Global convergence of the derivative-free trust region algorithm using inexact information on function values

Guidance

Associate Professor Nobuo YAMASHITA

Noritoshi KUROKAWA

Department of Applied Mathematics and Physics

Graduate School of Informatics

Kyoto University

February 2009

Global convergence of the derivative-free trust region algorithm using inexact information on function values

Noritoshi KUROKAWA
(System Optimization Laboratory)

Abstract

The goal of this paper is to propose a globally convergent algorithm for the unconstrained optimization problem where highly accurate evaluations of the objective function are computationally expensive. Specifically, we consider the case where the derivatives of the computed objective function are prohibitively expensive to compute, although the accuracy of the objective function values is controlled by the user. In fact, it is desirable to keep the accuracy as low as possible to save the computation time.

Carter proposed a trust-region algorithm that controls the accuracy of both objective function and gradient evaluations. He showed that his algorithm has the global convergence property if the relative error of the approximate gradient at each iteration is less than some constant. However, it is difficult to estimate the relative error in our situation, hence Carter's algorithm is not applicable. On the other hand, Conn *et al.* proposed a derivative-free trust-region method for unconstrained optimization. The method uses model functions constructed from some sample points and their exact function values, and does not exploit approximate gradients explicitly. Conn *et al.* showed its global convergence under the condition that the constructed model functions are fully-linear. Note that this condition can be ensured more easily than the relative error condition by Carter. However, the algorithm proposed by Conn *et al.* needs exact function evaluations.

In this paper, we develop a derivative-free trust-region algorithm that adaptively controls the accuracy of the objective function evaluations. We first give conditions on sample points and their point-wise accuracies under which the model function constructed from those points is guaranteed to be fully-linear. Then we propose a procedure of updating sample points and their accuracies according to those conditions and establish the global convergence of the proposed algorithm.

Contents

1 Introduction

The goal of this paper is to propose a globally convergent algorithm for numerical optimization of functions that are in general only inaccurately evaluable. Consider the unconstrained optimization problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \ f(\boldsymbol{x}), \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable with Lipschitz continuous gradient $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$. In this paper, we suppose that exact values of f and ∇f are unavailable, although f is smooth. Especially, we suppose the following situation for the evaluations of f:

(S0) Evaluations of f are computationally expensive.

(S1) Evaluated objective function value $\hat{f}(\boldsymbol{x})$ contains a certain amount of error e,

$$\hat{f}(\boldsymbol{x}) = f(\boldsymbol{x}) + e.$$

(S2) The upper bound of the error in the function evaluation is controllable, that is, we can compute $\hat{f}(\boldsymbol{x})$ at any $\boldsymbol{x}$ so that $|e| \leq \varepsilon$ for any accuracy ε.

(S3) The more accurate $\hat{f}$ is sought, the more computation time is required.

(S4) Derivatives of f are not available analytically, or it is not easy to compute.

Situation (S0)-(S4) often arises when a function value is evaluated by executing some complex computer simulation, such as numerically solving a system of partial differential equations, governing underlying the physical phenomena or numerically calculating the integrations by a Monte Carlo method. Examples of these types of problems can be found in [2, 15].

In the situation (S0)-(S4), derivative-based optimization algorithms (e.g. steepest descent method, Newton method, quasi-Newton method) cannot be applied to problem (1.1) directly. Indeed, derivative-based methods require the exact gradient and (approximate) Hessian of f to build a model function of f at each iteration $\boldsymbol{x}_k$. We note that the finite difference approximation based on inexact function evaluations is unreliable, and calculating it with high accuracy requires a lot of computation time [8].

Until now, a lot of algorithms that do not use the exact gradient information have been proposed. Those methods can be categorized into two classes. One is to use approximate gradients and (exact) function values, the other is to use function values only.

The well-known algorithms that use approximate gradients are *the trust region algorithm using inexact gradient information* by Carter [1, 2], and *the implicit filtering* by Kelley [9]. Carter considered the case where approximation of $\nabla f(\boldsymbol{x}_k)$ is available at each iteration. He proposed the trust region algorithm using an approximate gradient, say $\boldsymbol{g}_k$, to build a quadratic model function m_k at each iteration. In [1], he proved the global convergence of his algorithm under the following conditions:

- Evaluations of f are exact,
- The error between the gradient of the model and that of the objective function satisfies

$$\|\nabla f(\boldsymbol{x}_k) - \nabla m_k(\boldsymbol{x}_k)\| \leq \Theta \|\nabla m_k(\boldsymbol{x}_k)\| \tag{1.2}$$

  for all k and for some constant Θ (independent of k) which satisfies $0 \leq \Theta < 1$.

Note that the latter condition states that, the absolute error on the gradient must tend to zero when the gradient of the model itself converges to zero. Unfortunately, it is difficult to ensure this condition under

(S3)-(S4), and Carter's algorithm is not suitable for our situation. In addition, the implicit filtering also requires a similar condition for global convergence [12].

In fact, there exist many methods that do not use approximate derivatives explicitly, such as direct search methods, meta heuristics and *derivative-free* trust region methods [8]. In this paper we are interested in derivative-free trust region algorithms, which use model functions based on *sampling*. The term *derivative-free* means that these methods maintain model functions which are based only on the evaluated objective function at some sample points (i.e., not based on the approximate gradient). Moreover, various model functions have been proposed for these algorithms —such as linear models or quadratic models ([14], [4], [10], [16]), RBF models ([13], [19]) and so on. The corresponding model functions can be constructed by means of interpolation or regression or any other approximation technique. In this paper, these model functions are used to call by *derivative-free model functions*, or, *derivative-free models*[*1], in short.

Derivative-free trust region algorithms have been much studied recently, and some benchmark tests (e.g. [11]) report their efficiency when compared against other derivative-free methods (such as direct search method and meta heuristics). Moreover, some of the derivative-free trust region methods have a global convergence property when f is evaluated exactly [5, 8]. However, it is difficult to establish global convergence of these algorithms under our situation (S0)-(S4). Moreover, since most of these methods are constructed without taking into account (S2) and (S3), they are supposed to use a highly accurate function values, which causes a lot of computation time under (S3).

Under (S2) and (S3), since there is a trade-off between the function evaluation time and its accuracy, it would be reasonable to set a lower accuracy level when a point is far from a solution, and to set it higher at a point in the neighborhood of a solution, for saving computation time. Based on this idea, Conn *et al.* [3, Section 10.6] proposed a trust-region algorithm with dynamic accuracy using inexact gradients. They proved that their algorithm has a global convergent property if the following conditions are both satisfied: a) the algorithm updates the evaluation accuracy at the current point, say $\varepsilon_k^{\mathrm{cur}}$, so that the accuracy is less than the reduction of the model function, b) the gradient of the model satisfies (1.2) at each iteration. However, when we use derivative-free models in their method, we cannot guarantee its global convergence, since the gradient of the derivative-free model does not necessarily satisfy the condition (1.2). (Of course, the condition (1.2) can be ensured if f is evaluated with high accuracy at the sample points. However, it will take a lot of computational time, which is not allowed in our situation.)

In this work, we propose a globally convergent trust region algorithm based on derivative-free models with dynamic accuracy on function evaluations. At each iterations, the algorithm updates pointwise accuracy ν of the sample points based on the trust region radius Δ, that is, $\nu = \mathcal{O}(\Delta^2)$. Indeed, this idea was motivated by the algorithm of Takaki *et al.* [15]. Where, the trust-region algorithm they proposed uses a quadratic model function constructed by the support vector regression from sample points with pointwise accuracy, and updates their accuracy by taking into account the current trust region radius. They reported some numerical results and they concluded that their method is effective in terms of the total computation time. However, they did not prove its global convergence property. Here, we modify their algorithm so as to have global convergence. Then we give sufficient conditions on the accuracy and model functions for the global convergence. Moreover, we discuss how to construct model functions which satisfy those conditions.

[*1] If there is no fear of misunderstanding, we often omit the term *derivative-free*.

The paper is organized as follows. In the next section, we introduce a framework of derivative-free trust region method, and the concept of *fully-linear model*, which is essential for proving global convergence of the derivative-free trust region methods. In section 3.1, we will discuss the accuracy of the evaluated function values in order to construct fully-linear models. In section 3.2, we propose an derivative-free trust-region algorithm with dynamic accuracy on function evaluations, and give all additional conditions to establish the global convergence property of our method. We will prove the global convergence of our method in Section 3.3. In section 4, we discuss how to construct model functions which satisfy the conditions described in Section 3. Finally, Section 5 concludes the paper.

We use the following notations throughout this paper. $\|\cdot\|$ denotes the Euclidean norm (or the matrix norm induced by the Euclidean norm). For a matrix $A = (a_{ij}) \in \mathbb{R}^{m\times n}$, $\|A\|_F$ denotes the Frobenius norm of A, i.e., $\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$. A function $h : \mathbb{R}^n \to \mathbb{R}^m$ is said to be *Lipschitz* continuous with constant L in an open convex region Ω if $\|h(\boldsymbol{x}) - h(\boldsymbol{y})\| \le L\|\boldsymbol{x} - \boldsymbol{y}\|\ \forall \boldsymbol{x}, \boldsymbol{y} \in \Omega$. $\mathcal{L}(\boldsymbol{x}_k)$ denotes the *level set* of a function f at a point $\boldsymbol{x}_k \in \mathbb{R}^n$, that is,

$$\mathcal{L}(\boldsymbol{x}_k) = \{\, \boldsymbol{x} \in \mathbb{R}^n \,|\, f(\boldsymbol{x}) \le f(\boldsymbol{x}_k) \,\}.$$

For a set X, $|X|$ denotes the number of elements of X.

2 Framework of derivative-free trust-region algorithm

In this section, we introduce a general framework of derivative-free trust-region method and present sufficient conditiond for its global convergence.

First, we review the framework of the basic trust-region algorithms (see [3, 12], for more information). At each iteration k, trust-region algorithms build a model function $m_k(\boldsymbol{x})$ around the current iterate $\boldsymbol{x}_k$. It is assumed that this model approximates the objective function f sufficiently well within a neighborhood of the current iterate, the so-called trust-region. This region is taken for simplicity as the set of all points

$$B(\boldsymbol{x}_k; \Delta_k) = \{\, \boldsymbol{x} \in \mathbb{R}^n \,|\, \|\boldsymbol{x} - \boldsymbol{x}_k\| \le \Delta_k \,\},$$

where Δ_k is called the trust-region radius.

In order to obtain each step, trust-region algorithms seek a solution of the subproblem

$$\underset{\boldsymbol{s}\in\mathbb{R}^n}{\text{minimize}}\ \ m_k(\boldsymbol{x}_k + \boldsymbol{s}) \quad \text{subject to } \|\boldsymbol{s}\| \le \Delta_k. \tag{2.1}$$

As we describe below, we only need an approximate solution of (2.1) to obtain global convergence.

Given an approximate solution $\boldsymbol{s}_k$ to (2.1), the pair $(\boldsymbol{x}_k, \Delta_k)$ is updated according to the ratio of actual to predicted reduction,

$$\rho_k = \frac{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \boldsymbol{s}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}. \tag{2.2}$$

Given inputs $0 \le \eta_0 \le \eta_1$, $0 < \gamma_{\text{dec}} < 1 < \gamma_{\text{inc}}$, $0 < \Delta_0 \le \Delta_{\max}$, and $\boldsymbol{x}_0 \in \mathbb{R}^n$, a basic trust-region algorithm proceeds iteratively as follows:

Algorithm 2.1 Iteration k of a basic trust-region algorithm

Step 1: Build model m_k which approximates f in the trust-region $B(\boldsymbol{x}_k; \Delta_k)$.

Step 2: Obtain step $\boldsymbol{s}_k$ by approximately solving the subproblem (2.1).

Step 3: Evaluate $f(\boldsymbol{x}_k + \boldsymbol{s}_k)$ and compute ρ_k using (2.2).

Step 4: Update the trust-region and the current iterate:

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_{\rm inc}\Delta_k, \Delta_{\max}\} & \text{if } \rho_k \geq \eta_1 \\ \Delta_k & \text{if } \eta_0 \leq \rho_k < \eta_1 \\ \gamma_{\rm dec}\Delta_k & \text{if } \rho_k < \eta_0, \end{cases}$$

$$\boldsymbol{x}_{k+1} = \begin{cases} \boldsymbol{x}_k + \boldsymbol{s}_k & \text{if } \rho_k \geq \eta_0 \\ \boldsymbol{x}_k & \text{if } \rho_k < \eta_0. \end{cases}$$

From the design of Algorithm 2.1, we can say that all iterates $\boldsymbol{x}_k$ belong to the level set $\mathcal{L}(\boldsymbol{x}_0)$. However, when we consider models based on sampling, it might be possible (especially at early iterations) that f is evaluated outside $\mathcal{L}(\boldsymbol{x}_0)$. Thus, we need to define the region in which f is evaluated rigorously. In this paper, we assume that f is only sampled within the relaxed level set

$$\mathcal{L}_{\rm enl}(\boldsymbol{x}_0) = \{\, \boldsymbol{y} \in \mathbb{R}^n \,|\, \|\boldsymbol{x} - \boldsymbol{y}\| \leq \Delta_{\max} \text{ for some } \boldsymbol{x} \text{ with } f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0) \,\}, \tag{2.3}$$

where $\Delta_{\max}$ is a given maximum trust region radius.

2.1 Derivative-free models—Fully-linear models

In situation (S0)-(S4), exact derivatives are unavailable to construct a model m_k. Thus, we adopt derivative-free models which are based only on the evaluated objective function at some sample points. However, since we do not use exact derivatives, smoothness of the function f is no longer sufficient for guaranteeing that the model m_k approximates the function locally. Therefore, it is required that the derivative-free models, which have a uniform local behavior, similar to what is observed by derivative-based models. Conn *et al.* called such models, depending on their accuracy, *fully-linear* and *fully-quadratic*. Here, we focus on a class of fully-linear models, which can be formed using as few as $n+1$ function evaluations. The definition of fully-linear model is given below.

Definition 2.1 (fully-linear model, c.f. [8])
Let a function $f : \mathbb{R}^n \to \mathbb{R}$, that the gradient ∇f is Lipschitz continuous on $\mathcal{L}_{\rm enl}(\boldsymbol{x}_0)$, be given. Let positive constants κ_{ef} and κ_{eg} be given and fixed. Suppose that a model m is continuously differentiable on $\mathbb{R}^n$. For any given $\Delta \in (0, \Delta_{\max})$ and for any given $\boldsymbol{x}_c \in \mathcal{L}(\boldsymbol{x}_0)$, the model m is said to be ***fully-linear on $B(\boldsymbol{x}_c; \Delta)$ with respect to*** [*2] $\kappa_{\rm ef}$ ***and*** $\kappa_{\rm eg}$ *if the following inequalities are hold:*

- *the error between the gradient of the model and the gradient of the function satisfies*

$$\|\nabla f(\boldsymbol{x}) - \nabla m(\boldsymbol{x})\| \leq \kappa_{\rm eg}\Delta, \quad \forall \boldsymbol{x} \in B(\boldsymbol{x}_c; \Delta), \tag{2.4}$$

  and
- *the error between the model and the function satisfies*

$$|f(\boldsymbol{x}) - m(\boldsymbol{x})| \leq \kappa_{\rm ef}\Delta^2, \quad \forall \boldsymbol{x} \in B(\boldsymbol{x}_c; \Delta). \tag{2.5}$$

If the function evaluations are exact, fully-linear models (by means of interpolation or regression) are defined by geometric conditions on the sample points [6, 7]. For example, let us consider the case when

[*2] If there is no fear of misunderstanding, we often omit the term "with respect to $\kappa_{\rm ef}$ and $\kappa_{\rm eg}$".

we construct a linear model function by means of interpolation using the evaluated function values at $n+1$ sample points. In this case, if the sample points are "well" affinely independent, the resulting model function is fully-linear.

2.2 A globally convergent algorithm for 1st-order critical points (using exact function values)

Using fully-linear models, Conn *et al.* [5, 8] have established the global convergence of the derivative-free trust-region algorithm using exact function values. Their algorithm shares the common features of the standard (which means "derivative-based") trust-region methods. However, it differs from the standard framework in some aspects.

The most remarkable difference is that their algorithm has the "final criticality subroutine", which invoked if the gradient of the current model is small. This subroutine ensures that the gradient of the current model is not too far from the true gradient of the objective function at the current point. For the purpose of this, this subroutine shrink the trust-region radius and improve the model so that it becomes fully-linear on the trust-region until a certain stopping criterion is met. Fortunetely, it is proved that this subroutine will terminate in a finite number of iterations, and this subroutine is well-defined.

The detail of their proposed algorithm [8] is shown in Algorithm 2.2 (main algorithm) and Algorithm 2.3 (final criticality subroutine). Here, the standard trust-region inputs $\eta_0, \eta_1, \gamma_{\mathrm{dec}}, \gamma_{\mathrm{inc}}$ and the additional constants $\varepsilon_{\mathrm{cri}}, \beta, \mu, \alpha, \kappa_{\mathrm{md}}, \kappa_{\mathrm{ef}}, \kappa_{\mathrm{eg}}$ are given and satisfy the conditions $0 \leq \eta_0 \leq \eta_1 < 1$ (with $\eta_1 \neq 0$), $0 < \gamma_{\mathrm{dec}} < 1 < \gamma_{\mathrm{inc}}$, $\varepsilon_{\mathrm{cri}} > 0$, $\mu > \beta > 0$, $\alpha \in (0,1)$, $\kappa_{\mathrm{md}} \in (0,1)$, $\kappa_{\mathrm{ef}} > 0$ and $\kappa_{\mathrm{eg}} > 0$. We also assume that an initial point $\boldsymbol{x}_0$, $\Delta_{\max} > 0$, an initial model $m_o^{\mathrm{icb}}(\boldsymbol{x})$ around $\boldsymbol{x}_0$ and a trust-region radius $\Delta_0^{\mathrm{icb}} \in (0, \Delta_{\max}]$ are given. Let us point out that the model m_k and the trust-region radius Δ_k are set only at the end of the criticality step (Step 1). The iteration ends by defining an incumbent model m_{k+1}^{icb} and an incumbent trust-region radius $\Delta_{k+1}^{\mathrm{icb}}$ for the next iteration, which then might be changed or not by the criticality step.

Algorithm 2.2 Iteration k of a derivative-free trust-region algorithm [8] (without error on function evaluations)

Step 1 (criticality step)

If $\left\|\nabla m_k^{\mathrm{icb}}(\boldsymbol{x}_k)\right\| \leq \varepsilon_{\mathrm{cri}}$ and either m_k^{icb} is not certifiably fully-linear on $B(\boldsymbol{x}_k; \Delta_k^{\mathrm{icb}})$ or $\Delta_k^{\mathrm{icb}} > \mu \left\|\nabla m_k(\boldsymbol{x}_k)\right\|$:

Obtain $\tilde{m}_k$ and $\tilde{\Delta}_k$ by executing algorithm 2.3 (described below).

Set $m_k = \tilde{m}_k$ and $\Delta_k = \min\left\{\max\left\{\tilde{\Delta}_k, \beta\left\|\nabla \tilde{m}_k\right\|\right\}, \Delta_k^{\mathrm{icb}}\right\}$.

Otherwise Set $m_k = m_k^{\mathrm{icb}}$ and $\Delta_k = \Delta_k^{\mathrm{icb}}$.

Step 2: Obtain a step $\boldsymbol{s}_k$ by approximately solving the subproblem (2.1).

Step 3: Evaluate $f(\boldsymbol{x}_k + \boldsymbol{s}_k)$ and compute ρ_k using (2.2).

Step 4: Update the current point according to the ratio ρ_k and quality of the model:

$$\boldsymbol{x}_{k+1} = \begin{cases} \boldsymbol{x}_k + \boldsymbol{s}_k & \text{if } \rho_k \geq \eta_1 \\ \boldsymbol{x}_k + \boldsymbol{s}_k & \text{if } \rho_k > \eta_0 \text{ and } m_k \text{ is fully-linear on } B(\boldsymbol{x}_k; \Delta_k) \\ \boldsymbol{x}_k & \text{else} \end{cases}$$

Step 5: Update trust-region radius according to the ratio ρ_k and quality of the model:

$$\Delta_{k+1}^{\mathrm{icb}} \in \begin{cases} [\Delta_k, \min\{\gamma_{\mathrm{inc}}\Delta_k, \Delta_{\max}\}] & \text{if } \rho_k \geq \eta_1, \\ \{\gamma_{\mathrm{dec}}\Delta_k\} & \text{if } \rho_k < \eta_1 \text{ and } m_k \text{ is fully-linear on } B(\boldsymbol{x}_k; \Delta_k) \\ \{\Delta_k\} & \text{if } \rho_k < \eta_1 \text{ and} \\ & \quad m_k \text{ is not certifiably fully-linear on } B(\boldsymbol{x}_k; \Delta_k). \end{cases}$$

Step 6: Form a new model m_{k+1}^{icb} as follows:

If $\rho_k \geq \eta_1$ or if both $\rho_k \geq \eta_0$ and the model is fully-linear on $B(\boldsymbol{x}_k; \Delta_k)$:

Include the new iterate into the sample set. Form a new model m_{k+1}^{icb}.

Else if $\rho_k < \eta_1$ and m_k is not certifiably fully-linear:

Make a suitable improvement steps. Form a new model m_{k+1}^{icb}.

Else $m_{k+1}^{\mathrm{icb}} = m_k$. ∎

Algorithm 2.3 Final ciriticality subroutine [8]

Step 1: Set $\tilde{m}_k = m_k^{\mathrm{icb}}$, $\tilde{\Delta}_k = \Delta_k^{\mathrm{icb}}$. (initialization)

Step 2: Update $\tilde{m}_k$ so that it is fully-linear on $B(\boldsymbol{x}_k; \tilde{\Delta}_k)$.

Step 3: While $\tilde{\Delta}_k > \mu \|\nabla \tilde{m}_k(\boldsymbol{x}_k)\|$:

Set $\tilde{\Delta}_k \leftarrow \alpha \tilde{\Delta}_k$

Update $\tilde{m}_k$ so that it is fully-linear on $B(\boldsymbol{x}_k; \tilde{\Delta}_k)$. ∎

2.3 Assumptions for the global convergence of Algorithm 2.2 with Algorithm 2.3

In [5], Conn *et al.* proved the global convergence of the Algorithm 2.2 with Algorithm 2.3 to a first-order critical point. Hereafter are the assumptions for the global convergence of the Algorithm 2.2 with Algorithm 2.3.

Assumption 2.1 (assumptions for the global convergence of the Algorithm 2.2 with Algorithm 2.3)

Assumptions on the objective function

(AO1) *The objective function f is bounded below on $\mathcal{L}_{\mathrm{enl}}(\boldsymbol{x}_0)$.*

(AO2) *∇f is Lipschitz continuous on $\mathcal{L}_{\mathrm{enl}}(\boldsymbol{x}_0)$.*

Assumptions on the models

(AM1) *For all k, the model m_k is twice continuously differentiable on $B(\boldsymbol{x}_k; \Delta_k)$.*

(AM2) *For all k, the Hessian of the model m_k is uniformly bounded on $B(\boldsymbol{x}_k; \Delta_k)$.*

Assumption on the approximate solution $\boldsymbol{s}_k$ to the subproblem (2.1)

(AD1) *For all k,*

$$m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k) \geq \kappa_{md} \|\nabla m_k(\boldsymbol{x}_k)\| \min\left[\frac{\|\nabla m_k(\boldsymbol{x}_k)\|}{\|\nabla^2 m_k(\boldsymbol{x}_k)\|}, \Delta_k\right]. \tag{2.6}$$

for some pre-specified constant[*3] *$\kappa_{md} \in (0, 1]$.*

Here, we assume that $\|\nabla m_k(\boldsymbol{x}_k)\| / \left\|\nabla^2 m_k(\boldsymbol{x}_k)\right\| = \infty$ when $\nabla^2 m_k(\boldsymbol{x}_k) = 0$.

[*3] "md" stands for "model decrease".

Assumption on the fully-linear models

(AF1) *If a model m is fully-linear on $B(\boldsymbol{x}_c;\bar{\Delta})$ with respect to some (large enough) constants $\kappa_{\rm ef}$, $\kappa_{\rm eg}$ and for some $\bar{\Delta} \leq \Delta_{\max}$, m is also fully-linear on $B(\boldsymbol{x}_c;\Delta)$, for any $\Delta \in \left[\bar{\Delta}, \Delta_{\max}\right]$, with respect to the same constants $\kappa_{\rm ef}$ and $\kappa_{\rm eg}$.*

(AF2) *For any $\boldsymbol{x}_c \in \mathcal{L}(\boldsymbol{x}_0)$ and $\Delta \in (0, \Delta_{\max}]$, we can obtain a fully-linear model (with respect to $\kappa_{\rm ef}$, $\kappa_{\rm eg}$) on $B(\boldsymbol{x}_c;\Delta)$ in a finite, uniformly bounded (with respect to $\boldsymbol{x}_c$ and Δ) number of steps, say $N_{\max}$.*

∎

Note that, (AO1), (AO2) and (AM1) are classical assumptions for ensuring the global convergence of general trust-region method. While, (AM2) is assumed for simplicity. In fact, it is sufficient to assume that ∇m_k is Lipschitz continuous. Under the assumption (AM2), ∇m_k is Lipschitz continuous on $B(\boldsymbol{x}_k;\Delta_k)$.

In order to achieve global convergence to first-order critical point, (AD1) is essential. That is, the approximate solution $\boldsymbol{s}_k$ to the subproblem (2.1) is required to satisfy (2.6). This condition is the so-called *sufficient decrease condition*. If m_k is linear or quadratic, this condition will be satisfied with $\kappa_{\rm md} = 1/2$ by taking $\boldsymbol{s}_k$ as the Cauchy step[*4]. Indeed, it is unnecessary to actually find the Cauchy step to achieve global convergence. It is sufficient to relate the computed step to the Cauchy step.

If m_k is not linear nor quadratic, we might also calculate an approximate solution of (2.1) which satisfies the sufficient decrease condition (2.6) by using a simple backtracking strategy (for instance, by applying a line search at $\boldsymbol{s} = 0$ along $-\nabla m_k(\boldsymbol{x}_k)$ to the model $m_k(\boldsymbol{x}_k + \boldsymbol{s})$, and stop when (2.6) is satisfied). For more detail, see [3, Section 6.3.3].

We now consider the condition (AF1). If a model is fully-linear on $B(\boldsymbol{x}_c;\bar{\Delta})$ with respect to some constants $\kappa_{\rm ef}$ and $\kappa_{\rm eg}$, and for some $\bar{\Delta} \in (0, \Delta_{\max}]$, the model is not necessarily fully linear with respect to the same constants on $B(\boldsymbol{x}_c;\Delta)$ for any $\Delta \in [\bar{\Delta}, \Delta_{\max}]$. Conn *et al.* [8, Lemma 10.25] give the sufficient condition for ensuring this property. If the model is Lipschitz continuous, we can ensure this sufficient condition by choosing $\kappa_{\rm ef}$ and $\kappa_{\rm eg}$ to be sufficiently large.

Under Assumption 2.1, Conn *et al.* [5] proved the global convergence of Algorithm 2.2 with Algorithm 2.3.

Theorem 2.1 ([5, 8]) *Suppose that Assumption 2.1 holds. Let $\{\Delta_k\}$ and $\{\boldsymbol{x}_k\}$ be sequences generated by Algorithm 2.2 with Algorithm 2.3. Then,*

$$1. \quad \lim_{k\to\infty} \Delta_k = 0,$$
$$2. \quad \lim_{k\to\infty} \nabla f(\boldsymbol{x}_k) = 0.$$

Remark 2.1 *The first statement of Theorem 2.1 gives a natural stopping criterion for Algorithm 2.2. It results from the update of the trust-region radius at the criticalirty step.*

[*4] If we define

$$t_k^C = \operatorname*{argmin}_{t\geq 0;\boldsymbol{x}_k - t\nabla m_k(\boldsymbol{x}_k)\in B(\boldsymbol{x}_k;\Delta_k)} m_k(\boldsymbol{x}_k - t\nabla m_k(\boldsymbol{x}_k)),$$

then the Cauchy step is a step given by

$$\boldsymbol{s}_k^C = -t_k^C \nabla m_k(\boldsymbol{x}_k).$$

3 A globally convergent derivative-free trust-region algorithm using inexact information on function values

In this section, we propose a globally convergent derivative-free trust region algorithm for problem (1.1) in the situation (S0)-(S4). The algorithm is an extended version of Algorithm 2.2 with Algorithm 2.3 by taking into account the case where the objective function evaluation is inexact.

Here, we explicitly express situation of (S1) and (S2) as follows. For a given point $\boldsymbol{x} \in \mathbb{R}^n$ and an accuracy parameter $\varepsilon \geq 0$ of the function evaluation, we denote an estimation value of the objective value as $f_\varepsilon(\boldsymbol{x})$, that is, $f_\varepsilon(\boldsymbol{x})$ satisfies

$$f(\boldsymbol{x}) - \varepsilon \leq f_\varepsilon(\boldsymbol{x}) \leq f(\boldsymbol{x}) + \varepsilon. \tag{3.1}$$

Since there is a trade-off between the function evaluation time and its accuracy, under (S2) and (S3), it would be reasonable to set a lower accuracy level when a point is far from a solution, and set it higher at a point in the neighborhood of a solution for saving computation time. Based on this idea, we will modify Algorithm 2.2 and 2.3 so that ε is updated as large as possible while maintaining their global convergence properties.

First, we will focus on the accuracy of function evaluations in order to construct a fully-linear model, before we describe the proposed algorithm and show its global convergence.

3.1 Fully-linear model based on inexact function evaluations

In this subsection, we consider the accuracy of the evaluated function values in order to construct fully-linear models. Since fully-linear models are defined in a neighborhood of the current iterate $\boldsymbol{x}_c$, we may consider the accuracy of the model around $\boldsymbol{x}_c$.

Let $\mathcal{D}_c$ be a set of displacements $\boldsymbol{d}^i$ from $\boldsymbol{x}_c$ such that the function value at $\boldsymbol{x}_c + \boldsymbol{d}^i$ is evaluated with accuracy ν^i, i.e.,

$$f(\boldsymbol{x}_c + \boldsymbol{d}^i) - \nu^i \leq f_{\nu^i}(\boldsymbol{x}_c + \boldsymbol{d}^i) \leq f(\boldsymbol{x}_c + \boldsymbol{d}^i) + \nu^i, \quad i = 1, \cdots, |\mathcal{D}_c|. \tag{3.2}$$

We assume that there exists at least $n+1$ elements of $\mathcal{D}_c$ such that $\left\|\boldsymbol{d}^i\right\| \leq \Delta$ for some $\Delta > 0$, and we also assume that $\boldsymbol{d}^1 = 0$.

We see that, inequalities (3.2) can be written as

$$f_{\nu^i}(\boldsymbol{x}_c + \boldsymbol{d}^i) - \nu^i \leq f(\boldsymbol{x}_c + \boldsymbol{d}^i) \leq f_{\nu^i}(\boldsymbol{x}_c + \boldsymbol{d}^i) + \nu^i, \quad i = 1, \cdots, |\mathcal{D}_c|,$$

that is, the true objective values $f(\boldsymbol{x}_c + \boldsymbol{d}^i)$ lie in $\left[f_{\varepsilon^i}(\boldsymbol{x}_c + \boldsymbol{d}^i) - \nu^i, f_{\varepsilon^i}(\boldsymbol{x}_c + \boldsymbol{d}^i) + \nu^i\right]$. Hence, we impose the following condition on the model function m.

$$f_{\nu^i}(\boldsymbol{x}_c + \boldsymbol{d}^i) - \nu^i \leq m(\boldsymbol{x}_c + \boldsymbol{d}^i) \leq f_{\nu^i}(\boldsymbol{x}_c + \boldsymbol{d}^i) + \nu^i, \quad i = 1, \cdots, |\mathcal{D}_c|. \tag{3.3}$$

Note that, if $\nu^i = 0$, $(i = 1, \cdots, |\mathcal{D}_c|)$, then (3.3) is the so-called interpolation conditions at $\boldsymbol{x}_c + \boldsymbol{d}^i$, $(i = 1, \cdots, |\mathcal{D}_c|)$.

If the number of elements of $\mathcal{D}_c$ is greater than that of parameters which specify the model function, it is difficult to ensure (3.3) for all points $\boldsymbol{x}_c + \boldsymbol{d}^i$. In that case, we may choose a set $\mathcal{Y} = \left\{\boldsymbol{y}^1 = 0, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^{n+1}\right\}$ of $n+1$ elements of $(B(\boldsymbol{x}_c; \Delta) - \boldsymbol{x}_c) \cap \mathcal{D}_c$, and impose the condition (3.3) only at $\boldsymbol{x}_c + \boldsymbol{y}^i$ with $\boldsymbol{y}^i \in \mathcal{Y}$, that is,

$$f_{\varepsilon^i}(\boldsymbol{x}_c + \boldsymbol{y}^i) - \varepsilon^i \leq m(\boldsymbol{x}_c + \boldsymbol{y}^i) \leq f_{\varepsilon^i}(\boldsymbol{x}_c + \boldsymbol{y}^i) + \varepsilon^i, \quad i = 1, \cdots, n+1, \tag{3.4}$$

where ε^i denotes the accuracy parameter at $\boldsymbol{x}_c + \boldsymbol{y}^i$.

In what follows, we consider the set $\mathcal{Y}$ instead of $\mathcal{D}_c$. We show that, if the $n+1$ points in $\mathcal{Y}$ are "well" affinely independent, and if the accuracy at $\boldsymbol{x}_c + \boldsymbol{y}^i$ satisfies a certain condition, then the model function which satisfies (3.3) is fully-linear on $B(\boldsymbol{x}_c; \Delta)$.

Theorem 3.1 *Suppose that a model function m satisfies (3.4). Moreover, suppose that the following conditions hold for given $\boldsymbol{x}_c \in \mathbb{R}^n$ and $\Delta \in (0, \Delta_{\max}]$.*

- *The objective function f and the model function m are continuously differentiable on $B(\boldsymbol{x}_c; \Delta)$, and that ∇f and ∇m are Lipschitz continuous on $B(\boldsymbol{x}_c; \Delta)$ with Lipschitz constants ℓ_f and ℓ_m respectively.*
- *$\mathcal{Y} = \left\{\boldsymbol{y}^1 = 0, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^{n+1}\right\} \subset (B(\boldsymbol{x}_c; \Delta) - \boldsymbol{x}_c) \cap \mathcal{D}_c$ is affinely independent.*

Let $\varepsilon = \max_{1 \le i \le n+1} \varepsilon^i$, and also let $\Lambda = \left\|Y^{-1}\right\| \Delta$, where Y is a matrix defined by $\left[\boldsymbol{y}^2, \cdots, \boldsymbol{y}^{n+1}\right] \in \mathbb{R}^{n \times n}$. Then, we have the following properties:

- *The error between the gradient of the model and the gradient of the function satisfies*

$$\|\nabla f(\boldsymbol{x}) - \nabla m(\boldsymbol{x})\| \le \frac{5}{2}\sqrt{n}\Lambda\left(\ell_f + \ell_m\right)\Delta + 4\sqrt{n}\varepsilon/\Delta, \quad \forall \boldsymbol{x} \in B(\boldsymbol{x}_c; \Delta), \tag{3.5}$$

  and

- *the error between the model and the function satisfies*

$$|f(\boldsymbol{x}) - m(\boldsymbol{x})| \le \sqrt{n}\left(\ell_f + \ell_m\right)\left(\frac{5}{2}\Lambda + \frac{1}{2}\right)\Delta^2 + 2(2\sqrt{n}\Lambda + 1)\varepsilon, \quad \forall \boldsymbol{x} \in B(\boldsymbol{x}_c; \Delta). \tag{3.6}$$

Proof. First, we define

$$\begin{aligned} \hat{e}_i &:= f(\boldsymbol{x}_c + \boldsymbol{y}^i) - f_{\varepsilon^i}(\boldsymbol{x}_c + \boldsymbol{y}^i), \\ e_i &:= m(\boldsymbol{x}_c + \boldsymbol{y}^i) - f_{\varepsilon^i}(\boldsymbol{x}_c + \boldsymbol{y}^i), \end{aligned}$$

for $i = 1, \cdots, n+1$. From the definition of ε^i and (3.4), the quantities $\hat{e}_i$ and e_i satisfy

$$|\hat{e}_i| \le \varepsilon^i, \quad |e_i| \le \varepsilon^i, \quad (i = 1, \cdots, n+1),$$

respectively. We also define

$$\begin{aligned} \boldsymbol{s} &:= \boldsymbol{x} - \boldsymbol{x}_c, \\ e^g(\boldsymbol{s}) &:= \nabla m(\boldsymbol{x}_c + \boldsymbol{s}) - \nabla f(\boldsymbol{x}_c + \boldsymbol{s}), \\ e^f(\boldsymbol{s}) &:= m(\boldsymbol{x}_c + \boldsymbol{s}) - f(\boldsymbol{x}_c + \boldsymbol{s}). \end{aligned}$$

Note that, the left-hand sides of (3.5) and (3.6) are written as $\|e^g(\boldsymbol{s})\|$ and $|e^f(\boldsymbol{s})|$, respectively. Hence, we evaluate the upper bounds of $\|e^g(\boldsymbol{s})\|$ and $|e^f(\boldsymbol{s})|$ on $B(0, \Delta)$.

A first-order Taylor expansions around $\boldsymbol{x}_c + \boldsymbol{s}$ for both $f(\boldsymbol{x}_c + \boldsymbol{y}^i)$ and $m(\boldsymbol{x}_c + \boldsymbol{y}^i)$ give

$$f(\boldsymbol{x}_c + \boldsymbol{y}^i) = f(\boldsymbol{x}_c + \boldsymbol{s}) + \langle \nabla f(\boldsymbol{x}_c + \boldsymbol{s}), \boldsymbol{y}_i - \boldsymbol{s} \rangle + \int_0^1 \langle \nabla f\left(\boldsymbol{x}_c + \boldsymbol{s} + t(\boldsymbol{y}^i - \boldsymbol{s})\right) - \nabla f(\boldsymbol{x}_c + \boldsymbol{s}), \boldsymbol{y}^i - \boldsymbol{s} \rangle dt,$$

$$m(\boldsymbol{x}_c + \boldsymbol{y}^i) = m(\boldsymbol{x}_c + \boldsymbol{s}) + \langle \nabla m(\boldsymbol{x}_c + \boldsymbol{s}), \boldsymbol{y}_i - \boldsymbol{s} \rangle + \int_0^1 \langle \nabla m\left(\boldsymbol{x}_c + \boldsymbol{s} + t(\boldsymbol{y}^i - \boldsymbol{s})\right) - \nabla m(\boldsymbol{x}_c + \boldsymbol{s}), \boldsymbol{y}^i - \boldsymbol{s} \rangle dt,$$

for $i = 1, \cdots, n+1$, where $\langle \cdot, \cdot \rangle$ denotes the standard inner product on $\mathbb{R}^n$. Using these equalities, we have

$$\begin{aligned}
\langle e^g(\boldsymbol{s}), \boldsymbol{y}^i - \boldsymbol{s} \rangle =& \langle \nabla m(\boldsymbol{x}_c + \boldsymbol{s}) - \nabla f(\boldsymbol{x}_c + \boldsymbol{s}), \boldsymbol{y}^i - \boldsymbol{s} \rangle \\
=& \int_0^1 \langle \nabla f\left(\boldsymbol{x}_c + \boldsymbol{s} + t(\boldsymbol{y}^i - \boldsymbol{s})\right) - \nabla f(\boldsymbol{x}_c + \boldsymbol{s}), \boldsymbol{y}^i - \boldsymbol{s} \rangle dt \\
& - \int_0^1 \langle \nabla m\left(\boldsymbol{x}_c + \boldsymbol{s} + t(\boldsymbol{y}^i - \boldsymbol{s})\right) - \nabla m(\boldsymbol{x}_c + \boldsymbol{s}), \boldsymbol{y}^i - \boldsymbol{s} \rangle dt \\
& + m(\boldsymbol{x}_c + \boldsymbol{y}^i) - f_{\varepsilon^i}(\boldsymbol{x}_c + \boldsymbol{y}^i) + f_{\varepsilon^i}(\boldsymbol{x}_c + \boldsymbol{y}^i) - f(\boldsymbol{x}_c + \boldsymbol{y}^i) \\
& - (m(\boldsymbol{x}_c + \boldsymbol{s}) - f(\boldsymbol{x}_c + \boldsymbol{s})) \\
=& \int_0^1 \langle \nabla f\left(\boldsymbol{x}_c + \boldsymbol{s} + t(\boldsymbol{y}^i - \boldsymbol{s})\right) - \nabla f(\boldsymbol{x}_c + \boldsymbol{s}), \boldsymbol{y}^i - \boldsymbol{s} \rangle dt \\
& - \int_0^1 \langle \nabla m\left(\boldsymbol{x}_c + \boldsymbol{s} + t(\boldsymbol{y}^i - \boldsymbol{s})\right) - \nabla m(\boldsymbol{x}_c + \boldsymbol{s}), \boldsymbol{y}^i - \boldsymbol{s} \rangle dt + e_i - \hat{e}_i - e^f(\boldsymbol{s}), \qquad (3.7)
\end{aligned}$$

for $i = 1, \cdots, n+1$. Since $\boldsymbol{y}^1 = 0$, we have

$$\begin{aligned}
\langle e^g(\boldsymbol{s}), -\boldsymbol{s} \rangle =& \int_0^1 \langle \nabla f\left(\boldsymbol{x}_c + \boldsymbol{s} - t\boldsymbol{s}\right) - \nabla f(\boldsymbol{x}_c + \boldsymbol{s}), -\boldsymbol{s} \rangle dt \\
& - \int_0^1 \langle \nabla m\left(\boldsymbol{x}_c + \boldsymbol{s} - t\boldsymbol{s}\right)) - \nabla m(\boldsymbol{x}_c + \boldsymbol{s}), -\boldsymbol{s} \rangle dt + e_1 - \hat{e}_1 - e^f(\boldsymbol{s}). \qquad (3.8)
\end{aligned}$$

Subtracting (3.8) from (3.7) yields

$$\begin{aligned}
\langle e^g(\boldsymbol{s}), \boldsymbol{y}^i \rangle =& \int_0^1 \langle \nabla f\left(\boldsymbol{x}_c + \boldsymbol{s} + t(\boldsymbol{y}^i - \boldsymbol{s})\right) - \nabla f(\boldsymbol{x}_c + \boldsymbol{s}), \boldsymbol{y}^i - \boldsymbol{s} \rangle dt \\
& - \int_0^1 \langle \nabla m\left(\boldsymbol{x}_c + \boldsymbol{s} + t(\boldsymbol{y}^i - \boldsymbol{s})\right) - \nabla m(\boldsymbol{x}_c + \boldsymbol{s}), \boldsymbol{y}^i - \boldsymbol{s} \rangle dt \\
& - \int_0^1 \langle \nabla f\left(\boldsymbol{x}_c + \boldsymbol{s} - t\boldsymbol{s}\right) - \nabla f(\boldsymbol{x}_c + \boldsymbol{s}), -\boldsymbol{s} \rangle dt \\
& + \int_0^1 \langle \nabla m\left(\boldsymbol{x}_c + \boldsymbol{s} - t\boldsymbol{s}\right)) - \nabla m(\boldsymbol{x}_c + \boldsymbol{s}), -\boldsymbol{s} \rangle dt + (e_i - e_1) - (\hat{e}_i - \hat{e}_1), \qquad (3.9)
\end{aligned}$$

for each $\boldsymbol{y}^i$, $i = 2, \cdots, n+1$.

First, we consider the first right-hand side term of (3.9). For any $\boldsymbol{s} \in B(\boldsymbol{x}_c; \Delta) - \boldsymbol{x}_c$, we have

$$\begin{aligned}
& \left| \int_0^1 \langle \nabla f\left(\boldsymbol{x}_c + \boldsymbol{s} + t(\boldsymbol{y}^i - \boldsymbol{s})\right) - \nabla f(\boldsymbol{x}_c + \boldsymbol{s}), \boldsymbol{y}^i - \boldsymbol{s} \rangle dt \right| \\
& \qquad \leq \int_0^1 \left\| \nabla f\left(\boldsymbol{x}_c + \boldsymbol{s} + t(\boldsymbol{y}^i - \boldsymbol{s})\right) - \nabla f(\boldsymbol{x}_c + \boldsymbol{s}) \right\| \left\| \boldsymbol{y}^i - \boldsymbol{s} \right\| dt \\
& \qquad \leq \int_0^1 \ell_f \left\| \boldsymbol{y}^i - \boldsymbol{s} \right\|^2 t dt \\
& \qquad \leq 2\ell_f \Delta^2, \qquad (3.10)
\end{aligned}$$

where the second inequality follows from the Lipschitz continuity of ∇f, and the last inequality follows from the fact that $\left\| \boldsymbol{y}^i - \boldsymbol{s} \right\|^2 \leq 4\Delta^2$ for $\boldsymbol{s}$ and $\boldsymbol{y}^i \in B(\boldsymbol{x}_c; \Delta) - \boldsymbol{x}_c$. Similarly, we obtain

$$\begin{aligned}
|\text{the second right-hand side term of (3.9)}| &\leq 2\ell_m \Delta^2, \\
|\text{the third right-hand side term of (3.9)}| &\leq \frac{1}{2}\ell_f \Delta^2, \qquad (3.11) \\
|\text{the fourth right-hand side term of (3.9)}| &\leq \frac{1}{2}\ell_m \Delta^2.
\end{aligned}$$

On the other hand, it holds that

$$|(e_i - e_1) - (\hat{e}_i - \hat{e}_1)| \leq \varepsilon_1 + \varepsilon_i + \varepsilon_1 + \varepsilon_i \leq 4\varepsilon,$$

and

$$\left\|Y^T e^g(\boldsymbol{s})\right\|_\infty = \max_{1\leq i\leq n+1} \langle e^g(\boldsymbol{s}), \boldsymbol{y}^i\rangle.$$

It then follows from (3.9)-(3.11) that

$$\left\|Y^T e^g(\boldsymbol{s})\right\| \leq \sqrt{n}\left\|Y^T e^g(\boldsymbol{s})\right\|_\infty \leq \sqrt{n}\left(\frac{5}{2}\ell_f\Delta^2 + \frac{5}{2}\ell_m\Delta^2 + 4\varepsilon\right).$$

Since $\left\|Y^{-T}\right\| = \left\|Y^{-1}\right\| \leq \frac{\Lambda}{\Delta}$ (from the definition of Λ), it follows that

$$\|e^g(\boldsymbol{s})\| \leq \left\|Y^{-T}\right\|\left\|Y^T e^g(\boldsymbol{s})\right\| \leq \frac{5}{2}\sqrt{n}\Lambda(\ell_f + \ell_m)\Delta + 4\sqrt{n}\Lambda\varepsilon/\Delta,$$

which is the desired inequality (3.5).

Finally, using this inequality and (3.8), we obtain

$$\begin{aligned}\left|e^f(\boldsymbol{s})\right| &\leq \|e^g(\boldsymbol{s})\|\,\|\boldsymbol{s}\| + \frac{1}{2}\sqrt{n}\left(\ell_f\Delta^2 + \ell_m\Delta^2\right) + |e_1 + \hat{e}_1| \\ &\leq \sqrt{n}(\ell_f + \ell_m)\left(\frac{5}{2}\Lambda + \frac{1}{2}\right)\Delta^2 + 2\left(2\sqrt{n}\Lambda + 1\right)\varepsilon\end{aligned}$$

This complete the proof. ∎

From Theorem 3.1, if the accuracies at ε^i $\boldsymbol{x}_c + \boldsymbol{y}^i$ $(i = 1, \cdots, n)$ satisfy

$$\varepsilon = \max_{1\leq i\leq n+1} \varepsilon^i \leq \Gamma\Delta^2, \tag{3.12}$$

for some constant $\Gamma \geq 0$, then the model function satisfies

$$\|\nabla f(\boldsymbol{x}) - \nabla m(\boldsymbol{x})\| \leq \left(\frac{5}{2}\sqrt{n}\Lambda\left(\ell_f + \ell_m\right) + 4\sqrt{n}\Lambda\Gamma\right)\Delta, \tag{3.13}$$

$$|f(\boldsymbol{x}) - m(\boldsymbol{x})| \leq \left(\sqrt{n}\left(\ell_f + \ell_m\right)\left(\frac{5}{2}\Lambda + \frac{1}{2}\right) + 2(2\sqrt{n}\Lambda + 1)\Gamma\right)\Delta^2, \tag{3.14}$$

for all $\boldsymbol{x} \in B(\boldsymbol{x}_c; \Delta)$. Moreover, if the coefficient of Δ in (3.13) is less than or equal to κ_{eg}, and if the coefficient of Δ^2 in (3.14) is less than or equal to $\kappa_{\rm ef}$, then the model function m is fully-linear on $B(\boldsymbol{x}_c; \Delta)$ with respect to $\kappa_{\rm ef}$ and $\kappa_{\rm eg}$. The following corollary states this fact.

Corollary 3.2 *Suppose that all the assumptions of the Theorem 3.1 are all satisfied, In addition, the following conditions are assumed.*

- *∇f is Lipschitz continuous on $\mathcal{L}_{enl}(\boldsymbol{x}_0)$ with Lipshitz constant L_f.*
- *The Lipschitz constant ℓ_m of ∇m on $B(\boldsymbol{x}_c, \Delta)$ is bounded by a constant L, independent of the choice of $\boldsymbol{x}_c \in \mathcal{L}_{\rm enl}(\boldsymbol{x}_0)$ and $\Delta \in (0, \Delta_{\max}]$.*
- *For given constant $\Gamma \geq 0$, the accuracy ε^i of the evaluated function value at $\boldsymbol{x}_c + \boldsymbol{y}^i$ satisfy $\max_{1\leq i\leq n+1} \varepsilon^i \leq \Gamma$.*

Then, the model m is fully-linear on $B(\boldsymbol{x}_c; \Delta)$ with respect to some contants $\kappa_{\rm ef}$ and $\kappa_{\rm eg}$ such that

$$\begin{aligned}\kappa_{\rm eg} &\geq \frac{5}{2}\sqrt{n}\Lambda\left(L_f + L\right) + 4\sqrt{n}\Lambda\Gamma, \\ \kappa_{\rm ef} &\geq \sqrt{n}\left(L_f + L\right)\left(\frac{5}{2}\Lambda + \frac{1}{2}\right) + 2(2\sqrt{n}\Lambda + 1)\Gamma.\end{aligned}$$

Here, we discuss how to satisfy the assumptions of Theorem 3.1 and Corollary 3.2. Among these assumptions, the uniformly boundedness of ℓ_m seems difficult to hold. When we consider quadratic model functions, this assumption is satisfied if the models' Hessian are uniformly bounded. We will show how to build a model function which satisfy (3.4) and whose Hessian is bounded in Section 4. By the way, in order to make fully-linear models, Λ must be bounded above for a constant independent of Y, and Δ. The next lemma gives sufficient conditions on the set $\mathcal{Y}$ under which $\Lambda = \left\|Y^{-1}\right\| \Delta$ is bounded.

Lemma 3.3 ([19, Lemma 4.2])
Let $QR = \frac{1}{\Delta}Y$ denote a QR factorization of a matrix $\frac{1}{\Delta}Y$ whose columns satisfy $\left\|\frac{Y_j}{\Delta}\right\| \leq 1, j = 1, \cdots, n$. If $|R_{ii}| \geq \theta > 0$ for $i = 1, \cdots, n$, then $\left\|Y^{-1}\right\| \leq \frac{\bar{\Lambda}}{\Delta}$ for a constant $\bar{\Lambda}$ depending only on n and θ:

$$\bar{\Lambda} = n^{\frac{n-1}{2}}\theta^{-n}. \tag{3.15}$$

Based on this lemma, Wild *et al.* [17, 19] proposed the algorithm of generating a set $\mathcal{Y}$ of such points from a given set of candidates $\mathcal{D}_c$. The algorithm can be found in Appendix A.

3.2 Proposed derivative-free trust-region algorithm with dynamic accuracy on function evaluation

In this subsection, we propose a globally convergent derivative-free trust-region algorithm using inexact information on function values. The algorithm is a modified version of Algorithm 2.2 and Algorithm 2.3. Especially, the proposed algorithm updates the accuracy of function evaluations dynamically at each iteration.

First, we recall that Algorithm 2.2 relies on a ratio ρ_k, which is defined by using $f(\boldsymbol{x}_k)$ and $f(\boldsymbol{x}_k + \boldsymbol{s}_k)$. (See (2.2)). Thus, we need to define the accuracy parameters at the current point $\boldsymbol{x}_k$ and at the trial point $\boldsymbol{x}_k + \boldsymbol{s}_k$. We denote the accuracy parameter at $\boldsymbol{x}_k$ by $\varepsilon_k^{\mathrm{cur}}$, and the accuracy parameter at $\boldsymbol{x}_k + \boldsymbol{s}_k$ by $\varepsilon_k^{\mathrm{tri}}$. Then, from (3.1),

$$f(\boldsymbol{x}_k) - \varepsilon_k^{\mathrm{cur}} \leq f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) \leq f(\boldsymbol{x}_k) + \varepsilon_k^{\mathrm{cur}},$$

$$f(\boldsymbol{x}_k + \boldsymbol{s}_k) - \varepsilon_k^{\mathrm{tri}} \leq f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_k + \boldsymbol{s}_k) \leq f(\boldsymbol{x}_k + \boldsymbol{s}_k) + \varepsilon_k^{\mathrm{tri}}.$$

When we construct fully-linear models based on inexact function values, Corollary (3.2) suggests that $\varepsilon_k^{\mathrm{cur}}$ and $\varepsilon_k^{\mathrm{tri}}$ should be updated so that

$$\max\left\{\varepsilon_k^{\mathrm{cur}}, \varepsilon_k^{\mathrm{tri}}\right\} \leq C\Delta_k^2 \quad \text{for all } k, \tag{3.16}$$

where $C \leq \Gamma$ is a nonnegative constant.

The proposed Algorithm 3.1, described below, updates the current point $\boldsymbol{x}_k$, the trust-region radius Δ_k and the model function m_k, according to the ratio ρ_k and the quality of the model m_k. The updates of the proposed algorithm categorized into the following three types. For each type of updating, we adopt the same terminology in [8].

1. $\rho_k \geq \eta_1$;
   in this case, the new iterate is accepted and the trust-region radius is retained or increased. Such iterations are called **successful**. We will denote the set of indices of all successful iterations by $\mathcal{S}$.
2. $\rho_k < \eta_1$ and m_k is not certifiably fully-linear on $B(\boldsymbol{x}_k; \Delta_k)$ with respect to κ_{ef} and κ_{eg};
   in this case, the model is improved, and the trust-region radius is retained. Such iterations are called **model-improving**.

3. $\rho_k \leq \eta_1$ and m_k is fully-linear on $B(\boldsymbol{x}_k; \Delta_k)$ with respect to κ_{ef} and κ_{eg};
   in this case, no decrease was obtained and there is no need to improve the model. In addition, the trust-region is reduced. Such iterations are called **unsuccessful**.

We now formally state our proposed algorithm (derivative-free trust-region algorithm) with dynamic accuracy on function evaluations. Here, we suppose that the constants for standard derivative-free trust-region are given and satisfy the conditions $0 \leq \eta_1 < 1$, $0 < \gamma_{\text{dec}} < 1 < \gamma_{\text{dec}}$, $\Delta_{\max} > 0$, $\kappa_{\text{md}} \in (0, 1)$, $\kappa_{\text{ef}} > 0$, $\kappa_{\text{eg}} > 0$, $\varepsilon_{\text{cri}} > 0$, $\mu > \beta > 0$, $\alpha \in (0, 1)$. We also suppose that the additional constants C, M, ζ satisfy $C \geq 0, M \geq 0, \zeta \in (0, 1)$. To simplify the notation, we denote the gradient and Hessian of the model m_k at $\boldsymbol{x}_k$ by $\boldsymbol{g}_k$ and $\boldsymbol{H}_k$ respectively. That means $\boldsymbol{g}_k = \nabla m_k(\boldsymbol{x}_k)$ and $\boldsymbol{H}_k = \nabla^2 m_k(\boldsymbol{x}_k)$. We note that, the true value of the objective function at $\boldsymbol{x}_k$, $f(\boldsymbol{x}_k)$, never appears in the algorithm.

Algorithm 3.1
Derivative-free trust-region algorithm with dynamic accuracy on function evaluations

---

Step 0 (initialization):

Choose an algorithm for constructing fully-linear models using inexact function values.

Choose an initial point $\boldsymbol{x}_0$, and an initial trust region radius $\Delta_0^{\text{icb}} \in (0, \Delta_{\max}]$.

Give the accuracy parameter at the initial point, $\varepsilon_0^{\text{cur}}$, which satisfies $\varepsilon_0^{\text{cur}} \leq \min\left\{C\left(\Delta_0^{\text{icb}}\right)^2, M\right\}$.

Choose an initial model $m_0^{\text{icb}}(\boldsymbol{x})$ around $\boldsymbol{x}_0$ (with gradient and possibly Hessian at $\boldsymbol{x} = \boldsymbol{x}_0$ given by $\boldsymbol{g}_0^{\text{icb}}$ and $\boldsymbol{H}_0^{\text{icb}}$, respectively.)

Set $k = 0$.

Step 1 (criticality step):

If $\left\|\boldsymbol{g}_k^{\text{icb}}\right\| \leq \varepsilon_{\text{cri}}$ and either m_k^{icb} is not certifiably fully-linear on $B(\boldsymbol{x}_k; \Delta_k^{\text{icb}})$ or $\Delta_k^{\text{icb}} > \mu \left\|\boldsymbol{g}_k^{\text{icb}}\right\|$:

Obtain $\tilde{m}_k$ and $\tilde{\Delta}_k$ by executing Algorithm 3.2.

Set $m_k = \tilde{m}_k$ and $\Delta_k = \min\left\{\max\left\{\tilde{\Delta}_k, \beta\left\|\tilde{\boldsymbol{g}}_k\right\|\right\}, \Delta_k^{\text{icb}}\right\}$, Note that the accuracy parameter of the current point ($\varepsilon_k^{\text{cur}}$) might have been updated. The resulting accuracy must satisfy $\varepsilon_k^{\text{cur}} \leq \min\left\{C\left(\tilde{\Delta}_k^{\text{icb}}\right)^2, M\zeta^k\right\}$.

Otherwise Set $m_k = m_k^{\text{icb}}$ and $\Delta_k = \Delta_k^{\text{icb}}$.

Step 2: Set $\varepsilon_k^{\text{tri}} \in \left[0, \min\left\{C\Delta_k^2, M\zeta^{k+1}\right\}\right]$.

Step 3: Obtain a step $\boldsymbol{s}_k$ by solving the subproblem (2.1) exactly or using any approximation technique.

Step 4: Evaluate $f_{\varepsilon_k^{\text{tri}}}(\boldsymbol{x}_k + \boldsymbol{s}_k)$ and calculate

$$\rho_k = \frac{f_{\varepsilon_k^{\text{cur}}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\text{tri}}}(\boldsymbol{x}_k + \boldsymbol{s}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}. \tag{3.17}$$

Step 5: Update the current-point according to the ratio ρ_k and quality of the model:

$$\boldsymbol{x}_{k+1} = \begin{cases} \boldsymbol{x}_k + \boldsymbol{s}_k & \text{if iteration } k \text{ is } \textbf{successful}, \\ \boldsymbol{x}_k & \text{if iteration } k \text{ is } \textbf{unsuccessful} \text{ or } \textbf{model-improving}. \end{cases}$$

Step 6: Update trust-region radius according to the ration ρ_k and quality of the model:

$$\Delta_{k+1}^{\text{icb}} \in \begin{cases} [\Delta_k, \min\{\gamma_{\text{inc}}\Delta_k, \Delta_{\max}\}] & \text{if iteration } k \text{ is } \textbf{successful}, \\ \{\gamma_{\text{dec}}\Delta_k\} & \text{if iteration } k \text{ is } \textbf{unsuccessful}, \\ \{\Delta_k\} & \text{if iteration } k \text{ is } \textbf{model-improving}, \end{cases}$$

Step 7: Update $\varepsilon_k^{\mathrm{cur}}$ as follows:

$$\text{Set} \quad \varepsilon_{k+1}^{\mathrm{cur}} \in \begin{cases} \left\{\varepsilon_k^{\mathrm{tri}}\right\} & \text{if iteration } k \text{ is } \textbf{successful}, \\ \left[0, \min\left\{\varepsilon_k^{\mathrm{cur}}, C\left(\Delta_{k+1}^{\mathrm{icb}}\right)^2, M\zeta^{k+1}\right\}\right] & \text{if iteration } k \text{ is } \textbf{unsuccessful} \text{ or } \textbf{model-improving}. \end{cases}$$

If $\varepsilon_{k+1}^{\mathrm{cur}} < \varepsilon_k^{\mathrm{tri}}$, reevaluate $f_{\varepsilon_{k+1}^{\mathrm{cur}}}(\boldsymbol{x}_{k+1})$.

Step 8 (model building) :

If iteration k is **successful**:

Form a new model taking into consideration the new iterate. Define m_{k+1}^{icb} to be the resulting model.

Else if iteration k is **unsuccessful**:

If $\varepsilon_{k+1}^{\mathrm{cur}}$ was updated in Step 7 (i,e., $\varepsilon_{k+1}^{\mathrm{cur}} < \varepsilon_k^{\mathrm{tri}}$): ,

Form a new model taking into consideration the re-evaluated function value $(f_{\varepsilon_{k+1}^{\mathrm{cur}}}(\boldsymbol{x}_{k+1}))$. Define m_{k+1}^{icb} to be the resulting model.

Otherwise Set $m_{k+1}^{\mathrm{icb}} := m_k$.

Else (iteration k is **model-improving**):

Form a new model, which is fully-linear on $B(\boldsymbol{x}_{k+1}; \Delta_{k+1}^{\mathrm{icb}})$. Define m_{k+1}^{icb} to be the improved model.

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

The procedure invoked in the criticality step (Step 1 of Algorithm 3.1) is described in the following algorithm. The algorithm is almost the same one in [5], except for updating $\varepsilon_k^{\mathrm{cur}}$.

Algorithm 3.2 Final criticality subroutine c.f. [5]

Note that, this algorithm is applied only if $\left\|\boldsymbol{g}_k^{\mathrm{icb}}\right\| \leq \epsilon_{\mathrm{cri}}$ and at least one of the following conditions holds: the model m_k^{icb} is not certifiably fully-linear on $B(\boldsymbol{x}_k; \Delta_k^{\mathrm{icb}})$ or $\Delta_k^{\mathrm{icb}} > \mu\left\|\boldsymbol{g}_k^{\mathrm{icb}}\right\|$. The constant $\alpha \in (0,1)$ is chosen at Step 0 of Algorithm 3.1.

Step 1: Set $i = 0$. Set $m_k^{(0)} = m_k^{\mathrm{icb}}$.

Step 2: Repeat

Set $i \leftarrow i+1$.

If $\varepsilon_k^{\mathrm{cur}} > C\left(\alpha^{i-1}\Delta_k^{\mathrm{icb}}\right)^2$:

Set $\varepsilon_k^{\mathrm{cur}} \in \left[0, C\left(\alpha^{i-1}\Delta_k^{\mathrm{icb}}\right)^2\right]$ and evaluate $f_{\varepsilon_{k+1}^{\mathrm{cur}}}(\boldsymbol{x}_k)$

Update the previous model $m_k^{(i-1)}$ so that it is fully-linear on $B(\boldsymbol{x}_k; \alpha^{i-1}\Delta_k^{\mathrm{icb}})$.

Denote the new model by $m_k^{(i)}$.

Set $\tilde{\Delta}_k = \alpha^{i-1}\Delta_k^{\mathrm{icb}}$ and $\tilde{m}_k = m_k^{(i)}$.

Until $\tilde{\Delta}_k \leq \mu\left\|\boldsymbol{g}_k^{(i)}\right\|$. ∎

Algorithm 3.1 is a modified version of Algorithm 2.2 with $\eta_0 = \eta_1$. Main changes from Algorithm 2.2 are the following two things.

a) Added a mechanism of updating accuracy parameters.

b) Modified updating rule of model function.

We explain their details below.

First, we discuss the mechianism of updating accuracy parameters. The accuracy parameters $\varepsilon_k^{\rm cur}$ and $\varepsilon_k^{\rm tri}$ are updated so that (3.16) are satisfied (see Lemma 3.4 below). If the parameters satisfy only (3.16) and Δ_k does not tend to zero , then $\varepsilon_k^{\rm cur}$ and $\varepsilon_k^{\rm tri}$ might be still large. However, when we cannot sufficiently reduce the (inexact) objective function value, we need to reduce $\varepsilon_k^{\rm cur}$ and $\varepsilon_k^{\rm tri}$.

In order to ensure that

$$\varepsilon_k^{\rm cur} \to 0, \quad \varepsilon_k^{\rm tri} \to 0 \qquad (k \to \infty), \tag{3.18}$$

Algorithm 3.1 updates $\varepsilon_k^{\rm cur}$ and $\varepsilon_k^{\rm tri}$ so that

$$\varepsilon_k^{\rm cur} \leq M\zeta^k, \quad \varepsilon_k^{\rm tri} \leq M\zeta^{k+1}, \quad \zeta \in (0,1), \tag{3.19}$$

at each iteration (see Lemma 3.4 below). From some pilot experiments, we have observed that Δ_k tends to zero as Algorithm 2.2. Hence, (3.18) might be guaranteed by only imposing (3.16). However, at this stage, we need the condition (3.19) to establish the global convergence in the next subsection.

When we reduce $\varepsilon_k^{\rm cur}$ and $\varepsilon_k^{\rm tri}$, we need to calculate the objective function value with higher accuracy at the same point. Fortunately, for some applications, we can evaluate $f_\varepsilon(\boldsymbol{x})$ by using some products generated in the previous evaluations. For example, consider the case where $f_\varepsilon(\boldsymbol{x})$ is computed by some convergent iterative algorithm. Then, we may simply restart the algorithm from the last iteration of the previous evaluation in order to get the function value with higher accuracy. Hence, the reduction of $\varepsilon_k^{\rm cur}$ and $\varepsilon_k^{\rm tri}$ does not necessarily lead to a large computational expense.

Now, we explain the change b), that is, updating rule of the current model in Step 8. If the k-th iteration is model-improving, Algorithm 3.1 surely builds a fully-linear model. Practically, we do not need to build a fully-linear model at each model-improving iteration. However, in this paper, we let Algorithm 3.1 build a fully-linear model at each model-improving iteration in order to simplify the proof of the global convergence.

Hereafter, we list the remarks of the final critical subroutine (Algorithm 3.2).

- If $\left\|\boldsymbol{g}_k^{\rm icb}\right\| \leq \varepsilon_{\rm cri}$ holds and Algorithm 3.2 is invoked, then $\Delta_k^{\rm icb}$ is shrank. Hence, $\varepsilon_k^{\rm cur}$ has to be reduced in order to ensure (3.16). Algorithm 3.2 has such mechanism (Step 2).
- At the end of the criticality step, the model m_k is fully-linear on $B(\boldsymbol{x}_k; \tilde{\Delta}_k)$. it then follows that

  $$\|\nabla f(\boldsymbol{x}) - \boldsymbol{g}_k\| \leq \kappa_{\rm eg}\tilde{\Delta}_k \leq \kappa_{\rm eg}\mu \|\boldsymbol{g}_k\|$$

  from (2.4) and stopping criterion of Algorithm 3.2. Moreover, if (AF1) of assumption 2.1 holds, then m_k is also fully-linear on $B(\boldsymbol{x}_k; \Delta_k)$ (as well as on $B(\boldsymbol{x}_k; \mu\|\boldsymbol{g}_k\|)$).
- In Section 3.3, we will prove that, Algorithm 3.2 terminates after a finite number of steps if $\|\nabla f(\boldsymbol{x}_k)\| \neq 0$. If $\|\nabla f(\boldsymbol{x}_k)\| \neq 0$, then we will cycle in the criticality step until some stopping criterion is met. Thus, Algorithm 3.1 with Algorithm 3.2 is well-defined.

Finally, we prove that $\varepsilon_k^{\rm cur}$ and $\varepsilon_k^{\rm tri}$ satisfy (3.16) and (3.19) at each iteration.

Lemma 3.4 *The accuracy parameters $\varepsilon_k^{\rm cur}$ and ε_k^{tri} updated by Algorithm 3.1 with Algorithm 3.2 satisfy*

$$\varepsilon_k^{\rm cur} \leq C\Delta_k^2, \quad \varepsilon_k^{\rm cur} \leq M\zeta^k, \tag{3.20}$$

$$\varepsilon_k^{\rm tri} \leq C\Delta_k^2, \quad \varepsilon_k^{\rm tri} \leq M\zeta^{k+1}, \tag{3.21}$$

for all k.

Proof. The inequality (3.21) directly follows from the update rule of $\varepsilon_k^{\rm tri}$ in Step 2 of Algorithm 3.1. Thus, we show (3.20). From the mechanism of Step 7 of Algorithm 3.1, we have the following properties of $\varepsilon_{k+1}^{\rm cur}$ at the beginning of the $k+1$-th iteration. If the k-th iteration is successful, then

$$\varepsilon_{k+1}^{\rm cur} = \varepsilon_k^{\rm tri} \leq \min\left\{C\Delta_k^2, M\zeta^{k+1}\right\} \leq \min\left\{C(\Delta_{k+1}^{\rm icb})^2, M\zeta^{k+1}\right\}.$$

Otherwise, that is, the k-th iteration is unsuccessful or model-improving, then

$$\varepsilon_{k+1}^{\rm cur} \leq \min\left\{C(\Delta_{k+1}^{\rm icb})^2, M\zeta^{k+1}\right\}.$$

Thus, if the criticality step is not invoked at Step1 of Algorithm 3.1 in the $k+1$-iterate, then we have

$$\varepsilon_{k+1}^{\rm cur} \leq \min\left\{C\Delta_{k+1}^2, M\zeta^{k+1}\right\}.$$

Otherwise, we obtain

$$\varepsilon_{k+1}^{\rm cur} \leq \min\left\{C\tilde{\Delta}_{k+1}^2, M\zeta^{k+1}\right\} \leq \min\left\{C\Delta_{k+1}^2, M\zeta^{k+1}\right\},$$

from Step 1 of Algorithm 3.1. Therefore, it is deduced that

$$\varepsilon_k^{\rm cur} \leq C\Delta_k^2, \quad \varepsilon_k^{\rm cur} \leq M\zeta^k \quad \text{for all } k.$$

∎

3.3 Global convergence of the proposed method for first-order critical points.

In this subsection, we establish the global convergence of the proposed method. We need the following assumptions for the global convergence of the Algorithm 3.1 with Algorithm 3.2.

Assumption 3.1 (assumptions for the global convergence of the Algorithm 3.1 with 3.2)

Assumptions on the objective function

(AO1) *The objective function f is bounded below on $\mathcal{L}_{\rm enl}(\boldsymbol{x}_0)$, that is, there exists a constant κ_* such that $f(\boldsymbol{x}) \geq \kappa_*$ for all $\boldsymbol{x} \in \mathcal{L}_{\rm enl}(\boldsymbol{x}_0)$.*

(AO2) *∇f is Lipschitz continuous on $\mathcal{L}_{\rm enl}(\boldsymbol{x}_0)$.*

Assumptions on the models

(AM1) *For all k, the model m_k is twice continuously differentiable on $B(\boldsymbol{x}_k; \Delta_k)$.*

(AM2) *For all k, the Hessian of the model m_k is uniformly bounded on $B(\boldsymbol{x}_k; \Delta_k)$. i.e., there exists a constant*[*5] *$\kappa_{\rm bhm} > 0$ such that*

$$\max_{\boldsymbol{x} \in B(\boldsymbol{x}_k; \Delta_k)} \left\|\nabla^2 m_k(\boldsymbol{x})\right\| \leq \kappa_{\rm bhm} \quad \textit{for all } k. \tag{3.22}$$

Additional assumption on the models

(AM3) *the error between the model and function at the current point satisfies*

$$\left|m_k(\boldsymbol{x}_k) - f_{\varepsilon_k^{\rm cur}}(\boldsymbol{x}_k)\right| \leq \varepsilon_k^{\rm cur}. \tag{3.23}$$

Assumption on the approximate solution $\boldsymbol{s}_k$ to the subproblem (2.1)

[*5] "bhm" stands for "bound on the Hessian of the models".

(AD1) *For all k,*

$$m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k) \geq \kappa_{md} \|\boldsymbol{g}_k\| \min\left[\frac{\|\boldsymbol{g}_k\|}{\|\boldsymbol{H}_k\|}, \Delta_k\right], \tag{3.24}$$

for some pre-specified constant[*6] $\kappa_{md} \in (0, 1]$.

Here, we assume that $\|\boldsymbol{g}_k\| / \|\boldsymbol{H}_k\| = \infty$ when $\boldsymbol{H}_k = 0$.

Assumptiona on the fully-linear models

(AF1) *If a model m is fully-linear on $B(\boldsymbol{x}_c; \bar{\Delta})$ with respect to some (large enough) constants $\kappa_{\rm ef}$, $\kappa_{\rm eg}$ and for some $\bar{\Delta} \leq \Delta_{\max}$, then m is also fully-linear on $B(\boldsymbol{x}_c; \Delta)$, for any $\Delta \in [\bar{\Delta}, \Delta_{\max}]$, with respect to the same constants $\kappa_{\rm ef}$ and $\kappa_{\rm eg}$.*

(AF2) *For any $\boldsymbol{x}_c \in \mathcal{L}(\boldsymbol{x}_0)$ and $\Delta \in (0, \Delta_{\max}]$, we can obtain a fully-linear model (with respect to $\kappa_{\rm ef}$, $\kappa_{\rm eg}$) on $B(\boldsymbol{x}_c; \Delta)$ in a finite, uniformly bounded (with respect to $\boldsymbol{x}_c$ and Δ) number of steps, say $N_{\max}$.*

∎

We see that, the assumptions in Assumption 3.1 are the same list in Assumption 2.1, except for (AM3). Recall that, if $\varepsilon_k^{\rm cur} = 0$, then (3.23) is an interpolation condition at the current point. We discuss how to construct model functions which satisfy (AM1)-(AM3) in Section 4. The assumption (AF2) is imposed in order to prevent the infinite loop in Step 1 and Step 8 of Algorithm 3.1. Under (AM2) and (AM3), it is not difficult to satisfy the assumption (AF2). In fact, as we described in Section 3.1, if $n+1$ points in the trust-region are sufficiently affinely independent and if the accuracies at these points satisfies (3.12), then the model function has error bounds similar to (3.13) and (3.14). Moreover, such $n+1$ points can be found by using Algorithm A.1 with at most $n+1$ function evaluations.

Now, let us start the main part of the analysis. First, we will show that unless the current iterate is a first-order critical point, the algorithm will not loop infinitely in the criticality step of Algorithm 3.1.

Lemma 3.5 (c.f. [5, Lemma 5.1]) *If $\nabla f(\boldsymbol{x}_k) \neq 0$, then Step 1 of Algorithm 3.1 will terminate in a finite number of improvement steps (by applying Algorithm 3.2).*

Proof. Since the proof is almost same as that of Lemma 5.1 in [5], we omit it.

∎

Remark 3.1 *Lemma 3.5 states that if an infinite loop occurs within Step 1 of Algorithm 3.1, this must be because the current iterate is first-order critical. In this case, the number of successful iteration is finite.*

Next, we will give sufficient conditions for the k-th iteration to be successful.

Lemma 3.6 (c.f. [5, Lemma 5.2]) *Suppose that Assumption 3.1 holds. If m_k is fully-linear on $B(\boldsymbol{x}_k; \Delta_k)$ and*

$$\Delta_k \leq \|\boldsymbol{g}_k\| \min\left[\frac{1}{\kappa_{\rm bhm}}, \frac{\kappa_{\rm md}(1-\eta_1)}{2C + \kappa_{\rm ef}}\right],$$

*then, the k-th iteration is **successful**.*

[*6] "md" stands for "model decrease".

Proof. Since $\Delta_k \leq \frac{\|\boldsymbol{g}_k\|}{\kappa_{\mathrm{bhm}}}$, the sufficient decrease condition (3.24) and the uniformly boundedness of the model's Hessian (3.22) imply that

$$m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k) \geq \kappa_{\mathrm{md}} \|\boldsymbol{g}_k\| \min\left[\frac{\|\boldsymbol{g}_k\|}{\kappa_{\mathrm{bhm}}}, \Delta_k\right] = \kappa_{\mathrm{md}} \|\boldsymbol{g}_k\| \Delta_k. \tag{3.25}$$

From the definitions (3.1) of $f_\varepsilon(\boldsymbol{x})$, we have

$$\begin{aligned} \left|f(\boldsymbol{x}_k) - f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k)\right| &\leq \varepsilon_k^{\mathrm{cur}}, \\ \left|f(\boldsymbol{x}_k + \boldsymbol{s}_k) - f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_k + \boldsymbol{s}_k)\right| &\leq \varepsilon_k^{\mathrm{tri}}, \end{aligned} \tag{3.26}$$

respectively. Since the current model m_k is fully-linear on $B(\boldsymbol{x}_k; \Delta_k)$, it then follows from (2.5) that

$$|f(\boldsymbol{x}_k + \boldsymbol{s}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)| \leq \kappa_{\mathrm{ef}} \Delta_k^2. \tag{3.27}$$

Using (3.26), (3.27) and sufficient decrease condition (3.24), we obtain

$$\begin{aligned} 1 - \rho_k &\leq |\rho_k - 1| \\ &= \left|\frac{f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_k + \boldsymbol{s}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)} - 1\right| \\ &= \left|\frac{f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_k + \boldsymbol{s}_k) - m_k(\boldsymbol{x}_k) + m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}\right| \\ &\leq \left|\frac{f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_k + \boldsymbol{s}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}\right| + \left|\frac{f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}\right| \\ &= \left|\frac{f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_k + \boldsymbol{s}_k) - f(\boldsymbol{x}_k + \boldsymbol{s}_k) + f(\boldsymbol{x}_k + \boldsymbol{s}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}\right| + \left|\frac{f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}\right| \\ &\leq \left|\frac{f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_k + \boldsymbol{s}_k) - f(\boldsymbol{x}_k + \boldsymbol{s}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}\right| + \left|\frac{f(\boldsymbol{x}_k + \boldsymbol{s}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}\right| + \left|\frac{f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}\right| \\ &\leq \frac{1}{\kappa_{\mathrm{md}} \|\boldsymbol{g}_k\| \Delta_k} \times \left(\varepsilon_k^{\mathrm{tri}} + \kappa_{\mathrm{ef}} \Delta_k^2 + \varepsilon_k^{\mathrm{cur}}\right) \\ &\leq \frac{1}{\kappa_{\mathrm{md}} \|\boldsymbol{g}_k\| \Delta_k} \times \left(C\Delta_k^2 + \kappa_{\mathrm{ef}} \Delta_k^2 + C\Delta_k^2\right) \qquad \left(\because \varepsilon_k^{\mathrm{tri}} \leq C\Delta_k^2, \varepsilon_k^{\mathrm{cur}} \leq C\Delta_k^2\right) \\ &= \frac{(2C + \kappa_{\mathrm{ef}})\Delta_k}{\kappa_{\mathrm{md}} \|\boldsymbol{g}_k\|} \\ &\leq 1 - \eta_1, \end{aligned}$$

where the last inequality follows the assumption $\Delta_k \leq \|\boldsymbol{g}_k\| \cdot \frac{\kappa_{\mathrm{md}}(1-\eta_1)}{2C+\kappa_{\mathrm{ef}}}$. Therefore, $\rho_k \geq \eta_1$, and the iteration k is successful. ∎

Next, we show that if the gradient of the model is bounded away from zero, then the trust region radius is also bounded away from zero.

Lemma 3.7 (c.f. [5, Lemma 5.3]) *Suppose that Assumption 3.1 holds. Also suppose that there exists a constant $\kappa_1 > 0$ such that $\|\boldsymbol{g}_k\| \geq \kappa_1$ for all k. Then, there exists a constant $\kappa_2 > 0$ such that*

$$\Delta_k \geq \kappa_2$$

for all k.

Proof. From Step 1 of Algorithm 3.1, we have

$$\Delta_k \geq \min\{\beta \|\boldsymbol{g}_k\|, \Delta_k^{\mathrm{icb}}\} \quad \text{for all } k.$$

Thus,

$$\Delta_k \geq \min\{\beta\kappa_1, \Delta_k^{\mathrm{icb}}\} \quad \text{for all } k. \tag{3.28}$$

Here, let us consider how Δ_k can be updated at Step 5 of Algorithm 3.1. We consider two cases: $\Delta_k \leq \bar{\kappa}_2$ and $\Delta > \bar{\kappa}_2$, where $\bar{\kappa}_2 = \min\left[\frac{\kappa_1}{\kappa_{\mathrm{bhm}}}, \frac{\kappa_{\mathrm{md}}\kappa_1(1-\eta_1)}{2C+\kappa_{\mathrm{ef}}}\right]$.

First, suppose that $\Delta_k \leq \bar{\kappa}_2$. Since $\|\boldsymbol{g}_k\| \geq \kappa_1$ for all k, the k-th iteration is successful by Lemma 3.6 (if m_k is fully-linear, or m_k is not certifiably fully-linear but $\rho_k \geq \eta_1$), or model-improving (if m_k is not certifiably fully-linear and $\rho_k < \eta_1$). Thus, in either case, we have

$$\Delta_{k+1}^{\mathrm{icb}} \geq \Delta_k.$$

Next, suppose that $\Delta_k > \bar{\kappa}_2$. Then, the mechanism of the algorithm (Step 6) makes

$$\Delta_{k+1}^{\mathrm{icb}} \geq \gamma_{\mathrm{dec}}\Delta_k.$$

Combine the results for both cases, it results

$$\Delta_{k+1}^{\mathrm{icb}} \geq \min\{\Delta_k, \gamma_{\mathrm{dec}}\bar{\kappa}_2\} \quad \text{for all } k. \tag{3.29}$$

It then follows from (3.28) and (3.29) that

$$\Delta_{k+1} \geq \min\left\{\beta\kappa_1, \Delta_{k+1}^{\mathrm{icb}}\right\} \geq \min\{\beta\kappa_1, \Delta_k, \gamma_{\mathrm{dec}}\bar{\kappa}_2\} \quad \text{for all } k.$$

Using this inequality recursively, we obtain $\Delta_k \geq \min\{\beta\kappa_1, \Delta_0^{\mathrm{icb}}, \gamma_{\mathrm{dec}}\bar{\kappa}_2\} =: \kappa_2$. ∎

From Lemma 3.7, the trust-region radius cannot become too small as long as a current iterate is sufficiently far from a critical point.

The above results are sufficient to analyze criticality if the number of successful iterations is finite.

Lemma 3.8 (c.f. [5, Lemma 5.4]) *If the number of successful iterations is finite, then $\boldsymbol{x}_k = \boldsymbol{x}_*$ for k sufficiently large and $\nabla f(\boldsymbol{x}_*) = 0$.*

Proof. Note that, if an infinite loop occurs in Step 1, then the result follows from Lemma 3.5. So, we must focus on the case in which an infinite loop does not occur in Step 1.

Let us consider iterations that come after the "last" successful iteration. The mechanism of the algorithm 3.1 ensures that $\boldsymbol{x}_* = \boldsymbol{x}_{k_0+1} = \boldsymbol{x}_{k_0+j}$ for all $j > 0$, where k_0 is the index of the last successful iterate. Since there are no more successful iterations, then the iterations are unsuccessful or model-improving, and Δ_k is never increased.

Indeed, if the k-th iterate is model-improving, then the model m_{k+1}^{icb} is fully-linear on $B(\boldsymbol{x}_*; \Delta_{k+1}^{\mathrm{icb}})$ at the end of the k-th iterate. Thus, by (AF1), the model m_{k+1} is fully-linear on $B(\boldsymbol{x}_*; \Delta_{k+1})$ at the end of Step 1 in the k+1-iteration, independently of whether Algorithm 3.2 has been invoked. Therefore, the k+1-th iterate should be unsuccessful from the mechanism of the algorithm. In other words, for every two iterations, at least one of them is unsuccessful. Since trust region radius is decreased by a factor of γ_{dec} in the unsuccessful iteration, it follows that $\Delta_k \to 0$ as $k \to \infty$.

Now, for each j, let i_j be the index of the first iteration after the j-th iteration for which the model m_j is fully-linear on $B(\boldsymbol{x}_*; \Delta_{i_j})$. In our situation, $i_j = j + 1$ or $j + 2$, since for every two iterations, at least one of them is unsuccessful.

Since m_j is fully-linear on $B(\boldsymbol{x}_*; \Delta_{i_j})$, it holds that

$$\left\|\nabla f(\boldsymbol{x}_*) - \boldsymbol{g}_{i_j}\right\| \leq \kappa_{\mathrm{eg}} \Delta_{i_j}. \tag{3.30}$$

Let us derive a contradiction by assuming that $\nabla f(\boldsymbol{x}_*) \neq 0$. Since $\{\Delta_k\}$ converges to zero, (3.30) implies that $\left\|\boldsymbol{g}_{i_j}\right\| \geq \kappa_1 = \frac{1}{2}\left\|\nabla f(\boldsymbol{x}_*)\right\| > 0$ for $j \geq k_0$ sufficiently large. But Lemma 3.7 states that this situation is impossible, and hence $\nabla f(\boldsymbol{x}_*) = 0$. ∎

Now, we will prove a key property on Δ_k for the global convergence, which gives a natural stopping criterion for the derivative-free trust-region method.

Theorem 3.9 (c.f. [5], Lemma 5.5) *Suppose that assumption 3.1 holds. Then,*

$$\lim_{k\to\infty} \Delta_k = 0. \tag{3.31}$$

Proof. When $\mathcal{S}$ is finite, (3.31) is shown in the proof of Lemma 3.8. Let us consider the case when $\mathcal{S}$ is infinite. Since $\rho_k \geq \eta_1$ and $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$ for $k \in \mathcal{S}$, we have

$$f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - f_{\boldsymbol{\varepsilon_k^{\mathrm{tri}}}}(\boldsymbol{x}_{k+1}) \geq \eta_1 \left[m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)\right], \quad \text{for any } k \in \mathcal{S}.$$

By using the sufficient decrease condition (3.24), we have that

$$f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_{k+1}) \geq \eta_1 \kappa_{\mathrm{md}} \left\|\boldsymbol{g}_k\right\| \min\left[\frac{\left\|\boldsymbol{g}_k\right\|}{\left\|\boldsymbol{H}_k\right\|}, \Delta_k\right].$$

Due to Step 1 of Algorithm 3.1, we have $\left\|\boldsymbol{g}_k\right\| \geq \min\{\epsilon_{\mathrm{cri}}, \mu^{-1}\Delta_k\}$. It then follows from $\left\|\boldsymbol{H}_k\right\| \leq \kappa_{\mathrm{bhm}}$ (AM2) that

$$f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_{k+1}) \geq \eta_1 \kappa_{\mathrm{md}} \min\{\epsilon_{\mathrm{cri}}, \mu^{-1}\Delta_k\} \min\left[\frac{\min\{\epsilon_{\mathrm{cri}}, \mu^{-1}\Delta_k\}}{\kappa_{\mathrm{bhm}}}, \Delta_k\right]. \tag{3.32}$$

On the other hand, since $f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k)$ and $f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_{k+1})$ $(= f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_k))$ satisfy (3.1), we have

$$f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) + \varepsilon_k^{\mathrm{cur}} + \varepsilon_k^{\mathrm{tri}}.$$

Using this inequality, (3.32) can be rewritten as:

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) + \varepsilon_k^{\mathrm{cur}} + \varepsilon_k^{\mathrm{tri}} \geq \eta_1 \kappa_{\mathrm{md}} \min\{\epsilon_{\mathrm{cri}}, \mu^{-1}\Delta_k\} \min\left[\frac{\min\{\epsilon_{\mathrm{cri}}, \mu^{-1}\Delta_k\}}{\kappa_{\mathrm{bhm}}}, \Delta_k\right]. \tag{3.33}$$

To simplify the notation, let us denote the right-hand side of (3.33) by a_k.

Note that, $\boldsymbol{x}_k = \boldsymbol{x}_{k+1}$ for $k \notin \mathcal{S}$, and hence $f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) = 0$ for $k \notin \mathcal{S}$. Now, taking the summation of both sides of (3.33) over $0 \leq k \leq K, k \in \mathcal{S}$ for some positive constant K, we have

$$\begin{aligned}
\sum_{\substack{0\leq k\leq K\\ k\in\mathcal{S}}} (\text{the left-hand side of (3.33)}) &= \sum_{\substack{0\leq k\leq K\\ k\notin\mathcal{S}}} \{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1})\} + \sum_{\substack{0\leq k\leq K\\ k\in\mathcal{S}}} \{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1})\} + \sum_{\substack{0\leq k\leq K\\ k\in\mathcal{S}}} \left\{\varepsilon_k^{\mathrm{cur}} + \varepsilon_k^{\mathrm{tri}}\right\} \\
&= \sum_{k=0}^{K} \{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1})\} + \sum_{\substack{0\leq k\leq K\\ k\in\mathcal{S}}} \left\{\varepsilon_k^{\mathrm{cur}} + \varepsilon_k^{\mathrm{tri}}\right\} \\
&\leq f(\boldsymbol{x}_0) - f(\boldsymbol{x}_{K+1}) + \sum_{k=0}^{K} \varepsilon_k^{\mathrm{cur}} + \sum_{k=0}^{K} \varepsilon_k^{\mathrm{tri}} \\
&\leq f(\boldsymbol{x}_0) - f(\boldsymbol{x}_{K+1}) + \sum_{k=0}^{K} M\zeta^k + \sum_{k=0}^{K} M\zeta^{k+1},
\end{aligned}$$

where the last inequality follows from (3.20) and (3.21). Since $\zeta \in (0,1)$, it holds that

$$\sum_{k=0}^{K} M\zeta^k + \sum_{k=0}^{K} M\zeta^{k+1} \leq \sum_{k=0}^{\infty} M\zeta^k + \sum_{k=0}^{\infty} M\zeta^{k+1} = \frac{M}{1-\zeta} + \frac{M\zeta}{1-\zeta}.$$

Moreover, since the sequence $\{f(\boldsymbol{x}_k)\}$ is bounded from below by (AO1), there exists a constant κ_* such that $f(\boldsymbol{x}_k) \geq \kappa_*$, for all k. Then we have

$$\sum_{\substack{0\leq k\leq K\\ k\in\mathcal{S}}} (\text{the left-hand side of (3.33)}) \leq f(\boldsymbol{x}_0) - \kappa_* + \frac{M}{1-\zeta} + \frac{M\zeta}{1-\zeta} =: c_0. \tag{3.34}$$

On the other hand, from $a_k \geq 0$, for all k, it holds that

$$\sum_{\substack{0\leq k\leq K\\ k\in\mathcal{S}}} (\text{the right-hand side of (3.33)}) = \sum_{\substack{0\leq k\leq K\\ k\in\mathcal{S}}} a_k \geq 0. \tag{3.35}$$

Define $S_K := \sum_{\substack{0\leq k\leq K\\ k\in\mathcal{S}}} a_k$, and by using (3.34), (3.35), we have

$$0 \leq S_K \leq c_0, \quad \text{for all } K.$$

Since $a_k \geq 0$, for all k, then the sequence $\{S_K\}_{K\geq 0}$ is monotonic increasing and bounded from above. Thus, the sequence $\{S_K\}_{K\geq 0}$ converges to some constant S^*, which means

$$\sum_{k\in\mathcal{S}} a_k = S^*.$$

Since we assumed that $\mathcal{S}$ is infinite, it follows that

$$\lim_{\substack{k\to\infty\\ k\in\mathcal{S}}} a_k = 0.$$

Recall that $a_k = \eta_1 \kappa_{\text{md}} \min\{\epsilon_{\text{cri}}, \mu^{-1}\Delta_k\} \min\left[\frac{\min\{\epsilon_{\text{cri}}, \mu^{-1}\Delta_k\}}{\kappa_{\text{bhm}}}, \Delta_k\right]$, it follows that

$$\lim_{\substack{k\to\infty\\ k\in\mathcal{S}}} \Delta_k = 0.$$

Note that Δ_k can increase only during successful iterations. Let $k \notin \mathcal{S}$ be the index of an iteration (after the first successful one). Then $\Delta_k \leq \gamma_{\text{inc}}\Delta_{s_k}$, where s_k is the index of the last successful iteration before k. Since $\Delta_{s_k} \to 0$, then $\Delta_k \to 0$, for $k \notin \mathcal{S}$. Hence, $\Delta_k \to 0$ along all iterations. ∎

Since we have proved the global convergence property for the case where $\mathcal{S}$ is finite, then we now restrict our attention to the case where $\mathcal{S}$ is infinite. First, we will prove that at least one of the accumulation point of $\{\boldsymbol{x}_k\}$ must be critical when the sequence is infinite. In order to achieve this, the following lemma is needed.

Lemma 3.10 (c.f. [5, Lemma 5.6, 5.7]) *Suppose that Assumption 3.1 holds. Then,*

1. *$\{\|\boldsymbol{g}_k\|\}$ has a convergent subsequence, i.e.,*

$$\liminf_{k\to\infty} \|\boldsymbol{g}_k\| = 0.$$

2. *For any subsequence $\{k_i\} \subset \{0, 1, 2, \cdots\}$ such that*

$$\lim_{i\to\infty} \|\boldsymbol{g}_{k_i}\| = 0,$$

it also holds that

$$\lim_{i\to\infty} \|\nabla f(\boldsymbol{x}_{k_i})\| = 0.$$

Proof. Since the proof is almost same as that of Lemma 5.6 and 5.7 in [5], we omit it. ∎

Using this lemma, we obtain the following global convergence result.

Theorem 3.11 *Suppose that Assumption 3.1 holds. Then*

$$\liminf_{k\to\infty} \|\nabla f(\boldsymbol{x}_k)\| = 0.$$

Proof. This follows directly from Lemma 3.10. ∎

Finally, we prove that all limit points of the sequence generated by the proposed algorithm are first-order critical.

Theorem 3.12 *Suppose that Assumption 3.1 holds. Then*

$$\lim_{k\to\infty} \nabla f(\boldsymbol{x}_k) = 0.$$

Proof. When $\mathcal{S}$ is finite, the theorem follows from Lemma 3.8. So, let us consider the case when $\mathcal{S}$ is infinite. Let us derive a contradiction by assuming that there exists a subsequence $\{t_i\}_{i\geq 0} \subset \mathcal{S}$ such that

$$\|\nabla f(\boldsymbol{x}_{t_i})\| \geq \nu_0 > 0, \tag{3.36}$$

for some $\nu_0 > 0$. Here, we have ignored model-improvement and unsuccessful iterations because $\boldsymbol{x}_k$ does not change during such iterations. Note that $\{t_i\}_{i\geq 0}$ is a infinite sequence. From Lemma 3.10 (2), there exists $\nu > 0$ such that

$$\|\boldsymbol{g}_{t_i}\| \geq \nu > 0,$$

for all i sufficiently large. Without loss of generality, we assume

$$\nu \leq \min\left\{\frac{\nu_0}{2(2+\kappa_{\mathrm{eg}}\mu)}, \varepsilon_{\mathrm{cri}}\right\}. \tag{3.37}$$

Moreover, without loss of generality and without change of notation, we can obtain a subsequence of $\{t_i\}_{i\geq 0}$ indexed by $\{\ell_i\}$ such that

$$\|\boldsymbol{g}_{\ell_i}\| \geq \nu \quad \text{and} \quad \|\boldsymbol{g}_{\ell_i-1}\| < \nu,$$

for all i. In addition, $\{\ell_i\}_{i\geq 0}$ is also an infinite sequence. Lemma 3.10 (1) ensures the existence for each ℓ_i of a first iteration $u(\ell_i) > \ell_i$ such that $\left\|\boldsymbol{g}_{u(\ell_i)}\right\| < \nu$. Denoting $u_i := u(\ell_i)$, we thus obtain that there exist another subsequence of the indices of iterations indexed by $\{u_i\}_{i\geq 0}$ such that

$$\|\boldsymbol{g}_k\| \geq \nu \ \text{ for } \ell_i \leq k < u_i \ \text{ and } \ \|\boldsymbol{g}_{u_i}\| < \nu, \tag{3.38}$$

for all i.

Now, for each i, we define the set

$$\mathcal{K}_i := \{\, k \in \{0,1,2,\cdots\} \,\big|\, \|\boldsymbol{g}_k\| \geq \nu \ \text{ for } \ell_i \leq k < u_i \ \text{ and } \ \|\boldsymbol{g}_{u_i}\| < \nu \,\}.$$

An illustration of the definitions of the subsequences $\{t_i\}_{i\geq 0}$, $\{\ell_i\}_{i\geq 0}$, $\{u_i\}_{i\geq 0}$, and the sets $\mathcal{K}_i$ is presented in Figure 1, which it is similar to the figure in Conn *et al.* [3, Section 6.4]. Note that, in Figure 1, we have marked position k in each of the subsequences represented in the abscissa when k belongs to that subsequences. In this example, $\ell_0 = 1$, $u_0 = 5$, $\ell_1 = 7$, $u_1 = 7$, $\mathcal{K}_0 = \{1,2,3\}$, and so on.

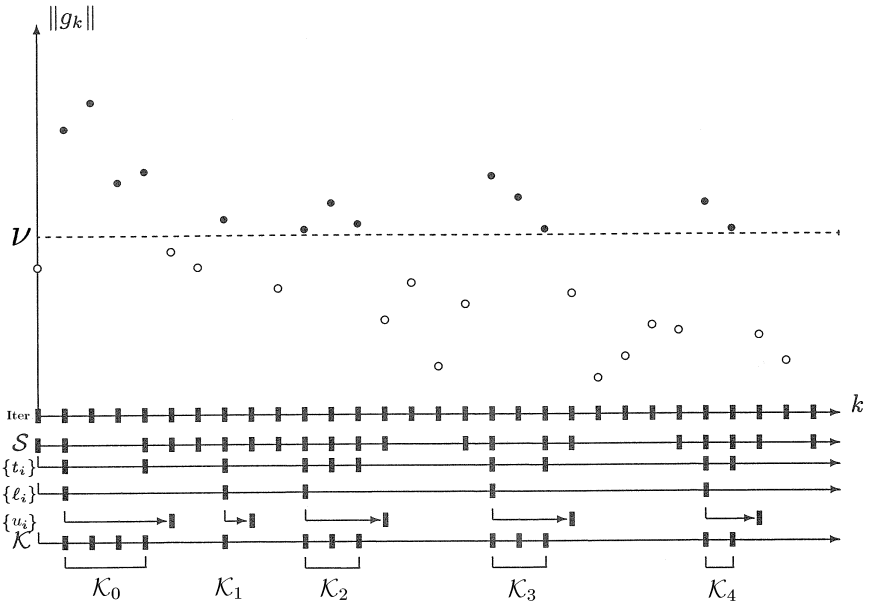

Figure 1 An example of the subsequences of the proof of Theorem 3.11 (c.f. [3, Section 6.4]).

From the definitions of $\{\ell_i\}_{i\geq 0}$ and $\{u_i\}_{i\geq 0}$, $u_i < \ell_{i+1}$ holds for all i. Thus, the collection of sets $\mathcal{K}_i$, $i \geq 0$ are pairwise disjoint. Let us define the set

$$\mathcal{K} := \bigcup_{i\geq 0} \mathcal{K}_i,$$

and let us restrict our attention to the iterations whose indices are in the set $\mathcal{K}$.

Since $\lim_{k\to\infty} \Delta_k = 0$ by Theorem 3.9, there exists a constant K_0 such that

$$\Delta_k \leq \nu \cdot \min\left\{\frac{1}{\kappa_{\rm bhm}}, \frac{\kappa_{\rm md}(1-\eta_1)}{2C+\kappa_{\rm ef}}\right\},$$

for all $k \geq K_0$. Now we recall that $\|\boldsymbol{g}_k\| \geq \nu$ for $k \in \mathcal{K}$. From the proof of Lemma 3.7, we can see that, for all $k \in \mathcal{K}$ and $k \geq K_0$, the k-th iteration is either successful or model improving. Since $\boldsymbol{x}_k = \boldsymbol{x}_{k+1}$ for a model improving iteration, we have for all i sufficiently large,

$$\|\boldsymbol{x}_{\ell_i} - \boldsymbol{x}_{u_i}\| \leq \sum_{j\in\mathcal{K}_i\cap\mathcal{S}} \|\boldsymbol{x}_j - \boldsymbol{x}_{j+1}\| \leq \sum_{j\in\mathcal{K}_i\cap\mathcal{S}} \Delta_j. \tag{3.39}$$

We will show that $\lim_{i\to\infty} \|\boldsymbol{x}_{\ell_i} - \boldsymbol{x}_{u_i}\| = 0$. For that purpose, we define

$$b_i := \sum_{j\in\mathcal{K}_i\cap\mathcal{S}} \Delta_j, \tag{3.40}$$

and we will show that $\lim_{i\to\infty} b_i = 0$.

Since $\|\boldsymbol{g}_k\| \geq \nu$ for $k \in \mathcal{K}$, it holds that for any $k \in \mathcal{K}\cap\mathcal{S}$ and $k \geq K_0$,

$$\begin{aligned} f_{\varepsilon_k^{\rm cur}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\rm tri}}(\boldsymbol{x}_{k+1}) &\geq \eta_1 \left[m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)\right] \\ &\geq \eta_1 \kappa_{\rm md} \|\boldsymbol{g}_k\| \min\left[\frac{\|\boldsymbol{g}_k\|}{\kappa_{\rm bhm}}, \Delta_k\right] \\ &\geq \eta_1 \kappa_{\rm md} \cdot \nu \cdot \min\left[\frac{\nu}{\kappa_{\rm bhm}}, \Delta_k\right] = \eta_1 \kappa_{\rm md} \nu \Delta_k. \end{aligned}$$

Thus, for any $k \in \mathcal{K} \cap \mathcal{S}$ and $k \geq K_0$, we have

$$\Delta_k \leq \frac{1}{\eta_1 \kappa_{\mathrm{md}} \nu} \left[f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_{k+1}) \right].$$

Here, let ℓ_{i^*} be the smallest element of $\{\ell_i\}_{i \geq 0}$ such that $\ell_i \geq K_0$. Then, for any $K \geq \ell_{i^*}$, we obtain

$$\begin{aligned}
f(\boldsymbol{x}_{\ell_{i^*}}) - f(\boldsymbol{x}_{K+1}) &= \sum_{k=\ell_{i^*}}^{K} \{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1})\} \\
&= \sum_{\substack{\ell_{i^*} \leq k \leq K \\ k \in \mathcal{S}}} \{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1})\} \qquad (\because \boldsymbol{x}_{k+1} = \boldsymbol{x}_k \text{ for } k \notin \mathcal{S}) \\
&\geq \sum_{\substack{\ell_{i^*} \leq k \leq K \\ k \in \mathcal{S}}} \left\{ f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_{k+1}) \right\} - \sum_{\substack{\ell_{i^*} \leq k \leq K \\ k \in \mathcal{S}}} \left\{ \varepsilon_k^{\mathrm{cur}} + \varepsilon_k^{\mathrm{tri}} \right\} \\
&\geq \sum_{\substack{\ell_{i^*} \leq k \leq K \\ k \in \mathcal{K} \cap \mathcal{S}}} \left\{ f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_{k+1}) \right\} - \sum_{\substack{\ell_{i^*} \leq k \leq K \\ k \in \mathcal{S}}} \left\{ M\zeta^k + M\zeta^{k+1} \right\} \\
&\geq \sum_{\substack{\ell_{i^*} \leq k \leq K \\ k \in \mathcal{K} \cap \mathcal{S}}} \left\{ f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_{k+1}) \right\} - \sum_{k=0}^{\infty} \left\{ M\zeta^k + M\zeta^{k+1} \right\} \\
&= \sum_{\substack{\ell_{i^*} \leq k \leq K \\ k \in \mathcal{K} \cap \mathcal{S}}} \left\{ f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_{k+1}) \right\} - \frac{M(1+\zeta)}{1-\zeta}, \qquad (3.41)
\end{aligned}$$

where the second inequality follows from $\sum_{\substack{\ell_{i^*} \leq k \leq K \\ k \notin \mathcal{K}, k \in \mathcal{S}}} \left\{ f_{\varepsilon_k^{\mathrm{cur}}}(\boldsymbol{x}_k) - f_{\varepsilon_k^{\mathrm{tri}}}(\boldsymbol{x}_{k+1}) \right\} \geq 0$, (3.20), and (3.21). Let q be any integer such that $q \geq i^*$. Then, the inequality (3.41) for $K = u_q - 1$ is written as,

$$\begin{aligned}
f(\boldsymbol{x}_{\ell_{i^*}}) - f(\boldsymbol{x}_{u_q - 1}) &\geq \eta_1 \kappa_{\mathrm{md}} \varepsilon \cdot \left(\sum_{\substack{\ell_{i^*} \leq j \leq u_q - 1 \\ j \in \mathcal{K} \cap \mathcal{S}}} \Delta_j \right) - \frac{M(1+\zeta)}{1-\zeta} \\
&= \eta_1 \kappa_{\mathrm{md}} \varepsilon \cdot \left(\sum_{i=i^*}^{q} \sum_{j \in \mathcal{K}_i \cap \mathcal{S}} \Delta_j \right) - \frac{M(1+\zeta)}{1-\zeta}.
\end{aligned}$$

Since $f(\boldsymbol{x}_{\ell_{i^*}}) - f(\boldsymbol{x}_{u_q - 1}) \leq f(\boldsymbol{x}_{\ell_{i^*}}) - \kappa_*$ by (AO1),

$$f(\boldsymbol{x}_{t_{i^*}}) - \kappa_* \geq \eta_1 \kappa_{\mathrm{md}} \nu \cdot \left(\sum_{i=i^*}^{q} b_i \right) - \frac{M(1+\zeta)}{1-\zeta}, \qquad \text{for all } q \geq i^*.$$

Therefore, the infinite sequence $\left\{ \sum_{i=i^*}^{q} b_i \right\}_{q \geq i^*}$, indexed by q, is monotonically increasing and bounded above. Thus, it converges. Since $\mathcal{K} \cap \mathcal{S}$ is the infinite subset, it follows that

$$\lim_{i \to \infty} b_i = \lim_{i \to \infty} \left(\sum_{j \in \mathcal{K}_i \cap \mathcal{S}} \Delta_j \right) = 0.$$

Then, from (3.39),

$$\lim_{i \to \infty} \| \boldsymbol{x}_{\ell_i} - \boldsymbol{x}_{u_i} \| = 0.$$

Here, using the triangle inequality, we have

$$\| \nabla f(\boldsymbol{x}_{\ell_i}) \| \leq \| \nabla f(\boldsymbol{x}_{\ell_i}) - \nabla f(\boldsymbol{x}_{u_i}) \| + \| \nabla f(\boldsymbol{x}_{u_i}) - \boldsymbol{g}_{u_i} \| + \| \boldsymbol{g}_{u_i} \|.$$

Now, let us investigate the upper bounds of all three terms in the right-hand side seperately. Since ∇f is Lipschitz continuous (AO2), the first term tends to 0 as $i \to \infty$. Thus, it holds that

$$\|\nabla f(\boldsymbol{x}_{\ell_i}) - \nabla f(\boldsymbol{x}_{u_i})\| \leq \nu,$$

for i sufficiently large. From (3.38), the third term is also bounded by ν, for i sufficiently large. Finally, we consider the second term. From (3.37), the mechanism of the criticality step at iteration u_i, and assumption (AF1), the model m_{u_i} is fully-linear on $B(\boldsymbol{x}_{u_i}; \mu \|\boldsymbol{g}_{u_i}\|)$. Therefore, using (2.4) and (3.38), we have

$$\|\nabla f(\boldsymbol{x}_{u_i}) - \boldsymbol{g}_{u_i}\| \leq \kappa_{\mathrm{eg}} \mu \|\boldsymbol{g}_{u_i}\| < \kappa_{\mathrm{eg}} \mu \nu,$$

for i sufficiently large.

As a consequence, we obtain from these bounds and from (3.37) that

$$\|\nabla f(\boldsymbol{x}_{\ell_i})\| \leq (2 + \kappa_{\mathrm{eg}} \mu)\nu \leq \frac{1}{2}\nu_0,$$

for i large enough. But, since $\{\ell_i\}_{i \geq 0}$ is an infinite subsequence of $\{t_i\}_{i \geq 0}$, this inequality contradicts (3.36). Hence, our initial assumption must be false and the theorem follows. ∎

4 Construction of a fully-linear model from sample points with pointwise errors

In this section, we consider how to construct a quadratic model $m : \mathbb{R}^n \to \mathbb{R}$ which satisfies the assumptions of Corollary 3.2. The most simplest way to construct such models is to adopt linear function as the models. However, it is well known that derivative-free trust-region method with linear function converges slowly, because the linear function does not involve curvature information on f. Hence, we are interested in quadratic models.

Let $\phi_L(\boldsymbol{x})$ and $\phi_Q(\boldsymbol{x})$ be vectors of monomials such that

$$\begin{aligned}
\phi_L(\boldsymbol{x}) &= [1, x_1, \cdots, x_n]^\top \in \mathbb{R}^{n+1}, \\
\phi_Q(\boldsymbol{x}) &= \left[\frac{1}{2}x_1^2, x_1 x_2, \cdots, x_{n-1}x_n, \frac{1}{2}x_n^2\right]^\top \in \mathbb{R}^{\frac{n(n+1)}{2}},
\end{aligned}$$

where x_i denotes the i-th component of a variable $\boldsymbol{x} \in \mathbb{R}^n$. And we also define

$$\phi(\boldsymbol{x}) = \left[\phi_L(\boldsymbol{x})^\top \phi_Q(\boldsymbol{x})^\top\right]^\top$$

whose components form a (natural) basis for the linear space of polynomials of degree less than or equal to 2 in $\mathbb{R}^n$. Then, any quadratic models $m : \mathbb{R}^n \to \mathbb{R}$ can be written in the form

$$m(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) = \boldsymbol{w}_L^\top \phi_L(\boldsymbol{x}) + \boldsymbol{w}_Q^\top \phi_Q(\boldsymbol{x}), \tag{4.1}$$

where $\boldsymbol{w} = \left[\boldsymbol{w}_L^\top, \boldsymbol{w}_Q^\top\right]^\top \in \mathbb{R}^N$ is a coefficient vector with $N = \frac{(n+1)(n+2)}{2}$.

If we determine a quadratic model by interpolation, we need at least N function evaluations in order to specify the coefficient $\boldsymbol{w}$ completely. Nevertheless, under (S0), it may be too expensive to evaluate f at N sample points nearby the current trust region. Then, we discuss how to construct fully-linear models with curvature information on f, based on possibly fewer than N function evaluations.

4.1 Preliminary: the case when f is evaluated exactly

When f is evaluated exactly, Wild [17, 19, 18] proposed several methods for constructing fully-linear models using function values evaluated at points fewer than N.

Let $\mathcal{D}_c$ be a set of displacements $\boldsymbol{d}^i$ from a certain point $\boldsymbol{x}_c$ such that $f(\boldsymbol{x}_c + \boldsymbol{d}^i)$ has been evaluated. Wild [18] proposed constructing the model by solving the following quadratic programming problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\boldsymbol{w}_Q\|^2 \\ \text{subject to} \quad & \boldsymbol{w}^\top \phi(\boldsymbol{x}_c + \boldsymbol{y}^i) - f(\boldsymbol{x}_c + \boldsymbol{y}^i) = 0 \quad \text{for all } \boldsymbol{y}^i \in \mathcal{Y}, \end{aligned} \tag{4.2}$$

where, $\mathcal{Y} = \left\{\boldsymbol{y}^1 = 0, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^{|\mathcal{Y}|}\right\} \subset \mathcal{D}_c$ is a set of distinct data points such that $n+1 \leq |\mathcal{Y}| \leq N$.

The problem is to find a model function m whose Hessian is smallest in terms of Frobenius norm subject to the interpolation conditions on $\boldsymbol{x}_c + \boldsymbol{y}^i$, $\boldsymbol{y}^i \in \mathcal{Y}$, since $\|\boldsymbol{w}_Q\| = \left\|\nabla^2 m_(\boldsymbol{x})\right\|_F$. If $|\mathcal{Y}| = n+1$, then we have $\boldsymbol{w}_Q = 0$, and hence the model m is a linear function. Let us consider the case where $n+1 < |\mathcal{Y}| \leq N$. In this case, if $n+1$ points (include 0) in $\mathcal{Y}$ are sufficiently affinely independent, then the resulting model m is certifiably fully-linear on $B(\boldsymbol{x}_c; \Delta)$ with respect to some constants κ_{ef} and κ_{eg}, where Δ is the maximum norm of the $n+1$ sufficiently independent points.

Moreover, the model's gradient ∇m is Lipschitz continuous on $\mathbb{R}^n$ with Lipschitz constant $\|\boldsymbol{w}_Q\|$. In addition, if the all points in $\mathcal{Y}$ satisfy some geometric conditions, then the Lipschitz constant can be bounded by some constant independent of $\mathcal{Y}$ and Δ.

4.2 The case when f is evaluated inexactly

Now, we explain how to construct a quadratic model function which satisfies (3.3), using inexact function values with different accuracies.

Let $\mathcal{D}_c$ be the same set defined in Section 3.1. We replace the constraints of problem (4.2) with

$$-\nu^i \leq \boldsymbol{w}^\top \phi(\boldsymbol{x}_c + \boldsymbol{y}^i) - f_{\nu^i}(\boldsymbol{x}_c + \boldsymbol{y}^i) \leq \nu^i, \tag{4.3}$$

for all $\boldsymbol{y}^i \in \mathcal{Y}$, where $\mathcal{Y} \subset \mathcal{D}_c$ and ν^i denotes the accuracy parameter at $\boldsymbol{x}_c + \boldsymbol{y}^i$. Then the problem (4.2) turns to

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\boldsymbol{w}_Q\|^2 \\ \text{subject to} \quad & -\nu^i \leq \boldsymbol{w}^\top \phi(\boldsymbol{x}_c + \boldsymbol{y}^i) - f_{\nu^i}(\boldsymbol{x}_c + \boldsymbol{y}^i) \leq \nu^i \quad \text{for all } \boldsymbol{y}^i \in \mathcal{Y}. \end{aligned} \tag{4.4}$$

First, we consider the case $n+1 \leq |\mathcal{Y}| \leq N$. We see that the feasible region of (4.4) is larger than that of (4.2). In fact, the optimal solution of (4.2) is a feasible solution of (4.4). This is because the optimal solution of (4.2), say $\bar{\boldsymbol{w}} \in \mathbb{R}^n$, satisfies

$$\bar{\boldsymbol{w}}^\top \phi(\boldsymbol{x}_c + \boldsymbol{y}^i) - f(\boldsymbol{x}_c + \boldsymbol{y}^i) = 0, \quad \text{for all } \boldsymbol{y}^i \in \mathcal{Y},$$

and it then follows from (3.1) that

$$-\nu^i \leq \bar{\boldsymbol{w}}^\top \phi(\boldsymbol{x}_c + \boldsymbol{y}^i) - f_{\nu^i}(\boldsymbol{x}_c + \boldsymbol{y}^i) \leq \nu^i, \quad \text{for all } \boldsymbol{y}^i \in \mathcal{Y}.$$

Since any optimal solutions $\boldsymbol{w}^*$ of (4.4) satisfy $\left\|\boldsymbol{w}_Q^*\right\| \leq \|\bar{\boldsymbol{w}}_Q\|$, the resulting model function with the solution of (4.4) also has a Lipschitz continuous gradient. Moreover, as we described in section 3.1,

if $n+1$ points (include 0) in $(B(\boldsymbol{x}_c;\Delta)-\boldsymbol{x}_c)\cap\mathcal{Y}$ are "sufficiently" affinely independent, and if the accuracy parameters ν^i of the corresponding points satisfy (3.12), then the resulting model is fully-linear on $B(\boldsymbol{x}_c;\Delta)$. (Note that, (3.4) are satisfied from the constraints of (4.4)).

Now we consider the case where $|\mathcal{Y}|\geq N$. Then, it might become difficult to satisfy all constraints of problem (4.4). In that case, we may construct fully-linear models as follows. By introducing the artificial variables $\boldsymbol{\xi}^+,\boldsymbol{\xi}^-\in\mathbb{R}^{|\mathcal{Y}|}$, problem (4.4) can be rewritten as follows:

$$\begin{aligned}
\underset{\boldsymbol{w}_Q,\boldsymbol{\xi}^+,\boldsymbol{\xi}^-}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{w}_Q\right\|^2+\sum_{i=1}^{|\mathcal{Y}|}C^i\left(\xi_i^+ +\xi_i^-\right), & \\
\text{subject to} \quad & f^i-\boldsymbol{w}^\top\phi(\boldsymbol{x}_c+\boldsymbol{y}^i)\leq\nu^i+\xi_i^+ & (i=1,\cdots,|\mathcal{Y}|), \\
& \boldsymbol{w}^\top\phi(\boldsymbol{x}_c+\boldsymbol{y}^i)-f^i\leq\nu^i+\xi_i^- & (i=1,\cdots,|\mathcal{Y}|), \\
& \xi_i^+,\,\xi_i^-\geq 0 & (i=1,\cdots,|\mathcal{Y}|),
\end{aligned} \tag{4.5}$$

where $f^i=f_{\nu^i}(\boldsymbol{x}_c+\boldsymbol{y}^i)$ and C^i is penalty parameters corresponding to the constraints (4.5) for each $\boldsymbol{x}_c+\boldsymbol{y}^i$. For some index i, the bigger penalty parameter C^i makes the constraints (4.5) for i the more satisfied. Especially, if C^i is taken as ∞ for some i, then the constraints (4.5) for i are surely satisfied.

Here, we assume that there exist $n+1$ points (include 0) in $(B(\boldsymbol{x}_c;\Delta)-\boldsymbol{x}_c)\cap\mathcal{Y}$ which are "sufficiently" affinely independent on $\mathbb{R}^n$. Without loss of generality, we denote such points as $\left\{\boldsymbol{y}^1=0,\boldsymbol{y}^2,\cdots,\boldsymbol{y}^{n+1}\right\}$. We take $C^i=\infty$ for $i=1,\cdots,n+1$. Since ξ_i^+ and ξ_i^- must be zero for $i=1,\cdots,n+1$, the constraints of (4.5) for $i=1,\cdots,n+1$ are equivalent to (4.3). The rest C^i, $i=n+2,\cdots,|\mathcal{Y}|$, are adjusted by using some suitable techniques. (For example, by adjusting the values of C^i, $i=n+2,\cdots,|\mathcal{Y}|$, we can weight a high-accuracy sample more than low-accuracy sample in the problem (4.5)).

This approach is similar to the one proposed by Takaki *et al.* [15]. They proposed the method to construct a model via support vector regression (SVR). In their method, the coefficients of the model (4.1) are defined as a minimizer of the following problem:

$$\underset{\boldsymbol{w}_L,\boldsymbol{w}_Q}{\text{minimize}}\,\frac{1}{2}\left\|\boldsymbol{w}_L\right\|^2+\frac{1}{2}\left\|\boldsymbol{w}_Q\right\|^2+\sum_{i=1}^{|\mathcal{Y}|}C^i\left|\boldsymbol{w}^\top\phi(\boldsymbol{x}_c+\boldsymbol{y}^i)-f^i\right|_{\nu^i}, \tag{4.6}$$

where $|\cdot|_\nu$ is the so-called *the linear ν-insensitive loss function* defined by

$$|\boldsymbol{x}|_\nu=\max\left\{0,|\boldsymbol{x}|-\nu\right\},$$

$f^i=f_{\nu^i}(\boldsymbol{x}_c+\boldsymbol{y}^i)$, and C^i, $(i=1,\cdots,|\mathcal{Y}|)$ are non-negative parameters. If we delete $\frac{1}{2}\left\|\boldsymbol{w}_L\right\|^2$ from the objective function in (4.6), then we obtain

$$\underset{\boldsymbol{w}_Q\in\mathbb{R}^{n(n+1)/2}}{\text{minimize}}\,\frac{1}{2}\left\|\boldsymbol{w}_Q\right\|^2+\sum_{i=1}^{|\mathcal{Y}|}C^i\left|\boldsymbol{w}^\top\phi(\boldsymbol{x}_c+\boldsymbol{y}^i)-f^i\right|_{\nu^i}, \tag{4.7}$$

which is equivalent to (4.7).

The merit of using (4.5) is that we can change the number of points in the data set $\mathcal{Y}$ dynamically throughout the optimization algorithm, and can exploit a large amount of information on the objective function that is obtained through the execution of the algorithm. Of course, we must maintain the set $\mathcal{Y}$ so that it satisfies some geometric conditions, in order to ensure the fully-linearity of the model and the boundedness of the Lipschitz constant of the model's gradient.

5 Concluding Remarks

In this paper, we have developed a derivative-free trust-region algorithm that adaptively controls the accuracy of the objective function evaluations. We have given conditions on sample points and their point-wise accuracies under which the model function constructed from those points is guaranteed to be fully-linear. These conditions are imposed only for the current point and its n nearby points. Then, we have proposed a procedure of updating sample points and their accuracies according to those conditions, and established the global convergence of the proposed algorithm. To our knowledge, the proposed algorithm is the first globally convergent algorithm based on derivative-free trust-region algorithm with dynamic accuracy.

For the proof of the global convergence, we have needed the technical conditions (3.19) on $\varepsilon_k^{\mathrm{cur}}$ and $\varepsilon_k^{\mathrm{tri}}$. From the practical point of view, this condition might be satisfied in most cases if we take M to be sufficiently large and ζ to be close to one. However, from the theoretical point of view, it is an open problem whether one can delete the condition (3.19) or replace it with a weaker condition.

This work has focused only on the theoretical property of the derivative-free trust-region method for the problem we consider. We have not verified whether the proposed algorithm is practical, i.e., whether it can find an approximate optimal solution with small computational costs. We hope that this work contributes to future development of efficient algorithms.

Acknowledgments

First of all, I would like to express my sincere appreciation to my supervisor, Associate Professor Nobuo Yamashita. Although I often troubled him, he always kindly looked after me and gave me plenty of precise advice. Without his help and support, I could have never written up this thesis. I am deeply indebted to Professor Masao Fukushima for useful conversations throughout the course of this work. In addition, I would like to thank Assistant Professor Shunsuke Hayashi for his help and advice, especially with our computer environment. I would also like to express my thanks to Emad Hamdy Ahmed and Ong Bun Theang, my colleagues, for many helpful comments on the earlier version of this paper and for helping me improve my English skills. I am very grateful to all members of System Optimization Laboratory for their support and warm friendship during the course. I had good time with them and I will keep a lot of nice memories. Finally, I would like to thank my family for their constant support and encouragement.

References

[1] R. G. Carter, *On the global convergence of trust region algorithms using inexact gradient information*, SIAM Journal on Numerical Analysis, 28 (1991), pp. 251–265.

[2] ——, *Numerical experience with a class of algorithms for nonlinear optimization using inexact function and gradient information*, SIAM Journal on Scientific Computing, 14 (1993), pp. 368–388.

[3] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2000.

[4] A. R. Conn, K. Scheinberg, and P. L. Toint, *Recent progress in unconstrained nonlinear optimization without derivatives*, Mathematical Programming, 79 (1997), pp. 397–414.

[5] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Global convergence of general derivative-free trust-region algorithms to first and second order critical points*, SIAM Journal on Optimization. to appear.

[6] ——, *Geometry of interpolation sets in derivative free optimization*, Mathematical Programming, 111 (2007), pp. 141–172.

[7] ——, *Geometry of sample sets in derivative free optimization. Part II: Polynomial regression and underdetermined interpolation*, IMA Journal of Numerical Analysis, 28 (2008), pp. 721–748.

[8] ——, *Introduction to Derivative-Free Optimization*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2009.

[9] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.

[10] M. Marazzi and J. Nocedal, *Wedge trust region methods for derivative free optimization*, Mathematical Programming, 91 (2002), pp. 289–314.

[11] J. J. Moré and S. M. Wild, *Benchmarking derivative-free optimization algorithms*, Tech. Rep. ANL/MCS-P1471-1207, Argonne National Laboratory, MCS Division, December 2007.

[12] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, second ed., 2006.

[13] R. Oeuvray and M. Bierlaire, *A new derivative-free algorithm for the medical image registration problem*, International Journal of Modelling and Simulation, 27 (2007), pp. 115–124.

[14] M. Powell, *UOBYQA: Unconstrained optimization by quadratic approximation*, Mathematical Programming, 92 (2002), pp. 555–582.

[15] J. Takaki and N. Yamashita, *A derivative-free trust-region algorithm for unconstrained optimization with controllable error*, working paper, 2008.

[16] F. Vanden Berghen and H. Bersini, *CONDOR, a new parallel, constrained extension of powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm*, Journal of Computational and Applied Mathematics, 181 (2005), pp. 157–175.

[17] S. M. Wild, *Derivative-Free Optimization Algorithms for Computationally Expensive Functions*, PhD thesis, Cornell University, 2008.

[18] ——, *MNH: A derivative-free optimization algorithm using minimal norm Hessians*, in Tenth Copper Mountain Conference on Iterative Methods, April 2008.

[19] S. M. Wild, R. G. Regis, and C. A. Shoemaker, *ORBIT: Optimization by radial basis function interpolation in trust-regions*, SIAM Journal on Scientific Computing, 30 (2008), pp. 3197–3219.

A Algorithm for obtaining "well affinely-independent" sample points in the trust-region

Wild [19] proposed the algorithm of generating a set of points $\mathcal{Y} = \{\boldsymbol{y}^1 = 0, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^{n+1}\}$ from given set of candidates $\mathcal{D}_c$ such that the matrix Y defined by $[\boldsymbol{y}^2, \cdots, \boldsymbol{y}^{n+1}] \in \mathbb{R}^{n\times n}$ satisfies Lemma 3.3. The detail of the algorithm is shown below.

Affpoints($\mathcal{D}_c, \theta, \Delta$) [19]:
Algorithm for obtaining "well affinely-independent" sample points in $B(\boldsymbol{x}_c, \Delta) - \boldsymbol{x}_c$

Step 0: Input $\mathcal{D}_c = \{\boldsymbol{d}^1, \cdots \text{ and } \boldsymbol{d}^{|\mathcal{D}_c|}\} \subset \mathbb{R}^n$, constants $\theta \in (0, 1]$ and $\Delta \in (0, \Delta_{\max}]$.

Step 1: Initialize $\mathcal{Y} = \{0\}$, $W = (\operatorname{span} \mathcal{Y})^\perp = \mathbb{R}^n$.

Step 2: For all $\boldsymbol{d}^j \in \mathcal{D}_c$ such that $\left\|\boldsymbol{d}^j\right\| \le \Delta$:
  If $\left\|\operatorname{proj}_W(\frac{1}{\Delta}\boldsymbol{d}^j)\right\| \ge \theta$,
    $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{\boldsymbol{d}^j\}$,
    Update $W = (\operatorname{span} \mathcal{Y})^\perp$.

Step 3a: If $|\mathcal{Y}| = n + 1$, then exit.

Step 3b: If $|\mathcal{Y}| < n + 1$, then
  Calculate an orthonormal basis for W, where $\left\{\boldsymbol{z}^1, \cdots, \boldsymbol{z}^{\dim W}\right\}$ stands for the resulting orthogonal basis.
  Evaluate $f(\boldsymbol{x}_c + \Delta \boldsymbol{z}^i)$, $i = 1, \cdots, \dim W$,
  $\mathcal{Y} \leftarrow \mathcal{Y} \cup \left\{\Delta \boldsymbol{z}^1, \cdots, \Delta \boldsymbol{z}^{\dim W}\right\}$, and exit.

∎

Here, we have used the following notations.

- For a vector $\boldsymbol{d} \in \mathbb{R}^n$ and a subspace W of $\mathbb{R}^n$, $\operatorname{proj}_W(\boldsymbol{d})$ denotes the orthogonal projection of $\boldsymbol{d}$ onto W.
- For a set of vectors $S = \left\{\boldsymbol{d}^1, \cdots, \boldsymbol{d}^m\right\} \subset \mathbb{R}^n (1 \le m \le n)$, $\operatorname{span} S$ denotes the subspace spanned by S.
- $W^\perp$ denotes the orthogonal complement of a subspace W of $\mathbb{R}^n$, i.e., $W^\perp$ is the set of all vectors in V that are orthogonal to every vector in $\mathbb{R}^n$.

When the above algorithm exited with $|\mathcal{Y}| = n + 1$, the points in $\mathcal{Y}$ satisfy $\left\|Y^{-1}\right\| \le \frac{\bar{\Lambda}}{\Delta}$, where the constant $\bar{\Lambda}$ is defined by (3.15).

Remark A.1 *If Δ in Corollary 3.2 (or Lemma 3.3) is chosen to be the current trust-region radis Δ_c, then there might be very few points within $B(\boldsymbol{x}_c; \Delta_c)$ at which f has been evaluated. For this reason, Wild et al. [19] also propose to make m_k fully-linear within an enlarged region defined by $\{\, \boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x} - \boldsymbol{x}_c\| \le \theta_0 \Delta_c \,\}$ for a constant $\theta_0 \ge 1$.*