - x\|_x$. Now $H^{-1/2} J H^{-1/2} \succeq 0$ since h is strictly convex and by [18, Corollary 5.1.5], we have $J \preceq \nabla^2 h(x)/(1 - \vartheta_h \|y - x\|_x)$ hence $\|H^{-1/2} J H^{-1/2}\|_2 \leq 1/(1 - \vartheta_h \|y - x\|_x)$. ■

Appendix D. Numerical Illustrations

D.1. Convergence comparisons

The empirical performance of the RGGN algorithm compared to a simple gradient descent on simple nonlinear control problems as in (8) in Fig. 1. The first problem considered in Fig. 1 consists in swinging up a pendulum to a vertical position in finite time, the second problem consists in controlling a simple model of a car to be at predefined positions at given times. The detailed experimental setting is presented in the next paragraph. Most importantly, the costs consists in quadratic state costs bounded below by 0, i.e., of the form $h_t(x_t) = (x_t - \hat{x}_t)^\top Q_t(x_t - \hat{x}_t)$ for Q_t positive semi definite and $\hat{x}_\tau$ a reference state with no costs on the control variables. In Fig. 1, we plot $\log(c_t/c_0)$

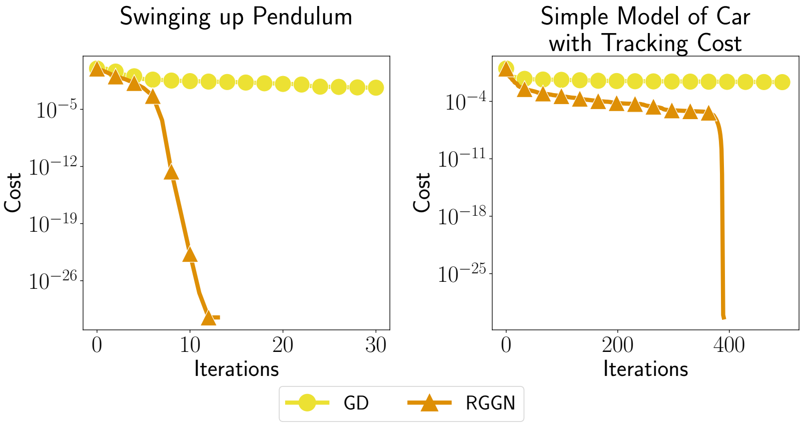

Figure 1: Convergence of Regularized Generalized Gauss-Newton (RGGN) and Gradient Descent (GD) on some nonlinear control problems with optimal cost 0.

where $c_t \geq 0$ denotes the total cost at iteration t computed by means of a gradient descent or a RGGN algorithm and c_0 denotes an initial cost given by initializing the control variables at 0. We observe that the RGGN algorithms converge to an optimal cost, i.e., $c_t = 0$. Moreover the RGGN agorithm outperforms a simple gradient descent and appears to exhibit a fast convergence after some iterations.

D.2. Experimental details

For both experiments, we implemented both algorithms, gradient descent (GD) and RGGN, with a line-search on either the stepsize for GD or the scaled regularization for RGGN. For both settings the control variables are initialized at 0.

Swinging up pendulum. The dynamics of the pendulum and the cost associated to swing up the pendulum are taken from [24, Section 10.2.1] with the constants detailed in [24, Appendix A] except that for Fig. 1 we consider no regularization on the control variables, i.e., $\rho = 0$ in the notations of [24, Section 10.2.1]. We considered an Euler Discretization of the dynamics [24, Section 10.1], an horizon $\tau = 100$ and a discretization step $\Delta = T/\tau = 2./100$ for T the continuous time horizon.

Simple Model of a Car with Tracking Costs. The dynamics of a simple model of a car are detailed in [24, Section 10.3.1, Equation (52)] with the constants detailed in [24, Appendix A]. In Fig. 1, we considered a tracking cost [24, Section 10.3.2. Equation (54)] on a simple track presented in [24, Figure 13] and we did not add any cost on the control variables, i.e., $\lambda = 0$ in the notations of [24, Section 10.3.2]. We considered a Runge-Kutta discretization method with varying control inputs [24, Section 10.1], an horizon $\tau = 50$ and a discretization step $\Delta = T/\tau = 2./50$ for T the continuous time horizon.