Journal of Optimization Theory and Applications (2023) 199:805–830
https://doi.org/10.1007/s10957-023-02270-9

A Fast and Simple Modification of Newton's Method Avoiding Saddle Points

Tuyen Trung Truong[1] · Tat Dat To[2,3] · Hang-Tuan Nguyen[4] · Thu Hang Nguyen[5] · Hoang Phuong Nguyen[5] · Maged Helmy[1,6]

Received: 18 January 2023 / Accepted: 27 June 2023 / Published online: 18 July 2023

Abstract
We propose in this paper New Q-Newton's method. The update rule is conceptually very simple, using the projections to the vector subspaces generated by eigenvectors of positive (correspondingly negative) eigenvalues of the Hessian. The main result of this paper roughly says that if a sequence $\{x_n\}$ constructed by the method from a random initial point x_0 **converges**, then the limit point is a critical point and not a saddle point, and the convergence rate is the same as that of Newton's method. A subsequent work has recently been successful incorporating Backtracking line search to New Q-Newton's method, thus resolving the global convergence issue observed for some (non-smooth) functions. An application to quickly find zeros of a univariate meromorphic function is discussed, accompanied with an illustration on basins of attraction.

Keywords Backtracking line search · Newton-type method · Rate of convergence · Roots of univariate meromorphic functions · Saddle points

Mathematics Subject Classification 37N40 · 49M15 · 49M37 · 65Exx · 65Hxx · 65K05 · 90C26

1 Introduction

In this paper, we consider the unconstrained optimization problem of finding $\min_{x\in\mathbf{R}^m} f(x)$ for a function $f : \mathbf{R}^m \to \mathbf{R}$. This includes, as a special case, the question of finding solutions to systems of equations. Smale's famous list of problems

Communicated by Ebrahim Sarabi.

✉ Tuyen Trung Truong
tuyentt@math.uio.no

Extended author information available on the last page of the article

[23] mentions efficient algorithms to solve systems of polynomial equations as important for twenty-first century mathematics. Throughout the paper, we assume only that f is a C^2 (for to use the algorithm) or C^3 function (to prove some theoretical results), and do not assume restrictive conditions such as the gradient or the Hessian of the function f must be Lipschitz continuous.

In general, finding global minima of a function is NP-hard. Hence, finding a (good) local minimum is the most one can aim for. Moreover, in reality one cannot hope to find closed-form solutions. Therefore, a common strategy is to design an iterative method, which from an initial point x_0 constructs a sequence $\{x_n\}$. To be useful, such a method should possess the following important requirements (for many interesting and useful cost functions):

Requirement 1: Any cluster point of $\{x_n\}$ is a critical point of f.

Requirement 2: If $\{x_n\}$ has a bounded subsequence, then $\{x_n\}$ itself converges.

Requirement 3: If x_0 is randomly chosen and $\{x_n\}$ converges, then the limit point is not a saddle point.

Requirement 4: The method is easy to implement and works well and stably with respect to its parameters.

For first-order methods, there are several algorithms satisfying all of these 4 requirements and work well even in large scale problems like in deep neural networks. A common theme of these methods is that they incorporate line search, in particular Armijo's [4]. We refer the readers to [27–29] and references therein for recent progress and a historical overview.

This paper concerns variants of Newton's method, which currently has no version satisfying all Requirements 1–4, even though it has a fast rate of convergence (when it does converge) and is easy to implement.

Main contributions of this paper We define a new variant of Newton's method, named New Q-Newton's method, which for a cost function on $\mathbf{R}^m$ depends on $m+1$ random parameters fixed from the beginning. The method is both conceptually simple and easy to implement. It modifies the Hessian of the cost function by a term depending on the size of the gradient of the function and the above mentioned $m+1$ parameters. It then uses the **absolute values** of the new matrix's eigenvalues, and not the eigenvalues of the new matrix like in Newton's method. Roughly speaking, this helps to stay away from negative eigenvalues of the Hessian and hence is good for the purpose of avoiding saddle points. At the same time, it behaves like Newton's method near critical points where the Hessian matrix is positive definite. Additionally, it satisfies Requirements 3 and 4 (see Theorem 3.1). The new method's advantages are illustrated in Theorem 3.3—the best result so far for finding roots of meromorphic functions in 1 complex variable.

Related works There are many variants of Newton's method in the literature, which makes it impossible to review all of them. Because of limited space, we are only able to pinpoint a few relevant representative classes.

There are algorithms which are computationally not too expensive, like BFGS and many quasi-Newton's methods, but for which no good theoretical guarantees for general non-convex cost functions are known.

There are algorithms which add a correction into the matrix used in the standard Newton's method, so that the final matrix is positive definite. For a direct application of Newton's method to a system of equations $F = 0$, one well-known method is that of the Levenberg–Marquardt algorithm [3, 9, 17, 18, 33], which adds a positive multiple of the identity matrix to the main matrix. Another method, that of Regularized Newton's method [20, 30, 31], adds a positive multiple of the identity into the Hessian of the associated cost function $f = \frac{1}{2}||F||^2$, where the multiple constant is bigger than the absolute value of the smallest negative eigenvalue of the Hessian matrix. These methods have a good rate of convergence near the solutions of the system of equation. There are also Backtracking versions of them [2], which—under some restrictions (e.g. requiring that the function has compact sublevels and its gradient is Lipschitz continuous)—have good global convergence (satisfying Requirement 1, Requirement 4, as well as Requirement 2 under some restrictions). On the other hand, there is no theoretical proof yet concerning whether these methods (and their Backtracking versions) can avoid saddle points. Experiments in [25] seem to show that, in general, these methods do not have global convergence guarantee or cannot avoid saddle points. These two methods are relevant to New Q-Newton's method and hence will be discussed more in Sect. 2.

There are algorithms which mix first- and second-order methods. One method, which is direction of negative curvature [10, 11], adds into the gradient a negative multiple of an eigenvector corresponding to a negative eigenvalue of the Hessian matrix. It behaves like gradient descent near non-degenerate local minima. One more recent method is that of inertia Newton's method [6], which is a discretization of a modification of Newton's flow. While it is originally of order 2, it is reduced to a system of order 1. These methods can avoid saddle points, but global convergence guarantees are only proven under several restrictive assumptions such as the gradient being Lipschitz continuous (see, for example, Sections 3 in [10, 11]). Moreover, the rate of convergence of these methods is slower than that of Newton's method, and more like first-order methods. For comparison, the Backtracking version of gradient descent is a variant of gradient descent which works very well and stable under wider general settings (including deep neural networks), both theoretically and practically.

There are methods which replace calculating the Hessian matrix with a trust region procedure at each step. One well-known representative is (adaptive) cubic regularization [7, 19]. A line search is integrated into adaptive cubic regularization [5], which under some assumptions—may be difficult to check beforehand, like requiring that the sequence $\|\nabla^2 f(x_n)\|$ is uniformly bounded and $\{f(x_n)\}$ and $\{\nabla f(x_n)\}$ are uniformly continuous—proving only that $\{\nabla f(x_n)\}$ converges and not the convergence of $\{x_n\}$ itself. Moreover, these methods require solving optimization subproblems in each iterative step, which makes satisfying Requirement 4 extremely difficult (see a discussion about this in [13], and experiments in this paper and in [25]).

In a subsequent work [25], Armijo's Backtracking line search [4] is incorporated into New Q-Newton's method. This is based on the observation that the final matrix used in New Q-Newton's method is positive definite. The resulting algorithm is named Backtracking New Q-Newton's method, which satisfies—besides Requirements 3 and 4—also Requirements 1 and 2. There, it is found also a more comprehensive analysis on variants of Newton's method.

Organization of the paper For the readers' convenience, in Sect. 2 we provide a concise review of variants of Newton's method related to New Q-Newton's method. The definition of New Q-Newton's method and its main theoretical properties are presented in Sect. 3, where some pictures on basins of attraction are given. After that, some conclusions are presented. The appendices present details of some technical proofs, implementation details, experimental settings, as well as some further experimental results. (Many more can be found in the arXiv version of this paper and in the paper [25].)

2 A Brief Review on Relevant Variants of Newton's Method

Let $f : \mathbf{R}^m \to \mathbf{R}$ be a C^2 function. We recall some common notations: ∇f is the gradient of f, and $\nabla^2 f$ is the Hessian of f. A point x_0 is a critical point of f if $\nabla f(x_0) = 0$. A critical point x_0 of f is non-degenerate if the Hessian $\nabla^2 f(x_0)$ is invertible. A critical point x_0 of f is a saddle point if x_0 is non-degenerate and $\nabla^2 f(x_0)$ has at least one negative eigenvalue. (Note that this definition is more general than the usual one, in that it includes also local maxima.) A function f is Morse if all of its critical points are non-degenerate.

In Newton's method, from $x_0 \in \mathbf{R}^m$ one defines subsequently:

$$x_{n+1} = x_n - [\nabla^2 f(x_n)]^{-1}.\nabla f(x_n). \tag{1}$$

Regularized Newton's method adds $c_1 \max\{0, -\lambda_{\min}(\nabla^2 f(x_n))\}Id$ into $\nabla^2 f(x_n)$, where $c_1 > 1$, $\lambda_{\min}(.)$ is the smallest eigenvalue of a square matrix, and Id is the identity matrix. If $f = ||F||^2/2$ for a map $F = (F_1, \ldots, F_{m'}) : \mathbf{R}^m \to \mathbf{R}^{m'}$, then another version (historically appearing first) is $x_{n+1} = x_n - [JF(x_n)^T JF(x_n)]^{-1} JF(x_n)^T.F(x_n)$, where JF is the Jacobian matrix of F and $(.)^T$ is the transpose of a matrix. Levenberg–Marquardt algorithm adds $\lambda_n Id$ into the matrix $JF(x_n)^T JF(x_n)$, where usually λ_n is $c_2||\nabla f(x_n)||^\gamma$ for $c_2, \gamma > 0$. The final matrices used in both methods are positive semi-definite.

New Q-Newton's method also adds a term $\lambda_n Id$ to the Hessian $\nabla^2 f(x_n)$, but does not require that the final matrix $\nabla^2 f(x_n)+\lambda_n Id$ is positive definite. Instead, we change the sign of negative eigenvalues of the matrix $\nabla^2 f(x_n) + \lambda_n Id$. We also simplify the choice of λ_n by letting $\lambda_n = \delta_j||\nabla f(x_n)||^\gamma$ for δ_j in a fixed set of $m+1$ real numbers. This turns out to provide the algorithm with very strong theoretical guarantees—in particular with its Backtracking version in [25]—while also making it straightforward to implement the algorithm and its variants. A detailed description of this method is in the next section.

3 New Q-Newton's Method

We first describe New Q-Newton's method, then prove some main theoretical properties, and apply to finding roots of meromorphic functions (with some pictures for basins of attraction provided).

3.1 The Algorithm

Here, we introduce New Q-Newton's method. An invertible square matrix A of dimension m with real coefficients is diagonalizable. That is, we can find an orthonormal basis $e_1, \ldots, e_m$ of $\mathbf{R}^m$ and m nonzero real numbers $\lambda_1, \ldots, \lambda_m$ such that $A.e_j = \lambda_j e_j$ for all $j = 1, \ldots, m$. For a vector $v \in \mathbf{R}^m$, we define: $pr_{A,+}(v) = \sum_{i:\ \lambda_i>0} < v, e_i > e_i$ and $pr_{A,-}(v) = \sum_{i:\ \lambda_i<0} < v, e_i > e_i$. If V_+ is the vector space generated by $\{e_i\}_{\lambda_i>0}$ and V_- is the vector space generated by $\{e_i\}_{\lambda_i<0}$, then V_+ and V_- are uniquely defined and are independent of the choice of the vectors $e_1, ..., e_m$. Then, $pr_{A,+}$ is simply the orthogonal projection to V_+, and $pr_{A,-}$ is simply the orthogonal projection to V_-.

Algorithm 1: New Q-Newton's method

Result: Find a critical point of $f : \mathbf{R}^m \to \mathbf{R}$
Given: $\Delta = \{\delta_0, \delta_1, \ldots, \delta_m\}$ (chosen **randomly**) and $\alpha > 0$; Initialization: $x_0 \in \mathbf{R}^m$;
for $n = 0, 1, 2 \ldots$ **do**
  $j = 0$
  if $\|\nabla f(x_n)\| \neq 0$ **then**
    while $\det(\nabla^2 f(x_n) + \delta_j \|\nabla f(x_n)\|^{1+\alpha} Id) = 0$ **do**
      $j = j + 1$
    end
  end
  $A_n := \nabla^2 f(x_n) + \delta_j \|\nabla f(x_n)\|^{1+\alpha} Id$
  $v_n := A_n^{-1} \nabla f(x_n)$
  $w_n := pr_{A_n,+}(v_n) - pr_{A_n,-}(v_n)$
  $x_{n+1} := x_n - w_n$
end

Remark 3.1 The choice $w_n = pr_{A_n,+}(v_n) - pr_{A_n,-}(v_n)$ is to change the sign of the negative eigenvalues of A_n. As the proof of the main results and the experiments show, in Algorithm 1 one does not need to have exact values of the Hessian, its eigenvalues, and eigenvectors for the algorithm to perform well. The randomness of the parameters $\delta_0, \delta_1, \ldots, \delta_m$ is only needed in the proof (see Theorem 3.1) that the algorithm can avoid saddle points. (For the existence of local Stable—Centre manifolds near saddle points, this randomness is not needed.) See "Appendix" for implementation details. See [25] for variants.

A disadvantage of (Backtracking) New Q-Newton's method is that in higher dimensions it is very costly to compute the Hessian matrix and its eigenvalues and eigenvectors. To resolve this issue is beyond the scope of the current paper. We note, however, that a simpler version, which does not use all negative eigenvalues but only the smallest negative eigenvalues, has been tested in [25] to perform similarly to Algorithm 1. This suggests that one may reduce the computational cost by using only a few eigenvalues of the Hessian matrix (which can be efficiently computed, e.g. by using Lanczos algorithm). Another efficient modification is to use two-way Backtracking line search, see [28, 29].

3.2 Rate of Convergence and Avoidance of Saddle Points

The main result we obtain is the following.

Theorem 3.1 *Let $f : \mathbf{R}^m \to \mathbf{R}$ be C^3. Let $\{x_n\}$ be a sequence constructed by New Q-Newton's method. Assume that $\{x_n\}$ converges to x_∞. Then,*

(1) *$\nabla f(x_\infty) = 0$, that is x_∞ is a critical point of f.*
(2) *If $\delta_0, \dots, \delta_m$ are chosen* **randomly**, *then there is a set $\mathcal{A} \subset \mathbf{R}^m$ of Lebesgue measure 0, so that if $x_0 \notin \mathcal{A}$, then x_∞ cannot be a saddle point of f.*
(3) *If $x_0 \notin \mathcal{A}$ (as defined in part 2) and $\nabla^2 f(x_\infty)$ is invertible, then x_∞ is a local minimum and the rate of convergence is quadratic.*
(4) *More generally, if $\nabla^2 f(x_\infty)$ is invertible (but x_0 does not need to be random), then the rate of convergence is at least linear.*
(5) *If x'_∞ is a non-degenerate local minimum of f, then for initial points x'_0 close enough to x'_∞, the constructed sequence $\{x'_n\}$ will converge to x'_∞.*

Next, we state an interesting immediate consequence of the theorem.

Corollary 3.1 *Let f be a C^3 function and Morse. Let x_0 be a random initial point, and let $\{x_n\}$ be a sequence constructed by New Q-Newton's method, where the hyperparameters $\delta_0, \dots, \delta_m$ are randomly chosen. If x_n converges to x_∞, then x_∞ is a local minimum and the rate of convergence is quadratic.*

Proof (Of Theorem 3.1)

(1) Since $\lim_{n\to\infty} x_n = x_\infty$, we have $w_n = x_{n+1} - x_n \to 0$. Moreover, $\nabla^2 f(x_n) \to \nabla^2 f(x_\infty)$. Then, by the definition of A_n, we have that $||A_n||$ is bounded. Note that by construction $||w_n|| = ||v_n||$ for all n, and hence $\lim_{n\to\infty} v_n = 0$. Then, $\nabla f(x_\infty) = \lim_{n\to\infty} \nabla f(x_n) = \lim_{n\to\infty} A_n v_n = 0$.
(2) For simplicity, we can assume that $x_\infty = 0$. We assume that x_∞ is a saddle point and will arrive at a contradiction. By (1) we have $\nabla f(0) = 0$, and by the assumption we have that $\nabla^2 f(0)$ is invertible. We define $A(x) = \nabla^2 f(x) + \delta(x)||\nabla f(x)||^{1+\alpha} Id$, and $A = \nabla^2 f(0) = A(0)$. We look at the following (may not be continuous) relevant dynamical system on $\mathbf{R}^m$: $F(x) = x - w(x)$, where $w(x) = pr_{A(x),+}(v(x)) - pr_{A(x),-}(v(x))$ and $v(x) = A(x)^{-1}\nabla f(x)$. The update rule of New Q-Newton's method is $x_{n+1} = F(x_n)$.

Then for an initial point x_0, the sequence constructed by New Q-Newton's method is exactly the orbit of x_0 under the dynamical system $x \mapsto F(x)$. Hence, $A(x)$ is C^1 near x_∞, say in an open neighbourhood U of x_∞, and at every point $x \in U$, the matrix $A(x)$ must be one of the $m+1$ maps $F_j(x) = \nabla^2 f(x) + \delta_j ||\nabla f(x)||^2 Id$ (for $j = 0, 1, \dots, m$), and therefore $F(x)$ must be one of the corresponding $m+1$ maps $F_j(x)$. Since f is assumed to be C^3, it follows that all of the corresponding $m+1$ maps F_j are locally Lipschitz continuous.

Now, we analyse the map $F(x)$ near the point $x_\infty = 0$. Since $\nabla^2 f(0)$ is invertible, near 0 we have $A(x) = \nabla^2 f(x) + \delta_0 ||\nabla f(x)||^{1+\alpha} Id$. Moreover, the maps $x \mapsto pr_{A(x),+}(A(x)^{-1}\nabla f(x))$ and $x \mapsto pr_{A(x),-}(A(x)^{-1}\nabla f(x))$ are C^1. [This assertion is probably well known to experts, in particular in the field of perturbations of linear

operators. Here, for completion we present a proof, following [15], by using an integral formula for projections on eigenspaces via the theory of resolvents. Let $\lambda_1, \ldots, \lambda_s$ be distinct solutions of the characteristic polynomial of A. By assumption, all λ_j are nonzero. Let $\gamma_j \subset \mathbf{C}$ be a small circle with positive orientation enclosing λ_j and not other λ_r's. Moreover, we can assume that γ_j does not contain 0 on it or inside it, for all $j = 1, \ldots, s$. Since $A(x)$ converges to $A(0)$, we can assume that for all x close to 0, all roots of the characteristic polynomial of $A(x)$ are contained well inside the union $\bigcup_{j=1}^{s} \gamma_j$. Then by the formula (5.22) on page 39, see also Problem 5.9, chapter 1 in [15], we have that $P_j(x) = -\frac{1}{2\pi i}\int_{\gamma_j}(A(x) - \zeta Id)^{-1}d\zeta$ is the projection on the eigenspace of $A(x)$ corresponding to the eigenvalues of $A(x)$ contained inside γ_j. Since $A(x)$ is C^1, it follows that $P_j(x)$ is C^1 in the variable x for all $j = 1, \ldots, s$. Then, by the choice of the circle γ_j, we have that $pr_{A(x),+} = \sum_{j:\,\lambda_j>0} -\frac{1}{2\pi i}\int_{\gamma_j}(A(x) - \zeta Id)^{-1}d\zeta$ is C^1 in the variable x. Similarly, $pr_{A(x),-} = \sum_{j:\,\lambda_j<0} -\frac{1}{2\pi i}\int_{\gamma_j}(A(x) - \zeta Id)^{-1}d\zeta$ is also C^1 in the variable x. Since $A(x)$ is C^1 in x and $f(x))$ is C^2, the proof of the claim is completed.]

Hence, since $x \mapsto (\nabla^2 f(x) + \delta_0||\nabla f(x)||^{1+\alpha} Id)^{-1}\nabla f(x)$ is C^1, it follows that the map $x \mapsto F(x)$ is C^1. We now compute the Jacobian of $F(x)$ at the point 0. Since $\nabla f(0) = 0$, it follows that $\nabla f(x) = \nabla^2 f(0).x + o(||x||)$; here, we use the small-o notation, and hence $(\nabla^2 f(x) + \delta_0||\nabla f(x)||^{1+\alpha} Id)^{-1}\nabla f(x) = x + o(||x||)$. It follows that $w(x) = pr_{A,+}(x) - pr_{A,-}(x) + o(||x||)$, which in turn implies that $F(x) = 2pr_{A,-}(x) + o(||x||)$. Hence, $JF(0) = 2pr_{A,-}$.

Therefore, we obtain the existence of local stable-central manifolds for the associated dynamical systems near saddle points of f (see Theorems III.6 and III.7 in [21]). We can then using the fact that under the assumptions the hyperparameters $\delta_0, \ldots, \delta_m$ are randomly chosen, to obtain:

Claim: $F(x)$ is—outside a set $\mathcal{E}$ of Lebesgue measure 0—locally invertible.

The relation between Claim and avoidance of saddle points is as follows. Let $\mathcal{A}_{loc}$ be the union of local stable-central manifolds around all saddle points. Then by the above arguments, we know that $\mathcal{A}_{loc}$ has Lebesgue measure 0. If x_0 is an initial point such that the constructed sequence $x_n = F^{\circ n}(x_0)$ converges to a saddle point, then there is some n_0 such that $F^{n_0}(x_0) \in \mathcal{A}_{loc}$. Choose $0 \leq m \leq n_0$ be the smallest number such that $F^{\circ m}(x_0) \in \mathcal{A}_{loc} \cup \mathcal{E}$. Then by Claim, x_0 is in the inverse image of a set of Lebesgue measure zero by a locally invertible map and hence belongs to a set of Lebesgue measure zero. Taking the union on all n_0 and m, we find the set $\mathcal{A}$ which has Lebesgue measure zero which contains x_0.

A similar claim has been established for another dynamical system in [27]—for a version of Backtracking gradient descent. The idea in [27] is to show that the associated dynamical system (depending on ∇f), which is locally Lipschitz continuous, has locally bounded torsion. The case at hand, where the dynamical system depends on the Hessian and also orthogonal projections to the eigenspaces of the Hessian, is more complicated to deal with.

We note that the fact that $\delta_0, \ldots, \delta_m$ should be random to achieve the truth of Claim has been overlooked in the arXiv version of this paper and has now been corrected in

a new work by the first author [26], under more general settings. Here is a sketch of how to prove Claim, see [26] for details.

Putting, as above, $A(x,\delta) = \nabla^2 f(x) + \delta||\nabla f(x)||^{1+\alpha} Id$. Let $\mathcal{C} = \{x \in \mathbf{R}^m : \nabla f(x) = 0\}$ be the set of critical points of f. Since $\det(A(x,\delta))$ is a polynomial and is nonzero for $x \notin \mathcal{C}$, there is a set $\Delta \subset \mathbf{R}$ of Lebesgue measure 0 so that for a given $\delta \notin \Delta$, the set $x \notin \mathcal{C}$ for which $A(x,\delta)$ is not invertible has Lebesgue measure 0. One then shows, using that $w(x,\delta)$ (that is, the $w(x)$ as above, but now we add the parameter δ in to make clear the dependence on δ), is a rational function in δ, and is nonzero (by investigating what happens when $\delta \to \infty$). This allows one to show that there is a set $\Delta' \subset \mathbf{R}\backslash\Delta$ of Lebesgue measure 0 so that for all $\delta \notin (\Delta \cup \Delta')$ the matrix $A(x,\delta)$ is invertible and the set where the **gradient** of the dynamical system $F(x) = x - w(x,\delta)$ is, locally outside $\mathcal{C}$, invertible. This proves Claim.

That $\delta_0, \ldots, \delta_m$ are random means that they should avoid the set $\Delta \cup \Delta'$.

(3) We can assume that $x_\infty = 0$ and define $A = \nabla^2 f(0)$. Part 1) and the assumption that $\nabla^2 f(0)$ is invertible imply that we can assume, without loss of generality, that $A_n = \nabla^2 f(x_n) + \delta_0||\nabla f(x_n)||^{1+\alpha} Id$ for all n, and that $\nabla^2 f(x_n)$ is invertible for all n. Since $\nabla f(0) = 0$ and f is C^3, we obtain by Taylor's expansion $\nabla f(x_n) = A.x_n + O(||x_n||^2)$. By Taylor's expansion, we find that

$$\begin{aligned} A_n^{-1} &= \nabla^2 f(x_n)^{-1}.(Id + \delta_0||\nabla f(x_n)||^{1+\alpha}\nabla^2 f(x_n))^{-1} \\ &= \nabla^2 f(x_n)^{-1}(Id - \delta_0||\nabla f(x_n)||^{1+\alpha}\nabla^2 f(x_n) \\ &\quad +(\delta_0||\nabla f(x_n)||^{1+\alpha}\nabla^2 f(x_n))^2 + \ldots) \\ &= \nabla^2 f(x_n)^{-1} + O(||x_n||^{1+\alpha}) = A^{-1} + O(||x_n||). \end{aligned}$$

Multiplying A_n^{-1} to both sides of the equation $\nabla f(x_n) = \nabla^2 f(0).x_n + O(||x_n||^2)$, using the above approximation for A_n^{-1}, we find that

$$v_n = A_n^{-1}\nabla f(x_n) = x_n + O(||x_n||^2).$$

Since we assume that $x_0 \notin \mathcal{A}$, it follows that A is positive definite. Hence, we can assume, without loss of generality, that A_n is positive definite for all n. Then from the construction, we have that $w_n = v_n$ for all n. Hence, we obtain $x_{n+1} = x_n - w_n = x_n - v_n = O(||x_n||^2)$ (quadratic convergence rate).

(4) The proof of part 3 shows that in general we still have $v_n = x_n + O(||x_n||^2)$. Therefore, we have $w_n = pr_{A_n,+}(v_n) - pr_{A_n,-}(v_n) = O(||x_n||)$. Hence, $x_{n+1} = x_n - w_n = O(||x_n||)$. (Convergence rate is at least linear.)

(5) This assertion follows immediately from the proof of part 3. □

3.3 Finding Roots of Meromorphic Functions in 1 Complex Variable

Here, we give an application of the new algorithm to quickly finding roots of meromorphic functions in 1 complex variable. As far as we know, the result in Theorem 3.3 is new and strongest among all existing iterative algorithms in contemporary literature. To illustrate the advantage of the new algorithm, we present some pictures for basins

of attraction taken from [25]. Because of the space limit, many lengthy proofs will be put in "Appendix A".

Concerning the direct use of Newton's method for solving systems of equations, even for polynomials $p(z)$ of 1 complex variable z of small degrees (e.g. 4), there is the well-known phenomenon of attracting cycles of at least 2 points. (Hence, as a consequence, Newton's method does not converge to a root of $p(z)$.) Among all previous variants of Newton's methods, we are aware of only one method which has theoretical guarantee for convergence to roots [24]. This is the Random damping Newton's method, which has the update rule $z_{n+1} = z_n - \gamma_n p'(z_n)/p(z_n)$, where γ_n is randomly chosen. However, this method has some disadvantages. First, the theoretical proof is very complicated. Second, it is not guaranteed when applied to a meromorphic function like Theorem 3.3, or to higher dimensions. Indeed, our extensive experiments seem to confirm that this method does not perform well in the more general setting, and in many instances it behaves like the original Newton's method.

Let g be a meromorphic function in 1 complex variable $z \in \mathbf{C}$. Then, outside a discrete set (poles of g), g is a usual holomorphic function. To avoid the trivial case, we can assume that g is non-constant. We write $z = x + iy$, where $x, y \in \mathbf{R}$. We define $u(x, y) =$ the real part of g, and $v(x, y) =$ the imaginary part of g. Then, we consider a function $f(x, y) = u(x, y)^2 + v(x, y)^2$. A zero $z = x + iy$ of g is a global minimum of f, at which the function value is 0. Therefore, optimization algorithm can be used to find roots of g, by applying to the function $f(x, y)$, provided the algorithm assures convergence to critical points and avoidance of saddle points, and provided that critical points of f which are not zeros of g must be saddle points of f.

Theorem 3.2 *Let $f(x, y)$ be the function constructed from a non-constant meromorphic function $g(z)$ as before. Assume that the constant $\alpha > 0$ in the definition of New Q-Newton's method does not belong to the set $\{(n-3)/(n-1) : n = 2, 3, 4, \ldots\}$ (e.g. we can choose $\alpha = 1$). Let (x_n, y_n) be a sequence constructed by Backtracking New Q-Newton's method from an arbitrary initial point which is not a pole of f. Then, either $\lim_{n\to\infty}(x_n^2 + y_n^2) = \infty$, or the sequence $\{(x_n, y_n)\}$ converges to a point (x^*, y^*) which is a critical point of f.*

Theorem 3.2 provides the needed convergence to critical points. Its proof will be given at the end of this subsection, after some preparations.

To prove avoidance of saddle points, we need to first classify critical points of the function f. This is done in Lemma A.1. For a generic meromorphic function g, the functions g' and gg'' have no common roots. Hence, by Lemma A.1 and Theorem 3.2 (more generally, Theorem A.1), we obtain:

Theorem 3.3 *Let g be a generic meromorphic function in 1 complex variable, and let $f(x, y)$ be the function in 2 real variables constructed from g as above. Let (x_n, y_n) be the sequence constructed by applying Backtracking New Q-Newton's method to f from a random initial point (x_0, y_0). Then either*

(i) $\lim_{n\to\infty}(x_n^2 + y_n^2) = \infty$,
or

(ii) *(x_n, y_n) converges to a point (x_∞, y_∞) so that $z_\infty = x_\infty + iy_\infty$ is a root of g, and the rate of convergence is quadratic.*

Moreover, if g is a polynomial, then f has compact sublevels, and hence, only case (ii) happens.

If h is a non-constant meromorphic function, then $g = h/h'$ has only simple zeros (which are either zeros or poles of h). Hence, they will be non-degenerate global minima of f. If h is a polynomial, then $g = h/h'$ has compact sublevels.

Now, we are ready to prove Theorem 3.2.

Proof *(Of Theorem 3.2)* Let Ω be the complement of the set of poles of f. Then as mentioned, f is real analytic on Ω. Let $z_n = (x_n, y_n)$ be a sequence constructed by Backtracking New Q-Newton's method in [25]. Then, since the sequence of function values $\{f(z_n)\}$ decreases and the value of f is infinity only at the poles of f, if the initial point is in Ω, then the whole sequence stays in Ω.

We know by [25] that any cluster point of $\{z_n\}$ is a critical point of f. Hence, it remains to show that $\{z_n\}$ converges. To this end, by the arguments in [1], it suffices to show that for every point $(x^*, y^*) \in \Omega$, if the point $z_n = (x_n, y_n)$ is in a small open neighbourhood of (x^*, y^*), then there is a constant $C > 0$ (depending on that neighbourhood but independent of the point z_n) so that

$$f(z_n) - f(z_{n+1}) \geq C||z_{n+1} - z_n|| \times ||\nabla f(z_n)||. \tag{2}$$

Lemma A.2 in "Appendix A", whose proof is lengthy, then completes the proof of the theorem. □

We finish this section with some pictures for basins of attraction in finding roots of a polynomial of degree 4: $P_4(z) = (z^2 + 1)(z - 2.3)(z + 2.3)$, which has 4 roots: $z_1^* = 2.3$, $z_2^* = -2.3$, $z_3^* = i$ and $z_4^* = -i$. The basins of attraction for Newton's method are then fractal. Moreover, there are sets of positive Lebesgue measure where Newton's method applied to an initial point in these sets will not converge to any of the roots. Basins of attraction for Backtracking gradient descent seem to be more regular than that for Newton's method, but less regular than that for Backtracking New Q-Newton's method. Figure 1 is created by choosing the initial point z_0 in a lattice $v + (0.1j, 0.1k)$, for $j, k \in [-30, 30]$, and where v is a randomly chosen point in $[-1, 1] \times [-1.1]$.

4 Conclusions

This paper presented New Q-Newton's method, a new variant of Newton's method which is conceptually simple, easy to implement and can avoid saddle points while having a fast convergence rate (when it converges). New Q-Newton's method has been combined with Backtracking line search, by the first author, to obtain an iterative optimization method which also has the needed convergence guarantee. As an application, we obtain a new result on finding roots of meromorphic functions in 1 complex variable. Some experimental results, reported in "Appendix B", show that

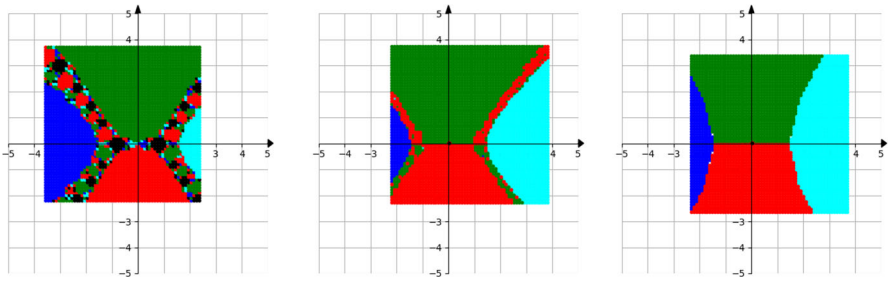

Fig. 1 Basins of attraction for the polynomial $P_4(z) = (z^2 + 1)(z - 2.3)(z + 2.3)$. The left image is for Newton's method, the middle image is for gradient descent method with Backtracking line search, and the right image is for Backtracking New Q-Newton's method. Blue: initial points z_0 for which the constructed sequence converges to z_1^*. Cyan: similar for the root z_2^*. Green: similar for the root z_3^*. Red: similar for the root z_4^*. Black: other points

the new algorithm works very well against well-known existing variants of Newton's method.

Acknowledgements The first author is supported by Young Research Talents Grant Number 300814 from the Research Council of Norway. The research of this paper was also facilitated by a travel Grant from the Trond Mohn Foundation for the first author to visit Torus Actions SAS. We thank Claire McLaughlin for checking the manuscript, and thank anonymous referees for constructive feedbacks.

Funding Open access funding provided by University of Oslo (incl Oslo University Hospital).

Data Availability Statement Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

A Proofs of some Results

In this "Appendix", we collect some needed results from [25] and proofs of some technical results for the readers' convenience.

Theorem A.1 *Let $f : \mathbf{R}^m \to \mathbf{R}$ be a C^3 function. Let x_0 be an initial point and $\{x_n\}$ the sequence constructed by Backtracking New Q-Newton's method*

(1) *$f(x_{n+1}) \leq f(x_n)$ for all n. Moreover, any cluster point of $\{x_n\}$ is a critical point of f.*
(2) *Assume moreover that f is Morse (that is, all its critical points are non-degenerate) and x_0 is randomly chosen. Then, we have two alternatives:*

(i) $\lim_{n\to\infty} ||x_n|| = \infty$,
or
(ii) $\{x_n\}$ *converges to a local minimum of* f*, and the rate of convergence is quadratic.*
Moreover, if f *has compact sublevels, then only case (ii) happens.*

We now discuss properties of critical points of $f(x, y) = u(x, y)^2 + v(x, y)^2$, outside poles of the meromorphic function $g(z) = u(z) + iv(z)$, where $z = x + iy$. Recall that by Cauchy–Riemann's equations, we have

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y},\ \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}.$$

Lemma A.1 *Let* $f(x, y) : \mathbf{R}^2 \to \mathbf{R}$ *be associated with a meromorphic function* $g(z)$ *as above.*

(1) *A point* (x^*, y^*) *is a critical point of* $f(x, y)$*, iff* $z^* = x^* + iy^*$ *is a zero of* $g(z)g'(z)$.
(2) *If* $z^* = x^* + iy^*$ *is a zero of* g*, then* (x^*, y^*) *is an isolated global minimum of* f. *Moreover, if* z^* *is not a root of* g'*, then* (x^*, y^*) *is a non-degenerate critical point of* f.
(3) *If* $z^* = x^* + iy^*$ *is a zero of* g'*, but not a zero of* gg", *then* (x^*, y^*) *is a saddle point of* f.

Proof We will write u_x for $\partial u/\partial x$, u_{xy} for $\partial^2 u/\partial x \partial y$ and so on.

(1) By calculation, we have $\nabla f = (2uu_x + 2vv_x, 2uu_y + 2vv_y)$. By Cauchy–Riemann's equations, a critical point (x^*, y^*) of f satisfies a system of equations

$$\begin{aligned} uu_x - vu_y &= 0, \\ uu_y + vu_x &= 0, \end{aligned}$$

Consider the above as a system of linear equations in variables u_x, u_y, we see that if (x^*, y^*) is not a root of g, then it must be a root of u_x, u_y. In the latter case, by Cauchy–Riemann's equations, (x^*, y^*) is also a root of v_x, v_y, and hence, $z^* = x^* + iy^*$ is a root of $g'(z)$.

(2) Since $f \geq 0$, and $f(x^*, y^*) = 0$ iff $z^* = x^* + iy^*$ is a root of g, such an (x^*, y^*) is a global minimum of f. Moreover, since the zero set of g is discrete, (x^*, y^*) is an isolated global minimum.

For the remaining claim, we need to show that if z^* is not a root of g', then $\nabla^2 f(x^*, y^*)$ is invertible. By calculation, the Hessian of f at a general point is 2 times of:

$$\begin{pmatrix} u_x^2 + v_x^2 + uu_{xx} + vv_{xx} & u_x u_y + v_x v_y + uu_{xy} + vv_{xy} \\ u_x u_y + v_x v_y + uu_{xy} + vv_{xy} & u_y^2 + v_y^2 + uu_{yy} + vv_{yy} \end{pmatrix}.$$

At (x^*, y^*), we have $u = v = 0$, and hence, by Cauchy–Riemann's equations the above matrix becomes:

$$\begin{pmatrix} u_x^2 + u_y^2 & 0 \\ 0 & u_x^2 + u_y^2 \end{pmatrix}$$

which is positive definite if z^* is not a root of g', as wanted.

(3) Since here (x^*, y^*) is a solution of $u_x = u_y = v_x = v_y = 0$, the Hessian of f at (x^*, y^*) is 2 times of:

$$\begin{pmatrix} uu_{xx} + vv_{xx} & uu_{xy} + vv_{xy} \\ uu_{xy} + vv_{xy} & uu_{yy} + vv_{yy} \end{pmatrix}.$$

Note that by Cauchy–Riemann's equations we have $u_{xx} + u_{yy} = 0$ and $v_{xx} + v_{yy} = 0$. Therefore, if we put $a = uu_{xx} + vv_{xx}$ and $b = uu_{xy} + vv_{xy}$, then the above matrix becomes:

$$\begin{pmatrix} a & b \\ b & -a \end{pmatrix}.$$

Since the determinant is $-a^2 - b^2$, we conclude that (x^*, y^*) is a saddle point of f, except the case where $a = b = 0$. In the latter case, by Cauchy–Riemann's equations we have $u_{xy} = v_{xx}$ and $v_{xy} = -u_{yy}$, and hence, (x^*, y^*) must be a solution to

$$\begin{aligned} uu_{xx} + vv_{xx} &= 0, \\ vu_{xx} - uv_{xx} &= 0. \end{aligned}$$

By Cauchy–Riemann's equations again, we find that this cannot be the case, except that z^* is a root of $gg'' = 0$. □

Lemma A.2 *Assumptions are as in Theorem 3.2. For every point $(x^*, y^*) \in \Omega$, if the point $z_n = (x_n, y_n)$ is in a small open neighbourhood of (x^*, y^*), then there is a constant $C > 0$ (depending on that neighbourhood but independent of the point z_n) so that*

$$f(z_n) - f(z_{n+1}) \geq C||z_{n+1} - z_n|| \times ||\nabla f(z_n)||. \tag{3}$$

Proof Let us recall that if w_n is the one constructed by New Q-Newton's method, then $z_{n+1} = z_n - \beta_n w_n$, where β_n is chosen from the Backtracking line search so that Armijo's condition

$$f(z_n) - f(z_{n+1}) \geq \frac{1}{2}\beta_n < w_n, \nabla f(z_n) > .$$

is satisfied.

For a 2×2 invertible matrix A, we define $sp(A) = \max\{|\lambda| : \ \lambda \text{ is an eigenvalue of } A\}$, and $minsp(A) = \min\{|\lambda| : \ \lambda \text{ is an eigenvalue of } A\}$. Then by the arguments in [25], we find that

$$\begin{aligned}\beta_n < w_n, \nabla f(z_n) > &\geq \beta_n ||w_n|| \times ||\nabla f(z_n)|| \times minsp(A_n)/sp(A_n) \\ &= ||z_n - z_{n+1}|| \times ||\nabla f(z_n)|| \times minsp(A_n)/sp(A_n),\end{aligned}$$

where $A_n = \nabla^2 f(z_n) + \delta ||\nabla f(z_n)||^{1+\alpha} Id$ is constructed by New Q-Newton's method. Here, recall that δ belongs to a finite set $\{\delta_0, \ldots, \delta_m\}$. Hence, to show that (3) is satisfied, it suffices to show that every point $(x^*, y^*) \in \Omega$ has an open neighbourhood U so that if $z_n \in U$, then $minsp(A_n)/sp(A_n) \geq C$ for some constant $C > 0$ depending only on U.

If (x^*, y^*) is not a critical point of f, then by the construction of Backtracking New Q-Newton's method, $minsp(A_n) \geq ||\nabla f(z_n)||^{1+\alpha}$ is bounded away from 0 in a small neighbourhood U of (x^*, y^*), while $sp(A_n)$ is bounded from above in the same neighbourhood. Hence, $minsp(A_n)/sp(A_n)$ is bounded away from 0 in U as wanted.

Hence, we need to check the wanted property only at the critical points of f. We saw in Lemma A.1 that (x^*, y^*) is a critical point of f iff $z^* = x^* + iy^*$ is a root of gg'. Hence, we will consider two separate cases. To simplify the arguments, we can assume that $z^* = 0$ is the concerned root of gg'.

Case 1: $z^* = 0$ is a zero of g.

We expand in a small neighbourhood of 0: $g(z) = \tau z^N + h.o.t$ (here h.o.t. means terms which converge to 0 quicker than z^N), where $N \geq 1$ is the multiplicity of 0. We first claim that when z is close to z^*, then the two eigenvalues of $\nabla^2 f(z)$ are $\lambda_1(z) \sim (2N^2 - N)|\tau|^2 r^{2N-2}$ and $\lambda_2(z) \sim N|\tau|^2 r^{2N-2}$, where $r = ||z||$. For simplicity, we can assume that $\tau = 1$.

Write $z = re^{i\theta}$. We have, by definition $u + iv = z^N + h.o.t.$, $u_x + iv_x = \frac{d}{dx}(x + iy)^N + h.o.t.$ and so on. Hence,

$$\begin{aligned}u &= r^N \cos(N\theta) + h.o.t., \\ v &= r^N \sin(N\theta) + h.o.t., \\ u_x &= Nr^{N-1} \cos((N-1)\theta) + h.o.t., \\ v_x &= Nr^{N-1} \sin((n-1)\theta) + h.o.t., \\ u_y &= -v_x = -Nr^{N-1} \sin((N-1)\theta) + h.o.t., \\ v_y &= u_x = Nr^{N-1} \cos((N-1)\theta) + h.o.t., \\ u_{xx} &= N(N-1)r^{N-2} \cos((N-2)\theta) + h.o.t., \\ v_{xx} &= N(N-1)r^{N-2} \sin((N-2)\theta) + h.o.t., \\ u_{yy} &= -u_{xx} = -N(N-1)r^{N-2} \cos((N-2)\theta) + h.o.t., \\ v_{yy} &= -v_{xx} = -N(N-1)r^{N-2} \sin((N-2)\theta) + h.o.t., \\ u_{xy} &= v_{yy} = -N(N-1)r^{N-2} \sin((N-2)\theta) + h.o.t., \\ v_{xy} &= u_{xx} = N(N-1)r^{N-2} \cos((N-2)\theta) + h.o.t.\end{aligned}$$

We recall that the Hessian matrix $\nabla^2 f(x, y)$ is:

$$\begin{pmatrix} u_x^2 + v_x^2 + uu_{xx} + vv_{xx} & u_x u_y + v_x v_y + uu_{xy} + vv_{xy} \\ u_x u_y + v_x v_y + uu_{xy} + vv_{xy} & u_y^2 + v_y^2 + uu_{yy} + vv_{yy} \end{pmatrix},$$

which by Cauchy–Riemann's equations becomes:

$$\begin{pmatrix} u_x^2 + v_x^2 + uu_{xx} + vv_{xx} & uu_{xy} + vv_{xy} \\ uu_{xy} + vv_{xy} & u_y^2 + v_y^2 + uu_{yy} + vv_{yy} \end{pmatrix}.$$

The two concerned eigenvalues are the two roots of the characteristic polynomial of $A = \nabla^2 f(x, y)$, which is $t^2 - tr(A)t + \det(A)$. By Cauchy–Riemann's equations again, we have

$$\begin{aligned} tr(A) &= u_x^2 + v_x^2 + u_y^2 + v_y^2 = 2N^2 r^{2N-2} + h.o.t., \\ \det(A) &= (u_x^2 + v_x^2)(u_y^2 + v_y^2) - (uu_{xx} + vv_{xx})^2 - (uu_{xy} + vv_{xy})^2 \\ &= (u_x^2 + v_x^2)(u_y^2 + v_y^2) - (u^2 + v^2)(u_{xx}^2 + v_{xx}^2) \\ &= N^4 r^{4N-4} - N^2(N-1)^2 r^{4N-4} + h.o.t. = N^2(2N+1)r^{4N-4} + h.o.t. \end{aligned}$$

From this, it is easy to arrive at the claimed asymptotic values for the two eigenvalues of $\nabla^2 f(x, y)$: $\lambda_1(z) \sim (2N^2 - N)|\tau|^2 r^{2N-2}$ and $\lambda_2(z) \sim N|\tau|^2 r^{2N-2}$, where $r = ||z||$.

Now, we complete the proof that (3) is satisfied in this case where $z^* = 0$ is a root of $g(z)$. We need to estimate $minsp(A_n)/sp(A_n)$ when $z_n = (x_n, y_n)$ is close to z^*. We note that $A_n = \nabla^2 f(z_n) + \delta||\nabla f(z_n)||^{1+\alpha}$. Hence, the two eigenvalues of A_n are $\lambda_1(z_n) + \delta||\nabla f(z_n)||^{1+\alpha}$ and $\lambda_2(z_n) + \delta||\nabla f(z_n)||^{1+\alpha}$. Note that

$$\begin{aligned} ||\nabla f(z_n)||^{1+\alpha} &= [(uu_x + vv_x)^2 + (uu_y + vv_y)^2]^{(1+\alpha)/2} \\ &= N^{1+\alpha} r^{(2N-1)(1+\alpha)} + h.o.t., \end{aligned}$$

which is of smaller size compared to $\lambda_1(z_n)$ and $\lambda_2(z_n)$. Therefore, we have $minsp(A_n)/sp(A_n) \sim 1/(2N-1)$ for z_n near z^*, which is bounded away from 0 as wanted.

Case 2: $z^* = 0$ is a root of $g'(z)$.

If z^* is also a root of $g(z)$, then we are reduced to Case 1. Hence, we can assume that z^* is not a root of $g(z)$. Therefore, we can expand, in a small open neighbourhood of $z^* = 0$: $g(z) = \gamma + \tau z^N + h.o.t.$, where $\gamma, \tau \neq 0$.

If $N = 1$, then z^* is not a root of $gg"$. Then by Lemma A.1, we obtain that z^* is a saddle point of f. Hence, for z_n near z^* we obtain

$$minsp(A_n)/sp(A_n) \sim minsp(\nabla^2 f(z^*))/sp(\nabla^2 f(z^*)),$$

which is bounded away from 0, as wanted.

Thus, we can assume that $N \geq 2$. Calculating as above we found:

$$tr(\nabla^2 f(z)) = 2|\tau|^2 N^2 r^{2N-2},$$

$$\det(\nabla^2 f(z)) = |\tau|^4 N^4 r^{4N-4} - |\gamma|^2|\tau|^2 N^2 (N-1)^2 r^{2N-4}.$$

Since $N \geq 2$, we have $|\det(\nabla^2 f(z))| >> |tr(\nabla^2 f(z))|^2$ near z^*. This means that the two eigenvalues $\lambda_1(z)$ and $\lambda_2(z)$ of $\nabla^2 f(z)$ are of the same size $\sim |\gamma\tau|N(N-1)r^{n-2}/2$.

Now, the term $||\nabla f(z)||^{1+\alpha}$, which is about the size of

$|\gamma|^{1+\alpha}|\tau|^{1+\alpha}N^{1+\alpha}r^{(N-1)(1+\alpha)}$, is of different size compared to $\lambda_1(z)$ and $\lambda_2(z)$, thanks to the condition that α does not belong to the set $\{(n-3)/(n-1) : n = 2, 3, \ldots\}$. Therefore, we obtain that $minsp(A_n)/sp(A_n) \sim 1$ near z^*. □

B Implementation and Experimental Results

In this "Appendix", we present some implementation details and experimental results on New Q-Newton's method.

B.1 Implementation Details

In this subsection, we present some practical points concerning implementation details, for the language Python. Source code is in the GitHub link [14].

Indeed, Python has already enough commands to implement New Q-Newton's method. There is a package, named numdifftools, which allows one to compute approximately the gradient and Hessian of a function. This package is also very convenient when working with a family $f(x, t)$ of functions, where t is a parameter. Another package, named scipy.linalg, allows one to find (approximately) eigenvalues and the corresponding eigenvectors of a square matrix. More precisely, given a square matrix A, the command $eig(A)$ will give pairs (λ, v_λ) where λ is an approximate eigenvalue of A and v_λ a corresponding eigenvector.

One point to notice is that even if A is a symmetric matrix with real coefficients, the eigenvalues computed by the command eig could be complex numbers, and not real numbers, due to the fact that these are approximately computed. This can be easily resolved by taking the real part of λ, which is given in Python codes by $\lambda.real$. Similarly, we can do this for the eigenvectors. A very convenient feature of the command eig is that it already computes (approximate) orthonormal bases for the eigenspaces.

Now, we present the coding detail of the main part of New Q-Newton's method: Given a symmetric invertible matrix A with real coefficients (in our case $A = \nabla^2 f(x_n) + \delta_j ||\nabla f(x_n)||^{1+\alpha}$), and a vector v, compute w which is the reflection of $A^{-1}.v$ along the direct sum of eigenspace of negative eigenvectors of A. First, we use the command eig to get pairs $\{(\lambda_j, v_j)\}_{j=1,\ldots,m}$, and use the command real to get real parts. If we write $v = \sum_{j=1}^m a_j v_j$, then $a_j =< v_j, v >$ (the inner product), which is computed by the Python command $np.dot(v_j, v)$. Then, $v_{inv} := A^{-1}v = \sum_{j=1}^m (a_j/\lambda_j)v_j$. Finally,

$$w = v_{inv} - 2 \sum_{j:\, \lambda_j < 0} (a_j/\lambda_j)v_j.$$

Remark A.1 (1) We do not need to compute exactly the gradient and the Hessian of the cost function f, only approximately. Indeed, the proof of Theorem 3.1 shows that if one wants to stop when $||\nabla f(x_n)||$ and $||x_n - x_\infty||$ are smaller than a threshold ϵ, then it suffices to compute the gradient and the Hessian up to an accuracy of order ϵ.

Similarly, we do not need to compute the eigenvalues and eigenvectors of the Hessian exactly, but only up to an accuracy of order ϵ, where ϵ is the threshold to stop.

In many experiments, we only calculate the Hessian inexactly using the numdifftools package in Python and still obtain good performance.

(2) While theoretical guarantees are proven only when the hyperparameters $\delta_0, \ldots, \delta_m$ are randomly chosen and fixed from the beginning, in experiments we have also tried to choose—at each iterate n—randomly a δ. We find that this variant, which will be named **Random New Q-Newton's method**, has a performance similar to or better than the original version.

(3) Note that similar commands are also available on PyTorch and TensorFlow, two popular libraries for implementing deep neural networks.

B.2 Some Experimental Results

Here, we present a couple of illustrating experimental results. Additional experiments, which are quite extensive, are available in the arXiv version of the paper. We use the python package numdifftools [12] to compute gradients and Hessian, since symbolic computation is not quite efficient. The experiments here are run on a small personal laptop. The unit for running time is seconds.

Here, we will compare the performance of New Q-Newton's method against several, including well-known, existing variants of Newton's method: the usual Newton's method, BFGS [32], adaptive cubic regularization [7, 19], as well as Random damping Newton's method [24] and Inertial Newton's method [6].

For New Q-Newton's method, we choose $\alpha = 1$ in the definition. Moreover, we will choose $\Delta = \{0, \pm 1\}$, even though for theoretical proofs we need Δ to have at least $m+1$ elements, where $m =$ the number of variables. The justification is that when running New Q-Newton's method it is almost never the case that both $\nabla^2 f(x)$ and $\nabla^2 f(x) \pm ||\nabla f(x)||^2 Id$ are not invertible. The experiments are coded in Python and run on a usual personal computer. For BFGS: we use the function scipy.optimize.fmin_bfgs available in Python and put $gtol = 1e-10$ and $maxiter = 1e+6$. For adaptive cubic regularization for Newton's method, we use the AdaptiveCubicReg module in the implementation in [13]. We use the default hyperparameters as recommended there, and use "exact" for the hessian_update_method. For hyperparameters in Inertial Newton's method, we choose $\alpha = 0.5$ and $\beta = 0.1$ as recommended by the authors of [6]. Source codes for the current paper are available at the GitHub link [14].

Legends We use the following abbreviations: "ACR" for adaptive cubic regularization, "BFGS" for itself, "Rand" for Random damping Newton method, "Newton" for

Newton's method, "Iner" for Inertial Newton's method, "NewQ" for New Q-Newton's method, "R-NewQ" for Random New Q-Newton's method.

Features reported We will report on the number of iterations needed, the function value, and the norm of the gradient at the last point, as well as the time needed to run.

B.2.1 A Toy Model for Protein Folding

This problem is taken from [22]. Here is a brief description of the problem. The model has only two amino acids, called A and B, among 20 that occurs naturally. A molecule with n amino acids will be called an n-mer. The amino acids will be linked together and determined by the angles of bend $\theta_2, \dots, \theta_{n-1} \in [0, 2\pi]$. We specify the amino acids by Boolean variables $\xi_1, \dots, \xi_n \in \{1, -1\}$, depending on whether the corresponding one is A or B. The intramolecular potential energy is given by:

$$\Phi = \sum_{i=2}^{n-1} V_1(\theta_i) + \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} V_2(r_{i,j}, \xi_i, \xi_j).$$

Here, V_1 is the backbone bend potential and V_2 is the non-bonded interaction, given by:

$$\begin{aligned} V_1(\theta_i) &= \frac{1}{4}(1 - \cos(\theta_i)), \\ r_{i,j}^2 &= \left[\sum_{k=i+1}^{j-1} \cos\left(\sum_{l=i+1}^{k} \theta_l\right)\right]^2 + \left[\sum_{k=i+1}^{j-1} \sin\left(\sum_{l=i+1}^{k} \theta_l\right)\right]^2, \\ C(\xi_i, \xi_j) &= \frac{1}{8}(1 + \xi_i + \xi_j + 5\xi_i\xi_j), \\ V_2(r_{i,j}, \xi_i, \xi_j) &= 4(r_{i,j}^{-12} - C(\xi_i, \xi_j) r_{i,j}^{-6}). \end{aligned}$$

Note that the value of $C(\xi_i, \xi_j)$ belongs to the finite set $\{1, 0.5, -0.5\}$.

In the first non-trivial dimension $n = 3$, we have $\Phi = V_1(\theta_2) + V_2(r_{1,3}, \xi_1, \xi_3)$ and $r_{1,3} = 1$. Hence,

$$\Phi = \frac{1}{4}(1 - \cos(\theta_2)) + 4(1 - C(\xi_1, \xi_3)).$$

Therefore, the global minimum (ground state) of Φ is obtained when $\cos(\theta_2) = 1$, at which the value of Φ is $4(1 - C(\xi_1, \xi_3))$. In the special case where $\xi_1 = 1 = \xi_3$ (corresponding to AXA), the global minimum of Φ is 0. This is different from the assertion in Table 1 in [22], where the ground state of Φ has value -0.65821 at $\theta_2 = 0.61866$. Our computations for other small dimension cases $n = 4, 5$ also obtain values different from that reported in Table 1 in [22]. In [22], results are reported for dimension ≤ 5, while those for dimensions 6 and 7 are available upon request.

Table 1 presents the optimal values for the potential-energy function Φ for molecules

Table 1 Optimal values for the potential energy function Φ for n-mers, where $n = 3, 4, 5$

Molecule	$min\ \Phi$	θ_2/π	θ_3/π	θ_4/π	$\Phi(\theta^*)$ in [22]
AAA	0	0			0.3410
AAAA	–0.0615	0	0		0.3226
AAAB	6.0322	0	0		6.3763
AABA	5.3417	0	0.6186		5.4681
ABAB	2.0322	0	0		2.3790
ABBA	11.3417	0	−0.6186		12.0995
BBBB	3.9697	0	0		4.3577
AAAAA	–1.6763	0	0.6183	0.3392	0.7042
AAAAB	5.4147	0	0.6176	−0.0513	6.3677
AAABA	4.5490	0	0.3326	0.6218	4.6503
AAABB	12.0672	0	0	0	12.4117
AABAA	10.3236	0	0.6183	0.3392	11.2914
AABAB	7.4147	0	0.6176	−0.0513	8.3433
AABBA	16.5490	0	0.3326	0.6218	17.4098
ABAAB	11.3506	0	−0.6176	1.2066	12.3050
ABABA	2.0589	0	0	0	4.5373
ABABB	8.0047	0	0	0	8.3525
ABBAB	13.3506	0	0.6176	−0.0667	14.1068
ABBBA	13.9638	0	−0.4768	−0.4768	14.8761
ABBBB	10.0047	0	0	0	10.9039
BAAAB	12.0617	0	0	0	14.1842
BABAB	4.0617	0	0	0	6.1938
BABBB	9.9992	0	0	0	10.4814
BBABB	13.8602	0	−0.5582	−0.3518	14.1087
BBBBB	5.8602	0	−0.5582	−0.3518	6.1185

To save space, only the cases different from [22] are reported. Here, θ^* is the point found by the methods in [22]

n-mer, where $n \leq 5$, found by running different optimization methods from many random initial points. The cases listed here are the same as those in Table 1 in [22]. For comparison, we also compute the function value at the points listed in Table 1 in [22].

Here, we will perform experiments for two cases: ABBBA (dimension 5) and ABBBABABAB (dimension 10). The other cases (of dimensions 5 and 10) yield similar results. We will generate random initial points and report on the performance of the different algorithms. We observe that the performance of Inertial Newton's method and adaptive cubic regularization is less stable, less accurate, or slower than the other methods.

(1) **For ABBBA:** In this case, the performance of New Q-Newton's method and that of Random New Q-Newton's method are very similar, so we report only that of New Q-Newton's method. We found that the optimal value seems to be about 13.963.

Table 2 Performance of different optimization methods for the toy protein folding problem for the 5-mer ABBBA at some random initial points

	ACR	BFGS	Newton	NewQ	Rand	Iner
	Initial point (−0.0534927, 1.61912758, 2.9567358)					
Iterations	7	57	17	31	31	14
f	5e+6	14.058	3e+5	13.963	3e+5	14.255
$\|\|\nabla f\|\|$	1e+8	1e−8	6e−6	5e−12	6e−6	0
Time	0.058	0.843	0.337	0.617	0.594	0.078
	Initial point (1.80953527, −1.74233202, 2.45974152)					
Iterations	5	26	27	15	51	13
f	14.117	13.963	13.963	13.963	14.463	5e+4
$\|\|\nabla f\|\|$	47.388	6e−11	4e−12	8e−12	4e−10	0
Time	0.114	0.1773	0.541	0.317	1.033	0.084
	Initial point (1.07689387, 2.97081771, 0.800213082)					
Iterations	19	57	32	48	32	15
f	283.822	13.963	13.963	13.963	13.963	39.726
$\|\|\nabla f\|\|$	3950.996	1e−10	1e−11	5e−10	4e−10	0
Time	2.760	0.398	0.626	0.642	0.928	0.085

The function values at the initial points are, respectively, 2555432869.1351156; 538.020; and 6596446021.145492

We will test for several (random) choices of initial points:

$$(\theta_2, \theta_3, \theta_4) = (-0.0534927, 1.61912758, 2.9567358),$$

with function value 2555432869.1351156;

$$(\theta_2, \theta_3, \theta_4) = (1.80953527, -1.74233202, 2.45974152),$$

with function value 538.020;
and

$$(\theta_2, \theta_3, \theta_4) = (1.07689387, 2.97081771, 0.800213082),$$

with function value 6596446021.145492.

Table 2 lists the performance of different methods (with a maximum number of 5000 iterates, but can stop earlier if $||\nabla f(z_n)|| < 1e-10$ or $||z_{n+1} - z_n|| < 1e-20$ or there is an unknown error).

(2) **For ABBBABABAB:** In this case, usually Newton's method and Random damping Newton's method encounter the error "Singular matrix". Hence, we have to take more special care of them and reduce the number of iterations for them to 50. In this case, Random New Q-Newton's method can obtain better performances than New Q-Newton's methods, so we report both of them. In this case, it seems that the optimal

Table 3 Performance of different optimization methods for the toy protein folding problem for the 10-mer ABBBABABAB at several random initial points

	ACR	BFGS	Newton	NewQ	Rand	Iner
	Initial point: Point 1					
Iterations	1e+4	197	50	35	50	13
f	7e+7	19.707	Err	1.2e+4	Err	2e+7
$\|\|\nabla f\|\|$	1e+10	6e−10	Err	8e−8	Err	0
Time	395.49	14.27	Err	16.20	Err	0.500
	Initial point: Point 2					
Iterations	66	79	50	70	47	13
f	5e+11	19.596	Err	20.151	20.207	5e+6
$\|\|\nabla f\|\|$	5e+13	5e−8	Err	1e−7	4e−8	0
Time	14.17	4.118	Err	32.76	21.47	0.479
	Initial point: Point 3					
Iterations	0	176	50	500	50	13
f	1e+13	19.727	Err	20.225	Err	3e+9
$\|\|\nabla f\|\|$	1e+15	7e−9	Err	2e−5	Err	0
Time	0	9.91	Err	380.1	Err	0.484
	Initial point: Point 4					
Iterations	1	83	50	95	50	14
f	2e+20	19.596	Err	3e+3	Err	7e+4
$\|\|\nabla f\|\|$	1e+7	3e−8	Err	2e−8	Err	0
Time	2.301	4.365	Err	43.55	Err	0.583

The function values at the initial points are, respectively: 4185029.6878152043; 895386751.0677216; 12479713199090.754; and 579425.218039767. For Newton's method and Random damping Newton's method, we often encounter singular matrix error

value is about 19.387061837218972, which is obtained near the point

$$\begin{aligned}&(\theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8)\\ &= (-4.7735907, -0.47766515, -1.02890588, -1.77319053,\\ &\quad -0.02340005, 0.08208585, -1.39102817, 0.27906532).\end{aligned}$$

Remark. We have tested with many random initial points and found that none of the algorithms here (adaptive cubic regularization, BFGS, Newton's method, New Q-Newton's method, Random Newton's method, Random New Q-Newton's method, and Inertial Newton's method) as well as Backtracking GD can find the above global minimum. The above global minimum value has been found by running Backtracking New Q-Newton's method with, for example, Point 1 below, with running time about 16.2 s.

We will test with 4 random initial points (see Table 3):

Point 1

$$(\theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8) = (-3.00156524, -1.5427558, 1.9394472, -2.74672374, -1.82664375, 1.96928115, -1.26350718, 2.82317321).$$

The function value at the initial point is 4185029.6878152043.

Point 2:

$$(\theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8) = (1.50386159, -1.36306552, 2.93979824, 1.01082799, -1.56261475, 1.61429959, -0.02311273, -1.8108999).$$

The function value at the initial point is 895386751.0677216.

Point 3:

$$(\theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8) = (2.89936055, 2.5913901, -1.40975004, -2.76032304, -3.05060738, 1.09171554, 1.33525563, -1.85212602).$$

The function value at the initial point is 12479713199090.754.

Point 4:

$$(\theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8) = (-1.3335047, 2.76782837, -1.89518385, 2.52345111, -0.33519698, -1.98794015, 0.02088706, -1.09200044).$$

The function value at the initial point is 579425.218039767.

B.2.2 Finding Roots of Univariate Meromorphic Functions

As discussed in Sect. 3.3, given a non-constant univariate function $g(z)$, we will construct a function $f(x, y) = u(x, y)^2 + v(x, y)^2$, where $z = x + iy$, $u =$ is the real part of g, and $v =$ is the imaginary part of g. Global minima of f are exactly roots of g, at which the function value of f is precisely 0. We will apply different optimization algorithms to f. See Table 4.

We will consider several functions. The first is a tricky polynomial [8], for which Lehmer's method [16] encountered errors:

$$\begin{aligned} g_1(z) = {} & 1250162561z^{16} + 385455882z^{15} + 845947696z^{14} + 240775148z^{13} \\ & +247926664z^{12} + 64249356z^{11} + 41018752z^{10} + 9490840z^9 \\ & +4178260z^{18} + 837860z^7 + 267232z^6 + 44184z^5 \\ & +10416z^4 + 1288z^3 + 242z^2 + 16z + 2. \end{aligned}$$

Table 4 Performance of different optimization methods for finding roots of meromorphic functions at random initial points

	ACR	BFGS	Newton	NewQ	Rand	Iner
	Function g_1					
Iterations	Err	Err	149	149	149	Err
f	Err	Err	6e−14	6e−14	6e−14	Err
$\|\|\nabla f\|\|$	Err	Err	9e−11	9e−11	9e−11	Err
Time	Err	Err	2.076	1.935	1.922	Err
	Function g_2, Point 1					
Iterations	0	25	11	11	33	4
f	6482	1e−23	1e−39	1e−40	8e−22	3e+78
$\|\|\nabla f\|\|$	2900	1e−11	0	0	9e−10	0
Time	0.002	0.107	0.112	0.112	0.331	0.015
	Function g_2, Point 2					
Iterations	4	10	5	9	19	6
f	1e−10	4e−24	1	3e−43	1	2e+160
$\|\|\nabla f\|\|$	4e−5	8e−12	0	0	9e−10	0
Time	0.014	0.062	0.051	0.094	0.188	0.020
	Function g_3					
Iterations	Err	1	13	18	Err	Err
f	Err	0.040	0.387	5e−28	Err	Err
$\|\|\nabla f\|\|$	Err	0.205	6e−10	3e−14	Err	Err
Time	Err	16.77	15.43	22.06	Err	Err
	Function g_4					
Iterations	46	132	56	56	54	Err
f	2e−9	8e−15	2e−14	2e−14	2e−14	Err
$\|\|\nabla f\|\|$	7e−7	8e−11	2e−11	2e−11	2e−11	Err
Time	0.159	0.558	0.572	0.578	0.547	Err
	Function g_5					
Iterations	Err	2	18	46	16	Err
f	Err	Err	0.9999	1e−30	0.9999	Err
$\|\|\nabla f\|\|$	Err	Err	2e−11	3e−14	4e−11	Err
Time	Err	79.04	23.55	59.94	20.85	Err

See Sect. B.2.2 for more detail. "Err" means some errors encountered

The (randomly chosen) initial point is $(x, y) = (6.58202917, -7.93929341)$, at which point the function value of f is $4e + 50$.

The second is a simple function, for which the point $(0, 0)$ is a saddle point of the function f:

$$g_2(z) = z^2 + 1.$$

We look at 2 (random initial) points. Point 1: $(x, y) = (4.0963223, -8.0935966)$, at which point the value of f is 6482. Point 2: (closer to the point $(0, 0)$): $(x, y) = (0.317, -0.15)$, at which point the function value of f is 1.171.

The third is a meromorphic function, which is the derivative of the function in formula (7.4) in [8]:

$$g_3(z) = \frac{d}{dz}\left[\frac{1 - 1.005e^{-z} + 0.525e^{-2z} - 0.475e^{-3z} - 0.045e^{-4z}}{2.27e^{-z} - 2.19e^{-2z} + 1.86e^{-3z} - 0.38e^{-4z}}\right].$$

The root of smallest absolute value of g_3 is close to $0.3430042 + 1.0339458i$. It has a pole near $-0.227 + 1.115i$ of absolute value just slightly larger than that of this root, and hence, when one applies the method in [8] one has to be careful. We choose (randomly) an initial point which is close to the pole of g_3: $(x, y) = (-0.227, 1.115)$, at which point the value of f is 0.0415.

The fourth is a polynomial function with multiple roots:

$$g_4(z) = z(z-1)^2(z-2)^3(z-5)^5.$$

We consider a (random) initial point $(x, y) = (4.48270522, 3.79095724)$, at which point the function value is $1e + 14$.

The fifth is the 1001-st summand of the series defining Riemann zeta function:

$$g_5(z) = \sum_{n=1}^{1001} n^{-z}.$$

Here, recall that $n^{-z} = e^{-ln(n)z}$. We choose a (randomly chosen) initial point

$$(x, y) = (9.76536427, -4.15647151),$$

at which the function value is 0.9977.

References

1. Absil, P.-A., Mahony, R., Andrews, B.: Convergence of the iterates of descent methods for analytic cost functions. SIAM J. Optim. **16**(2), 531–547 (2005). https://doi.org/10.1137/040605266
2. Ahookhosh, M., Fleming, R.M.T., Vuong, P.T.: Finding zeros of Hölder metrically subregular mappings via globally convergent Levenberg–Marquardt methods. Optm. Methods Softw. **37**(1), 113–149 (2022). https://doi.org/10.1080/10556788.2020.1712602
3. Ahookhosh, M., Artacho, F.J.A., Fleming, R.M.T., Vuong, P.T.: Local convergence of the Levenberg–Marquardt method under Hölder metric subregularity. Adv. Comput. Math. **45**, 2771–2806 (2019). https://doi.org/10.1007/s10444-019-09708-7
4. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. Pac. J. Math. **16**(1), 1–3 (1966)
5. Bianconcini, T., Sciandrone, M.: A cubic regularization algorithm for unconstrained optimization using line search and nonmonotone techniques. Optim. Methods Softw. **31**(5), 1008–1035 (2016). https://doi.org/10.1080/10556788.2016.1155213

6. Bolte, J., Castera, C., Pauwels, E., Févotte, C.: An inertial Newton algorithm for deep learning. J. Mach. Learn. Res. **22**(134), 1–31 (2021)
7. Cartis, C., Gould, N.I.M., Toint, P.L.: Adaptive cubic regularisation methods for unconstrained optimization. Part 1: motivation, convergence and numerical results. Math. Program. Ser. A **127**, 245–295 (2011). https://doi.org/10.1007/s10107-009-0286-5
8. Delves, L.M., Lyness, J.N.: A numerical method for locating the zeros of an analytic function. Math. Comput. **21**, 543–560 (1967)
9. Fan, J.-Y., Yuan, Y.-X.: On the Quadratic convergence of the Levenberg–Marquardt method without nonsingularity assumption. Computing **74**, 23–39 (2005). https://doi.org/10.1007/s00607-004-0083-1
10. Gill, P.E., Kungurtsev, V., Robinson, D.P.: A stabilized SQP method: global convergence. IMA J. Numer. Anal. **37**(1), 407–443 (2016). https://doi.org/10.1093/imanum/drw004
11. Gill, P.E., Kungurtsev, V., Robinson, D.P.: A stabilized SQP method: superlinear convergence. Math. Program. **163**, 369–410 (2016). https://doi.org/10.1007/s10107-016-1066-7
12. GitHub link for Python's package numdifftools. https://github.com/pbrod/numdifftools
13. GitHub link for adaptive cubic regularization for Newton's method. https://github.com/cjones6/cubic_reg. Accessed 4 Mar 2021
14. GitHub links for Python source codes for New Q-Newton's method and backtracking new Q-Newton's method. https://github.com/hphuongdhsp/Q-Newton-method. https://github.com/tuyenttMathOslo/New-Q-Newton-s-method-Backtracking. https://github.com/tuyenttMathOslo/NewQNewtonMethodBacktrackingForSystemEquations
15. Kato, T.: Perturbation Theory for Linear Operators. In: Originally Published as Volume 132 of the Grundlehren der Mathematischen Wissenschaften. Springer, Berlin (1995). https://doi.org/10.1007/978-3-642-66282-9
16. Lehmer, D.H.: A machine method for solving polynomial equations. J. Assoc. Comput. Mach. **8**, 151–162 (1961). https://doi.org/10.1145/321062.321064
17. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. Q. Appl. Math. **2**(2), 164–168 (1944). https://doi.org/10.1090/qam/10666
18. Marquardt, D.: An algorithm for least-squares estimation of nonlinear parameters. SIAM J. Appl. Math. **11**(2), 431–441 (1963). https://doi.org/10.1137/0111030
19. Nesterov, Y., Polyak, B.T.: Cubic regularization of Newton method and its global performance. Math. Program. Ser. A **108**, 177–205 (2006). https://doi.org/10.1007/s10107-006-0706-8
20. Shen, C., Chen, X., Liang, Y.: A regularized Newton method for degenerate unconstrained optimization problems. Optim. Lett. **6**, 1913–1933 (2012). https://doi.org/10.1007/s11590-011-0386-z
21. Shub, M.: Global Stability of Dynamical Systems. Springer, Berlin (1987). https://doi.org/10.1007/978-1-4757-1947-5
22. Stillinger, F.H., Head-Gordon, T., Hirshfeld, C.L.: Toy model for protein folding. Phys. Rev. E **48**(2), 1469–1477 (1983). https://doi.org/10.1103/PhysRevE.48.1469
23. Smale, S.: Mathematical problems for the next century. Math. Intell. **20**(2), 7–15 (1998). https://doi.org/10.1007/BF03025291
24. Sumi, H.: Negativity of Lyapunov exponents and convergence of generic random polynomial dynamical systems and random relaxed Newton's method. Commun. Math. Phys. **384**, 1513–1583 (2021). https://doi.org/10.1007/s00220-021-04070-6
25. Truong, T.T.: Backtracking new Q-Newton's method: a good algorithm for optimizaton and solving systems of equations. arXiv:2209.05378 (2022)
26. Truong, T.T.: Unconstrained optimisation on Riemannian manifolds. arXiv:2008.11091 (2020)
27. Truong, T.T.: Convergence to minima for the continuous version of backtracking gradient descent. arXiv:1911.04221 (2019)
28. Truong, T.T., Nguyen, T.H.: Backtracking gradient descent method and some applications to large scale optimisation. Part 1: theory. Minimax Theory Appl. **7**(1), 079–108 (2022)
29. Truong, T.T., Nguyen, T.H.: Backtracking gradient descent method and some applications in large scale optimisation. Part 2: algorithms and experiments. Appl. Math. Optim. **84**, 2557–2586 (2021). https://doi.org/10.1007/s00245-020-09718-8
30. Ueda, K., Yamashita, N.: A regularized Newton method without line search for unconstrained optimization. Comput. Optim. Appl. **59**, 321–351 (2014). https://doi.org/10.1007/s10589-014-9656-x

31. Ueda, K., Yamashita, N.: Convergence properties of the regularized Newton method for the unconstrained nonconvex optimization. Appl. Math. Optim. **62**, 27–46 (2010). https://doi.org/10.1007/s00245-009-9094-9
32. Wikipedia page on Quasi-Newton's method. https://en.wikipedia.org/wiki/Quasi-Newton_method
33. Yamashita, N., Fukushima, M.: On the rate of convergence of the Levenberg–Marquardt method. Computing **15**, 237–249 (2021)

Authors and Affiliations

Tuyen Trung Truong[1] · Tat Dat To[2,3] · Hang-Tuan Nguyen[4] · Thu Hang Nguyen[5] · Hoang Phuong Nguyen[5] · Maged Helmy[1,6]

Tat Dat To
tat-dat.to@imj-prg.fr

Hang-Tuan Nguyen
hnguyen@axon.com

Thu Hang Nguyen
hangntt@torus-actions.fr

Hoang Phuong Nguyen
hphuongdhsp@gmail.com

Maged Helmy
magedaa@ifi.uio.no; office@odimedical.com

1 University of Oslo, Oslo, Norway

2 Ecole Nationale de l'Aviation Civile, Toulouse, France

3 Present Address: Sorbonne University, Paris, France

4 Axon AI Research, Seattle, WA, USA

5 Torus Actions SAS, Toulouse, France

6 ODI Medical AS, Oslo, Norway