TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC QUY NHƠN

Phương pháp Newton chỉnh hóa không chính xác cho tối ưu không ràng buộc với tốc độ hội tụ nhanh

Nguyễn Văn Vũ, Trần Ngọc Nguyên*

Khoa Toán và Thống kê, Trường Đại học Quy Nhơn, Việt Nam

Ngày nhận bài: 10/09/2021; Ngày nhận đăng: 22/11/2021

TÓM TẮT

Bài báo đề xuất một phương pháp Newton chỉnh hóa không chính xác để giải các bài toán tối ưu không ràng buộc. Thuật toán được đề xuất thuộc vào lớp lược đồ lặp trong - ngoài. Thay vì giải các hệ tuyến tính một cách chính xác, các chương trình giải các hệ tuyến tính lặp sẽ được áp dụng để tìm ra các hướng tìm kiếm xấp xỉ. Chúng tôi sẽ chứng minh rằng thuật toán không chính xác sẽ bảo toàn tính chất hội tụ địa phương nhanh của các thuật toán chính xác. Một số thực nghiệm số sẽ được thực hiện để chỉ ra những điểm tốt của thuật toán được đề xuất.

Từ khóa: *Tối ưu không ràng buộc, phương pháp Newton không chính xác, chỉnh hóa, chặn sai số địa phương.*

* *Tác giả liên hệ chính.*
Email: tranngocnguyen@qnu.edu.vn

https://doi.org/10.52111/qnjs.2022.16107

JOURNAL OF SCIENCE
QUY NHON UNIVERSITY

Inexact regularized Newton method for unconstrained optimization with rapid rate of convergence

Van Vu Nguyen and Tran Ngoc Nguyen*

Faculty of Mathematics and Statistics, Quy Nhon University, Vietnam

Received: 10/09/2021; Accepted: 22/11/2021

ABSTRACT

This paper proposes an inexact regularized Newton method for solving unconstrained optimization problems. The proposed algorithm belongs to the class of outer-inner iteration scheme. Instead of solving exactly linear systems, iterative linear solver will be applied to find approximate search directions. We will show that the inexact algorithm preserved the fast local convergence property of exact algorithms. Some numerical experiments are also conducted to show the benefits of our proposed algorithm.

Keywords: *Unconstrained optimization, inexact Newton method, regularization, local error bound.*

1. INTRODUCTION

Let us consider the following unconstrained minimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice differentiable function. This type of problem has many applications in machine learning, engineering, economics,....[1,2] Let $\mathcal{S}$ be the solution set of the problem (1). We may assume that the minimization problem (1) has a local solution $x^* \in \mathcal{S}$. A conventional assumption which is used when solving this problem is the second order sufficient condition (SOSC), i.e., the Hessian matrix $\nabla^2 f(x^*)$ is positive definite, see, e.g. Theorem 2.4 in[2]. This assumption is very important because it implies that x^* is the unique local solution of (1). This is crucial for the fast local convergence of Newton algorithm, see, e.g. Theorem 3.5 in[2]. However, in practice, this assumption is somewhat strict and it limits the applicability of numerical algorithm. With the lack of this assumption, behavior of numerical methods can be very bad and sometimes algorithms cannot work. Recently, regularized optimization methods emerge as good alternatives for resolving problems of type (1), in both contexts of convex optimization[3–5] and of nonconvex optimization.[6,7] The main idea of these regularized methods is tend to solve a sequence of regularized problem of the form

$$\min_{x\in\mathbb{R}^n} F_k(x) := f(x) + \frac{\theta_k}{2}\|x - x_k\|^2, \tag{2}$$

where x_k is the iterate at iteration k, $\theta_k > 0$ is a regularization parameter. Instead of the SOSC, in these papers, the authors assume that the gradient of objective provides a local error bound condition at some $x^* \in \mathcal{S}$. Such a condition means that the distance from a point to the solution set of the problem $\mathcal{S}$ can be upper bounded by some term related to the gradient at that point (see (10) below).

* *Corresponding author.*
Email: tranngocnguyen@qnu.edu.vn

https://doi.org/10.52111/qnjs.2022.16107

Very recently, authors in[8] proposed a Newton method applied to the first order optimality conditions of (2). This algorithm belongs to the class of outer-inner iteration scheme. The main role of outer iterations is to compute a trial iterate x_k^+ such that

$$(H_k + \theta_k I)(x_k^+ - x_k) = -\nabla f(x_k), \qquad (3)$$

where H_k is an approximation of the Hessian matrix $\nabla^2 f(x_k)$, x_k is the current iterate and θ_k is updated at the beginning of each iteration. If the trial iterate makes a sufficient reduction on the objective function $F_k(x_k^+)$ and its gradient $\nabla F_k(x_k^+)$, it will be set as the starting point of the next iteration, i.e., $x_{k+1} = x_k^+$. If this is not the case, a sequence of inner iterations for minimizing F_k will be applied for obtaining a sufficient reduction on $F_k(x_{k+1})$ and $\nabla F_k(x_{k+1})$. This outer-inner iterations are also applied in the framework of constrained optimization, see[9–15] for further reading. In both outer and inner iterations, the main computational cost lies in solving exactly the linear system (3). In practical applications where the size of problems are very enormous, factorizing a large matrix to solve such a linear system may take a long time to execute the algorithm. To deal with this problem, inxact Newton method has been proposed for solving nonlinear equation[16] and unconstrained optimization.[17] This method is also applied in the framework of constrained optimization, e.g..[14,18,19] However, in these papers, one still resorts to the second order sufficient conditions for the fast local convergence of their algorithms. In[20], the authors proposed a regularized trust-region Newton method for solving (1). In numerical experiments, authors considered the possibility of applying an inexact solver for solving subproblem which is somewhat similar to (3). However, convergence analysis for their algorithm in this inexact case has not been studied.

In this paper, we will propose an inexact regularied Newton method for solving problem (1). Our algorithm has the same vain of outer-inner algorithm scheme as.[8] However, instead of solving exactly linear systems at each iteration which maybe too expensive, we will introduce an application of inexact method to their algorithm. In this case, the tolerance of the inexactness will be considered carefully so that the fast convergence of the algorithm is still preserved. In particular, at each iteration k, an inexact linear solver, e.g. conjugate gradient, will be applied to solve the "inexact" solution x_k^+ that satisfies

$$\|(H_k + \theta_k I)(x_k^+ - x_k) + \nabla f(x_k)\| \leq \eta_k, \quad (4)$$

where $\{\eta_k\}$ is a sequence of positive number which must be chosen. Many iterative linear solvers can be applied for solving (4), e.g., conjugate gradient method,[21] MINRES,[22] LSQR,[23] GMRES,[24] LSMR.[25] Since the coefficient matrix $H_k + \theta_k I$ is positive definite, we will choose the conjugate gradient method for solving (4). We will prove that this inexactness does not affect to the fast local convergence of the algorithm in[8] even for degenerate cases. More specifically, our proposed algorithm attains a superlinear convergence under a local error bound condition which is milder than the usual SOSC. These good theoretical results will be verified by some numerical experiments. In addition, numerical results also show us that the proposed inexact algorithm can help to reduce the computational time compared with exact algorithms.

The paper is organized as follows. Some notations and description of algorithm will be introduced in Section 2. Section 3 is devoted to the convergence analysis of the proposed algorithm. Some numerical experiments are reported in Section 4 to verify

https://doi.org/10.52111/qnjs.2022.16107

theoretical results and to demonstrate the effectiveness of our proposed algorithm. The paper ends with some conclusion.

2. NOTATION AND ALGORITHM DESCRIPTION

Algorithm 1: (kth iteration)

Input: $m \in \mathbb{N}$, $\gamma > 0, \sigma \in (0,1)$, $\bar{\theta} > 0$, $\kappa > 0$, $\epsilon > 0$, $\eta_{-1} > 0$ and $x_0 \in \mathbb{R}^n$

Output: an optimal solution x_k

1 If $\|\nabla f(x_k)\| \leq \epsilon$, then terminate the algorithm.

2 Choose $\theta_k > 0, \delta_k > 0$ such that $H_k = \nabla^2 f(x_k) + \delta_k I \succeq 0$ and set $\theta_k = \min\{\gamma \|\nabla f(x_k)\|^\sigma, \bar{\theta}\}$, $\eta_k = \kappa \min\{\|\nabla f(x_k)\|^{1+\sigma}, \eta_{k-1}\}$.

3 Compute an trial iterate x_k^+ which satisfies the condition (4).

4 Choose $\zeta_k > 0$ such that $\{\zeta_k\} \to 0$. If $\|\nabla f(x_k^+)\| \leq \zeta_k$, then set $x_{k+1} = x_k^+$. Otherwise, apply a sequence of inner iterations to find x_{k+1} such that

$$\|\nabla f(x_{k+1})\| \leq \zeta_k. \qquad (5)$$

5 Set $k \leftarrow k+1$ and go to Step 1.

Notation For two real vectors $x = [x_1, x_2, \ldots, x_n]^\top$ and $y = [y_1, y_2, \ldots, y_n]^\top$ in the vector space $\mathbb{R}^n$, $x^\top y$ is used to denote the Euclidean scalar product. The associated norm is the ℓ_2-norm, i.e., $\|x\| = (x^\top x)^{1/2}$. The notation $x \leq y$ ($x \geq y$) indicates that $x_i \leq y_i$ (resp. $x_i \geq y_i$) for all $i = \overline{1,n}$. For a vector $x \in \mathbb{R}^n$, the notation $\mathrm{diag}(x)$ stands for the diagonal matrix whose diagonal entries are components of vector x. The induced norm of a rectangular matrix M is defined by $\|M\| = \max\{\|Mx\| : \|x\| \leq 1\}$. Let M be a square symmetric matrix, i.e. $M = M^\top$. The smallest eigenvalue of the matrix M is denoted by $\lambda_{\min}(M)$. The notation $M \succeq 0$ means that M is a positive semi-definite matrix, i.e. $\lambda_{\min}(M) \geq 0$. The open ball of radius r and center x is denoted by $B(x,r) = \{y | \|y - x\| < r\}$. The solution set of the problem (1) will be denoted by $\mathcal{S}$. For every $x \in \mathbb{R}^n$, the notation $\mathrm{d}(x) = \inf_{y \in \mathcal{S}} \|x - y\|$ denotes the distance from x to the solution set $\mathcal{S}$. In this case, if the solution set is nonempty, the notation $\bar{x}$ will be used to denote the projection of x onto $\mathcal{S}$, i.e., $\|x - \bar{x}\| = \mathrm{d}(x)$.

We now introduce our proposed algorithm in this paper. Let m be a natural number and $\gamma > 0, \sigma \in (0,1)$, $\kappa > 0$, $\eta_{-1} > 0$, $\bar{\theta} > 0$ and $\epsilon > 0$. At the beginning of the algorithm, a starting point $x_0 \in \mathbb{R}^n$ should be defined. The details of our algorithm is given in Algorithm 1.

The first step introduces the stopping condition of the algorithm. In Step 2, the regularization parameter δ_k will be chosen such that the approximation of the Hessian matrix is positive semi-definite. In particular, this parameter is chosen such that

$$\delta_k \leq \beta_1 \max\{0, -\lambda_{\min}(\nabla^2 f(x_k))\}, \forall k \in \mathbb{N}, \qquad (6)$$

for some $\beta_1 \geq 1$. This choice means that δ_k is not less than the absolute value of the minimum eigenvalue of $\nabla^2 f(x_k)$. This implies that the matrix $H_k := \nabla^2 f(x_k) + \delta_k I$ is positive semi-definite. In[8] and ,[20] authors proposed to choose $\delta_k = \beta_1 \max\{0, -\lambda_{\min}(\nabla^2 f(x_k))\}$. Apparently, this choice validates the requirement (6). In our algorithm, we adopt the same strategy to choose δ_k. It worths to note that recently, authors in[26] propose a simple search algorithm based on the indefinite factorization method MA57[27] to find an δ_k satisfying (6). The regularization parameter θ_k and the "forced" parameter will be defined based on the norm of the gradient of

https://doi.org/10.52111/qnjs.2022.16107

the current iteration x_k. Step 3 is devoted to calculate an approximation solution x_k^+ with the tolerance η_k defined in Step 2. In Step 4, we choose a tolerance for the condition to trigger the inner iteration algorithm. If the trial iterate x_k^+ creates a sufficient reduction on the first order optimality condition, we set it as the starting point of the next iteration. Otherwise, we will apply a sequence of inner iteration to find an iterate x_{k+1} such that the condition (5) holds true.

Because our main concern in this paper is the local behavior of an inexact regularized Newton method, the globalization scheme is not mentionned here. Nevertheless, we introduce Step 4 to show the possibility of applying globalization strategies. It should be of interest to develop our local scheme to global one by using globalization techniques as in.[8,26]

3. ASYMPTOTIC ANALYSIS OF THE ALGORITHM

Asymptotic analysis of Algorithm 1 is conducted under the following assumptions.

Assumption 1. The function f is twice differentiable, $\nabla^2 f$ is locally Lipschitz continuous and the set $\mathcal{S}$ of minimizers to (1) is nonempty.

From the closeness of $\mathcal{S}$ and the coerciveness of the norm, for every $x \in \mathbb{R}^n$, there exists $\bar{x} \in \mathcal{S}$ such that

$$\mathrm{d}(x) = \|x - \bar{x}\|. \tag{7}$$

Assumption 2. The gradient provides a local error bound condition at some $x^* \in \mathcal{S}$.

From the two above assumptions, there exist postive numbers ℓ, L, r, τ such that for all $x, y \in B(x^*, r)$,

$$\|\nabla f(x) - \nabla f(y)\| \le \ell\|x - y\|, \tag{8}$$
$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \le L\|x - y\|, \tag{9}$$
$$\mathrm{d}(x) \le \tau\|\nabla f(x)\|, \tag{10}$$
$$\gamma\|\nabla f(x)\|^\sigma \le \bar{\theta}. \tag{11}$$

From the definition of θ_k and η_k in Step 2 of Algorithm 1 and inequalities (8), (10) and (11) for all $k \in \mathbb{N}$ such that $x_k \in B(x^*, r)$,

$$\theta_k = \gamma\|\nabla f(x_k)\|^\sigma \ge \gamma b^\sigma \mathrm{d}(x_k)^\sigma, \tag{12}$$
$$\eta_k = \kappa\|\nabla f(x_k)\|^{1+\sigma} \le \kappa\ell^{1+\sigma}\mathrm{d}(x_k)^{1+\sigma}, \tag{13}$$

where $b = 1/\tau$.

Firstly, we recall a result about the upper bound of the regularization parameter δ_k satisfying (6) via the distance function.

Lemma 1 (Lemma 2 in[8])**.** *For all $k \in \mathbb{N}$ such that $x_k \in B(x^*, r/2)$, we then have $\delta_k \le \beta_1 L\mathrm{d}(x_k)$.*

Next lemma to demonstrate that the search direction of the inexact Newton method, i.e. $x_k^+ - x_k$, obtained from (4) will be upper bounded by the distance function evaluated at the current iterate x_k.

Lemma 2. *Let*

$$C_1 := \left(\kappa\ell^{1+\sigma} + \left(\frac{1}{2} + \beta_1\right) Lr^{1-\sigma}\right)\frac{1}{\gamma b^\sigma} + 2.$$

For all $k \in \mathbb{N}$ such that $x_k \in B(x^*, r/2)$,

$$\|x_k^+ - x_k\| \le C_1\mathrm{d}(x_k). \tag{14}$$

Proof. Let $k \in \mathbb{N}$ such that $x_k \in B(x^*, r/2)$. Firstly, let us select $\bar{x}_k \in \mathcal{S}$ with $\mathrm{d}(x_k) = \|x_k - \bar{x}_k\|$. We then have $\nabla f(\bar{x}_k) = 0$. This fact and the Lipschitz continuity of f imply that

$$\begin{aligned}
&\nabla f(x_k)\\
=&\nabla f(x_k) - \nabla f(\bar{x}_k)\\
=&\int_0^1 \nabla^2 f(\bar{x}_k + t(x_k - \bar{x}_k))(x_k - \bar{x}_k)\mathrm{d}t\\
=&\int_0^1 \left[\nabla^2 f(\bar{x}_k + t(x_k - \bar{x}_k)) - \nabla^2 f(x_k)\right] \times\\
&\times (x_k - \bar{x}_k)\mathrm{d}t + \nabla^2 f(x_k)(x_k - \bar{x}_k).
\end{aligned} \tag{15}$$

By noting that $\nabla^2 f(x_k) = (H_k+\theta_k I)-(\delta_k+\theta_k)I$, one gets

$$\begin{aligned}&(H_k+\theta_k I)^{-1}\nabla^2 f(x_k)(x_k-\bar{x}_k)\\ =&x_k-\bar{x}_k-(\delta_k+\theta_k)(H_k+\theta_k I)^{-1}(x_k-\bar{x}_k).\end{aligned} \tag{16}$$

From (4), (15), (16) and using (9), we get

$$\begin{aligned}&\|x_k^+-x_k\|\\ \le&\left\|(H_k+\theta_k I)^{-1}\left[(H_k+\theta_k I)(x_k^+-x_k)\right.\right.\\ &\left.\left.+\nabla f(x_k)\right]\right\|+\left\|(H_k+\theta_k I)^{-1}\nabla f(x_k)\right\|\\ \le&\left\|(H_k+\theta_k I)^{-1}\right\|\left(\eta_k+\frac{L}{2}\|x_k-\bar{x}_k\|^2\right)\\ &+\|x_k-\bar{x}_k\|+(\delta_k+\theta_k)\left\|(H_k+\theta_k I)^{-1}\right\|\times\\ &\times\|x_k-\bar{x}_k\|.\end{aligned} \tag{17}$$

Since $H_k\succeq 0$ and $\theta_k>0$, we have $\|(H_k+\theta_k I)^{-1}\|\le\frac{1}{\theta_k}$. Using this inequality, Lemma 1, (12) and noting that $\|x_k-\bar{x}_k\|=\mathrm{d}(x_k)\le r$, we then deduce from (17) that

$$\begin{aligned}&\|x_k^+-x_k\|\\ \le&\frac{1}{\theta_k}\left(\kappa l^{1+\sigma}\mathrm{d}(x_k)^{1+\sigma}+\frac{L}{2}\mathrm{d}(x_k)^2\right)\\ &+2\mathrm{d}(x_k)+\frac{\beta_1 L}{\theta_k}\mathrm{d}(x_k)^2\\ \le&\left(\left(\kappa\ell^{1+\sigma}+\left(\frac{1}{2}+\beta_1\right)Lr^{1-\sigma}\right)\frac{1}{\gamma b^\sigma}+2\right)\times\\ &\times\mathrm{d}(x_k),\end{aligned}$$

from which completes the proof. □

We now show that the sequence of distances from iterates generated by Algorithm 1 to the solution set $\mathcal{S}$ will decrease with a superlinear rate.

Lemma 3. *Let*

$$C_2=\tau\left(\kappa\ell^{1+\sigma}+\left(\left(\beta_1 L+\frac{L}{2}C_1\right)r^{1-\sigma}+\gamma l^\sigma\right)C_1\right).$$

For all $k\in\mathbb{N}$ such that $x_k\in B(x^*,\frac{r}{2(1+C_1)})$, we have

$$\mathrm{d}(x_k^+)\le C_2\mathrm{d}(x_k)^{1+\sigma}.$$

Proof. Let $k\in\mathbb{N}$ be such that $x_k\in B(x^*,\frac{r}{2(1+C_1)})$. Since $C_1>2$, by virtue of Lemma 2, one gets

$$\begin{aligned}\|x_k^+-x^*\|&\le\|x_k^+-x_k\|+\|x_k-x^*\|\\ &\le C_1\mathrm{d}(x_k)+\|x_k-x^*\|\\ &\le(C_1+1)\|x_k-x^*\|\\ &<\frac{r}{2}.\end{aligned}$$

This means that $x_k^+\in B(x^*,\frac{r}{2})$. Hence, local error bound condition (10) holds at x_k^+, i.e.,

$$\mathrm{d}(x_k^+)\le\tau\|\nabla f(x_k^+)\|. \tag{18}$$

With the notation $u_k^+:=x_k^+-x_k$ we deduce from (4) that

$$\begin{aligned}&\|\nabla f(x_k)+\nabla^2 f(x_k)u_k^+\|\\ \le&\left\|(H_k+\theta_k I)(u_k^+)+\nabla f(x_k)\right\|\\ &+\|(\theta_k+\delta_k)u_k^+\|\\ \le&\eta_k+(\theta_k+\delta_k)\|u_k^+\|.\end{aligned} \tag{19}$$

The differentiability of f gives us

$$\begin{aligned}&\nabla f(x_k^+)\\ =&\nabla f(x_k)+\int_0^1\nabla^2 f(x_k+tu_k^+)u_k^+\mathrm{d}t\\ =&\nabla f(x_k)+\nabla^2 f(x_k)u_k^+\\ &+\int_0^1\left[\nabla^2 f(x_k+tu_k^+)-\nabla^2 f(x_k)\right]u_k^+\mathrm{d}t.\end{aligned}$$

Taking the norm on both sides, using (9), (19) and Lemma 2, we obtain

$$\begin{aligned}&\|\nabla f(x_k^+)\|\\ \le&\eta_k+(\delta_k+\theta_k)\|x_k^+-x_k\|\\ &+\frac{L}{2}\|x_k^+-x_k\|^2\\ \le&\eta_k+(\delta_k+\theta_k)C_1\mathrm{d}(x_k)+\frac{L}{2}C_1^2\mathrm{d}(x_k)^2.\end{aligned} \tag{20}$$

Combining (18) with (20) and using (8), (13) Lemma 1, we then get

$$\begin{aligned}&\mathrm{d}(x_k^+)\\ \le&\tau\left(\kappa l^{1+\sigma}\mathrm{d}(x_k)^{1+\sigma}+(\beta_1 L\mathrm{d}(x)+\gamma l^\sigma\mathrm{d}(x)^\sigma)\times\right.\end{aligned}$$

$$\times C_1 \mathrm{d}(x_k) + \frac{L}{2} C_1^2 \mathrm{d}(x_k)^2 \Big)$$
$$\leq \tau \left(\kappa \ell^{1+\sigma} + \left(\left(\beta_1 L + \frac{L}{2} C_1 \right) r^{1-\sigma} + \gamma l^{\sigma} \right) \times \right.$$
$$\left. \times C_1 \right) \mathrm{d}(x_k)^{1+\sigma},$$

which completes the proof. □

By following the same argument as in Section 2 of ,[26] we can prove that if the sequence of iterates is close enough to the region of the solution set $\mathcal{S}$, the inner iteration algorithm will never be triggered. Moreover, Algorithm 1 will converge to some solution of the problem (1) with a superlinear rate of converence. The main result of this paper is summarized in the theorem below.

Theorem 4. *Let Assumptions 1 and 2 hold at $x^* \in \mathcal{S}$. Assume that Algorithm 1 generates an infinite sequence of iterates $\{x_k\}$. There exists $R > 0$ such that if at an iteration $k_0 \in \mathbb{N}$, $x_{k_0} \in B(x^*, R)$, then for all $k \geq k_0$, $x_{k+1} = x_k^+$, $\{x_k\}$ converges to $\widehat{x} \in \mathcal{S}$ and*

$$\lim_{k\to\infty} \frac{\|x_{k+1} - \widehat{x}\|}{\|x_k - \widehat{x}\|} = 0.$$

4. NUMERICAL EXPERIMENTS

In this section, we will make some numerical experiments to show the advantages of the our proposed algorithm. Algorithm 1 is implemented in MATLAB R2012a. Parameters of this algorithm are chosen as below: $\epsilon = 10^{-8}$, $\sigma = 0.5$, $\bar{\theta} = 0.1$, $\gamma = 10^{-2}$, $\kappa = 0.99$, $\eta_{-1} = 0.1$ $\beta_1 = 2$. The conjugate gradient method[2] will be applied to solve system (4). Because we are only interested in the local behavior of Algorithm 1, the globalization strategy is not implemented. In particular, we do not invoke Steps 4 in Algorithm 1. Instead, we will choose starting points which are sufficiently close to the optimal solution of the problem for which Theorem 4 can be applied. The investigation related to global behavior of this algorithm is out of the current work and should be the topic of another research in the future.

4.1. Superlinear convergence of Algorthm 1

This section is devoted to verify the theoretical research developed in this paper. In particular, we will show that Algorithm 1 attains the superlinear rate of convergence in some neighborhood of an optimal solution in which the local error bound condition holds true.

Let us consider the problem (1) in $\mathbb{R}^2$, where

$$f(x) = \begin{cases} \frac{1}{2}(x_2 - 1)^2 & \text{if } x_1 \in [1, 11] \\ \frac{1}{8}(x_1 - 1)^4 (x_1 - 11)^4 & \\ +\frac{1}{2}(x_2 - 1)^2 & \text{otherwise.} \end{cases} \tag{21}$$

The first and second derivatives of f are

$$\nabla f(x) = \begin{cases} \begin{pmatrix} 0 \\ x_2 - 1 \end{pmatrix} & \text{if } x_1 \in [1, 11] \\ \begin{pmatrix} (x_1 - 1)^3 \times \\ \times (x_1 - 11)^3 (x_1 - 6) \\ x_2 - 1 \end{pmatrix} & \text{otherwise.} \end{cases}$$

and

$$\nabla^2 f(x) = \begin{cases} \mathrm{diag}([0, 1]^\top) & \text{if } x_1 \in [1, 11] \\ \mathrm{diag}([(x_1 - 1)^2 (x_1 - 11)^2 \times \\ \times (7x_1^2 - 84x_1 + 227), 1]^\top), \\ & \text{otherwise.} \end{cases}$$

The function f is twice continuously differentiable and the second derivative $\nabla^2 f$ is Lipschitz continuous on $\mathbb{R}^2$. The solution set is $\mathcal{S} = [1, 10] \times \{1\}$. The local error bound condition (10) holds at any $x^* = (x_1^*, 1) \in \mathcal{S}$ such that $1 < x_1^* < 10$. Indeed,

let $r = \min\{x_1^* - 1, 10 - x_1^*\} > 0$. For all $x = (x_1, x_2) \in B(x^*, r)$, we have

$$\begin{aligned} d(x, \mathcal{S})^2 &= (x_2 - 1)^2 \\ &= \|\nabla f(x)\|^2 \end{aligned}$$

which implies that the error bound condition (10) is validated at x^* with $\tau = 1$. We note that, however, the matrix $\nabla^2 f$ is singular for all $x = (x_1, x_2)$ such that $x_1 \in [1, 10]$ which means that the SOSC does not holds in this example. From the starting point $x_0 = (9.0, -50)$, our algorithm converges to solution $\widehat{x} = (9.0, 1.0)$ after 4 iterates. The behavior of Algorithm 1 in this example is showed in Figure 1. In this figure, we plot norms of $x_k - x^*$, where $x^* = [9, 1]$ is the optimal solution. From this figure, we can see that the slope of the graph after each iteration will be more negative. This means that the sequence $\{\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|}\}$ tends to zero. In other word, Algorithm 1 attains the superlinear convergence in this case.

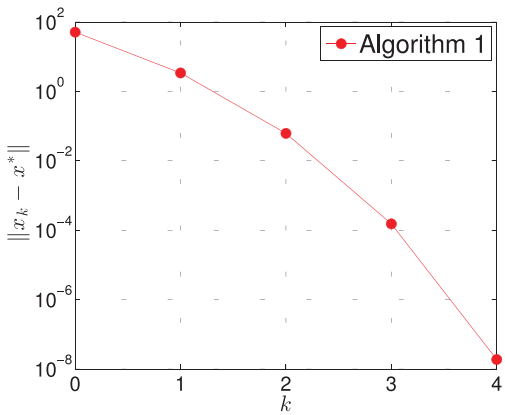

Figure 1: *Behavior of Algorithm 1 when solving* (1) *with* f *is given by* (21).

4.2. Execution time on large-scale problems

As we have mentioned in Section 1, when solving large-scale problems, an exact algorithm may take a long time to solve the linear system (3) which is the main burden in Newton method. Our aim in this section is to show the advantage of our inexact scheme when solving large-scale optimization problems. In particular, we will implement an exact version of Algorithm 1 in which the linear system 3 will be solved (exactly) instead of Step 3 of Algorithm 1. To solve the linear sytem (3), the coefficient matrix $H_k + \theta_k I$ will be factorized by an LDL decomposition, see, e.g.,.[28] Because there is no computation of the square roots of the diagonal elements is needed, this decomposition is more stable and more efficient than Gaussian elimination or Cholesky decomposition.

Problem	Size (n)	Inexact algorithm	Exact algorithm
`Arwhead`	5000	**58.02**	70.99
`bdqrtic`	5000	**2.43**	2.54
`broydn7d`	1000	**19.91**	19.95
`brybnd`	5000	**87.30**	94.86
`dqdrtic`	5000	52.08	**44.12**
`edensch`	5000	**6.48**	6.83
`engval1`	5000	88.27	**87.85**
`freuroth`	5000	**165.98**	169.00
`noncvxun`	1000	4.20	**3.44**
`penalty1`	1000	**7.65**	8.86
`sensors`	1000	**116.56**	122.10

Table 1: *Execution times (in second) of the inexact and the exact algorithms on large-scale problem in CUTEst collection (n is the number of variables)*

In this section, we will consider some large-scale unconstrained problems under form (1) in the CUTEst collection.[29] We will compare CPU times to solve each problem by the inexact and the exact algorithms. Table 1 shows us the numerical results when applying these two algorithms in solving some problems in CUTEst. We will collect problems which satisfy two requirements:

https://doi.org/10.52111/qnjs.2022.16107

the size of a problem (the number of variables) is greater than 1000, and both algorithms are succeed in solving the problem within 200s. We compare only the problems with $n \geq 1000$ because these two algorithms solve the others very fast. This makes the comparisons unfair. From this table, we can see that in most of problems, the inexact algorithm take less time to solve than the exact one. This demonstrates the benefit when using an inxact algorithm instead of the exact algorithm to solve unconstrained optimization problem.

5. CONCLUSION

In this paper, we propose an inexact regularized Newton method for unconstrained optimization. The algorithm is a variant of the algorithm in[8] where linear systems for finding search direction are solved approximately with a suitable tolerance. Assymptotic convergence analysis is performed to show that under some local error bound condition, the algorithm attains a superlinear rate of convergence. Some numerical experiments are conducted to verify theoretical results and to show the advantage of our proposed algorithm. In the future, some researches for globalizing our proposed algorithm should be interested.

Acknowledgment

The authors would like to thank the anonymous referee(s) for his/her careful reading and insightful comments. *This research is conducted within the framework of science and technology projects at institution level of Quy Nhon University under the project code T2021.699.02.*

REFERENCES

1. L. T. Biegler. *Nonlinear Programming.* Society for Industrial and Applied Mathematics, 2010.
2. J. Nocedal and S. J. Wright. *Numerical optimization*, Springer Series in Operations Research and Financial Engineering, New York: Springer, second ed., 2006.
3. C. Humes, Jr. and P. J. S. Silva. Inexact proximal point algorithms and descent methods in optimization, *Optimization and Engineering*, **2005**, *6*(2), 257–271.
4. D.-H. Li, M. Fukushima, L. Qi, and N. Yamashita. Regularized Newton methods for convex minimization problems with singular solutions, *Computational Optimization and Applications*, **2004**, *28*(2), 131–147.
5. Y.-J. Li and D.-H. Li. Truncated regularized Newton method for convex minimizations, *Computational Optimization and Applications*, **2009**, *43*(1), 119–131.
6. W. W. Hager and H. Zhang. Self-adaptive inexact proximal point methods, *Computational Optimization and Applications*, **2008**, *39*(2), 161–181.
7. M. Fuentes, J. Malick, and C. Lemaréchal. Descentwise inexact proximal algorithms for smooth optimization, *Computational Optimization and Applications*, **2012**, *53*(3), 755–769.
8. P. Armand and I. Lankoandé. An inexact proximal regularization method for unconstrained optimization, *Mathematical Methods of Operations Research*, **2017**, *85*(1), 43–59.

9. P. Armand, J. Benoist, and D. Orban. From global to local convergence of interior methods for nonlinear optimization, *Optimization Methods and Software*, **2013**, *28*(5), 1051–1080.

10. P. Armand, J. Benoist, R. Omheni, and V. Pateloup. Study of a primal-dual algorithm for equality constrained minimization, *Computational Optimization and Applications*, **2014**, *59*(3), 405–433.

11. P. Armand and R. Omheni. A globally and quadratically convergent primal-dual augmented Lagrangian algorithm for equality constrained optimization, *Optimization Methods and Software*, **2017**, *32*(1),1–21.

12. P. Armand and R. Omheni. A mixed logarithmic barrier-augmented Lagrangian method for nonlinear optimization, *Journal of Optimization Theory and Applications*, **2017**, *173*(2), 523–547.

13. P. Armand and N. N. Tran. An augmented lagrangian method for equality constrained optimization with rapid infeasibility detection capabilities, *Journal of Optimization Theory and Applications*, **2019**, *181*, 197–215.

14. S. Arreckx and D. Orban. A Regularized Factorization-Free Method for Equality-Constrained Optimization, *SIAM Journal on Optimization*, **2018**, *28*(2), 1613–1639.

15. P. Armand and N. N. Tran. Rapid infeasibility detection in a mixed logarithmic barrier-augmented lagrangian method for nonlinear optimization, *Optimization Methods and Software*, **2019**, *34*(5), 991–1013.

16. R. S. Dembo, S. C. Eisenstat, and T Steihaug. Inexact newton methods, *SIAM Journal on Numerical Analysis*, **1982**, *19*(2), 400–408.

17. R. S. Dembo and T. Steihaug. Truncated-newtono algorithms for large-scale unconstrained optimization, *Mathematical Programming*, **1983**, *26*, 190–212.

18. J.-S. Pang. Inexact newton methods for the nonlinear complementarity problem, *Mathematical Programming*, **1986**, *36*, 54–71.

19. P. Armand, J. Benoist, and J.-P. Dussault. Local path-following property of inexact interior methods in nonlinear programming, *Computational Optimization and Applications*, **2012**, *52*, 209–238.

20. K. Ueda and N. Yamashita. A regularized Newton method without line search for unconstrained optimization, *Computational Optimization and Applications*, **2014**, *59*(1-2), 321–351.

21. M. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems, *Journal of Research of the National Institute of Standards and Technology*, **1952**, *49*, 409–435.

22. C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations, *SIAM Journal on Numerical Analysis*, **1975**, *12*(4), 617–629.

23. C. C. Paige and M. A. Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares, *ACM Transactions on Mathematical Software*, **1982**, *8*, 43–71.

https://doi.org/10.52111/qnjs.2022.16107

24. Y. Saad and M. H. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on Scientific and Statistical Computing*, **1986**, 7(3), 856–869.

25. D. C.-L. Fong and M. Saunders. Lsmr: An iterative algorithm for sparse least-squares problems, *SIAM Journal on Scientific Computing*, **2011**, *33*(5), 2950–2971.

26. P. Armand and N. N. Tran. Local convergence analysis of a primal–dual method for bound-constrained optimization without sosc, *Journal of Optimization Theory and Applications*, **2021**, *189*, 96–116.

27. I. S. Duff, MA57—a code for the solution of sparse symmetric definite and indefinite systems, *ACM Transactions on Mathematical Software*, **2004**, *30*(2), pp. 118–144.

28. G. H. Golub and C. F. van Loan. *Matrix Computations*. JHU Press, fourth ed., 2013.

29. N. I. M. Gould, D. Orban, and P. L. Toint. Cutest: a constrained and unconstrained testing environment with safe threads for mathematical optimization, *Computational Optimization and Applications*, **2015**, *60*, 545–557.

https://doi.org/10.52111/qnjs.2022.16107