Comput Optim Appl (2012) 51:481–507
DOI 10.1007/s10589-010-9367-x

On the local convergence of a derivative-free algorithm for least-squares minimization

Hongchao Zhang · Andrew R. Conn

Received: 20 July 2010 / Published online: 23 October 2010

Abstract In Zhang et al. (accepted by SIAM J. Optim., 2010), we developed a class of derivative-free algorithms, called DFLS, for least-squares minimization. Global convergence of the algorithm as well as its excellent numerical performance within a limited computational budget was established and discussed in the same paper. Here we would like to establish the local quadratic convergence of the algorithm for zero residual problems. Asymptotic convergence performance of the algorithm for both zero and nonzero problems is tested. Our numerical experiments indicate that the algorithm is also very promising for achieving high accuracy solutions compared with software packages that do not exploit the special structure of the least-squares problem or that use finite differences to approximate the gradients.

Keywords Derivative-free optimization · Least-squares · Trust region · Levenberg-Marquardt method · System of nonlinear equations · Local convergence · Asymptotic convergence

This material is based upon work supported by the National Science Foundation under Grant 1016204.

H. Zhang (✉)
Department of Mathematics, Louisiana State University, 140 Lockett Hall, Baton Rouge, LA 70803-4918, USA
e-mail: hozhang@math.lsu.edu
url: http://www.math.ufl.edu/~hozhang

A.R. Conn
Department of Mathematical Sciences, IBM T.J. Watson Research Center, Route 134, P.O. Box 218, Yorktown Heights, NY 10598, USA
e-mail: arconn@us.ibm.com

1 Introduction

In [20], we have developed a class of derivative-free algorithms, called DFLS, for the least-squares minimization problem:

$$\min \Phi(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{m} f_i^2(\mathbf{x}) = \frac{1}{2}\|F(\mathbf{x})\|^2, \tag{1.1}$$

where the norm is the standard Euclidian norm, $F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T$ and $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, are general nonlinear twice continuously differentiable functions, but none of their first order or second order derivatives are explicitly available. These algorithms are based on modeling the objective function by multivariate interpolation in combination with trust region techniques. Unlike other model-based methods in the literature, DFLS takes full advantages of the least-squares problem structure by building polynomial interpolation models for each function f_i in the least-squares. One important feature of DFLS is that it applies Powell's minimum Frobenius norm updating technique [15] and typically only uses $2n + 1$ sampling points (significantly less than the $(n + 1)(n + 2)/2$ required for building a fully quadratic model, even for moderate n) to asymptotically build at least fully-linear models for each of the nonlinear functions in the least-squares minimization. Details for fully linear and fully quadratic models are given in [4], Sect. 6.1. Informally speaking fully linear/quadratic models behave like first-order/second-order Taylor series from the point of view of the local truncation errors. Since the same $2n + 1$ interpolation points are used for each function, the computational cost for finding the updates of the coefficients of all the models can be retained within $\mathcal{O}(mn)$. Hence, the total cost of building these models at each iteration is only $\mathcal{O}(mn^2)$, except for the occasional iteration which requires a shift of the model origin, whereupon an extra $\mathcal{O}(mn^3)$ operations are needed.

It is well known that in order to achieve the goal of retaining global convergence and fast local convergence, care needs to be taken in the trust region [1] management and the choice of the sampling points for the model. DFLS combines a generalized Levenberg-Marquardt approach with trust-region techniques to minimize the least-squares residual of the problem. It can adaptively update a single trust-region radius to serve both these purposes, i.e. retaining global convergence and fast local convergence, simultaneously. Global convergence of DFLS have been established in [20] under a trust-region framework. Numerical experiments carried out in the same paper indicate excellent performance of DFLS within a limited computational budget. In this paper, we would like to study its local convergence behavior from both a theoretical and a numerical performance point of view. More specifically, we will establish local quadratic convergence of the algorithm under some local error bound conditions [9, 10, 17, 18]. These conditions are considerably weaker than a non-singularity assumption on the Jacobian [6, 8, 17, 19].

The paper is organized as follows. In Sect. 2, we briefly review the DFLS algorithm and some basic properties of the interpolation models associated with the algorithm. We establish the local quadratic convergence of the algorithm for zero residual problems in Sect. 3. The long-term numerical performances of the algorithm

for achieving high accuracy solutions are given in Sect. 4. Finally, we make some concluding remarks in Sect. 5.

Notation

Unless otherwise specified, the norm $\|\cdot\|$ is the 2-norm for a vector and the induced 2-norm for a matrix. Given any set $\mathbf{S}$, $|\mathbf{S}|$ denotes the cardinality of $\mathbf{S}$. We let $\mathcal{B}$ denote a closed ball in $\mathbb{R}^n$ and $\mathcal{B}(\mathbf{z}, \Delta)$ denote the closed ball centered at z, with radius $\Delta > 0$. $\mathcal{P}_n^d$ means the space of all polynomials of degree $\leq d$ in $\mathbb{R}^n$. We let $C_{n+1}^1 = n + 1$ and $C_{n+2}^2 = (n + 1)(n + 2)/2$ throughout the paper.

There are some constants used in this paper which are denoted by κ (sometimes with acronyms for the subscripts that are meant to be helpful). We collected their definitions here for convenience. The actual meaning of the constants will become clear when each of them is introduced in the paper.

$\hat{\kappa}_{ef}$, κ_{ef}	error in the function value
$\hat{\kappa}_{eg}$, κ_{eg}	error in the gradient
κ_{eH}	error in the Hessian
κ_H^1 , κ_H^2 and κ_H^3	associated with the definition of the model Hessian
κ_m	bound on all the separate models
κ_g	bound on the gradients of all the separate models
κ_H	bound on the Hessians of all the separate models
κ_H^ϕ	bound on the (modified) Hessian of ϕ

2 The algorithm and model properties

For convenience, in this section we give a brief summary of the major results presented in [20]. For a more detailed explanation and motivation, see [20] and the references therein. The following definition of a Λ-poised set $\mathbf{Y} \subset \mathbb{R}^n$ is given in [4].

Definition 2.1 Let $\Lambda > 0$ and $\mathcal{P}$ be a space of polynomials on $\mathbb{R}^n$ with a basis $\varphi = \{\varphi_0(\mathbf{x}), \varphi_1(\mathbf{x}), \ldots, \varphi_p(\mathbf{x})\}$. Then, a set $\mathbf{Y} = \{\mathbf{y}^0, \mathbf{y}^1, \ldots, \mathbf{y}^p\}$ is said to be Λ-poised in $\mathcal{B}$ for $\mathcal{P}$ (in the interpolation sense) if and only if for any $\mathbf{x} \in \mathcal{B} \subset \mathbb{R}^n$ there exists $\lambda(\mathbf{x}) \in \mathbb{R}^{p+1}$ such that:

$$\sum_{i=0}^{p} \lambda_i(\mathbf{x})\varphi(\mathbf{y}^i) = \varphi(\mathbf{x}) \quad \text{with } \|\lambda(\mathbf{x})\|_\infty \leq \Lambda.$$

Another, in some sense equivalent, definition of a Λ-poised set $\mathbf{Y} \subset \mathbb{R}^n$ for minimum Frobenius norm interpolation is given as the following:

Definition 2.2 Let $\Lambda > 0$ and a set $\mathcal{B} \in \mathbb{R}^n$ be given. Let $\varphi = \{\varphi_0(\mathbf{x}), \varphi_1(\mathbf{x}), \ldots, \varphi_{C_{n+2}^2-1}(\mathbf{x})\}$ be the natural basis of monomials of $\mathcal{P}_n^2$ (ordered by degree). A poised set $Y = \{\mathbf{y}^0, \mathbf{y}^1, \ldots, \mathbf{y}^p\}$, with $C_{n+1}^1 \leq p + 1 \leq C_{n+2}^2$, is said to be Λ-poised in $\mathcal{B}$

(in the minimum Frobenius norm sense) if and only if for any $\mathbf{x} \in \mathcal{B}$ there exists a solution $\lambda(\mathbf{x}) \in \mathbb{R}^{p+1}$ of

$$\min \quad \sum_{j=C_{n+1}^1}^{C_{n+2}^2-1} \left(\sum_{i=0}^{p} \lambda_i(\mathbf{x}) \varphi_j(\mathbf{y}_i) - \varphi_j(\mathbf{x}) \right)^2 \tag{2.1}$$

$$\text{such that} \quad \sum_{i=0}^{p} \lambda_i(\mathbf{x}) \varphi_j(\mathbf{y}_i) = \varphi_j(\mathbf{x}), \quad 0 \le j \le n \tag{2.2}$$
$$\text{and} \quad \|\lambda(\mathbf{x})\|_\infty \le \Lambda.$$

In this paper, we call a given set $\mathbf{Y} \subset \mathcal{B}(\mathbf{z}, \Delta)$, with $C_{n+1}^1 \le |\mathbf{Y}| \le C_{n+2}^2$, Λ-poised whenever it is Λ-poised in $\mathcal{B}(\mathbf{z}, \Delta)$ for some polynomial space $\mathcal{P}$ with $\mathcal{P}_n^1 \subseteq \mathcal{P} \subseteq \mathcal{P}_n^2$ in the sense of Definition 2.1. On the other hand, it can also be shown that this just described Λ-poisedness is equivalent to Λ-poisedness in the minimum Frobenius norm sense given by Definition 2.2 using the natural monomial basis, just with a possibly different but related constant $\bar{\Lambda}$ (see Theorem 5.8 in [5]). Hence, when stating that a set $\mathbf{Y} \subset \mathcal{B}(\mathbf{z}, \Delta)$, with $C_{n+1}^1 \le |\mathbf{Y}| \le C_{n+2}^2$, is Λ-poised, one can actually understand it in the sense of either of the two definitions and choose whichever is convenient. More details on the connections and motivations between these two definitions of a Λ-poised set can be found in [4, 5, 20].

Now suppose we have a Λ-poised set $\mathbf{Y} \subset \mathcal{B}(\mathbf{z}, \Delta)$, with $C_{n+1}^1 \le |\mathbf{Y}| \le C_{n+2}^2$. Then for any $\mathbf{y} \in \mathcal{B}(\mathbf{z}, \Delta)$, in [20] we give the local quadratic model $\phi(\mathbf{y}, \mathbf{s})$ of $\Phi(\cdot)$ around $\mathbf{y}$ as

$$\phi(\mathbf{y}, \mathbf{s}) = c_\phi(\mathbf{y}) + \mathbf{g}_\phi(\mathbf{y})^\top \mathbf{s} + \frac{1}{2} \mathbf{s}^\top H_\phi(\mathbf{y}) \mathbf{s}, \tag{2.3}$$

where

$$c_\phi(\mathbf{y}) = \frac{1}{2} \mathbf{m}(\mathbf{y})^\top \mathbf{m}(\mathbf{y}), \qquad \mathbf{g}_\phi(\mathbf{y}) = J(\mathbf{y})^\top \mathbf{m}(\mathbf{y}),$$

$$H_\phi(\mathbf{y}) = \begin{cases} J(\mathbf{y})^\top J(\mathbf{y}) & \text{if } \|\mathbf{g}_\phi(\mathbf{y})\| \ge \kappa_H^1, \\ J(\mathbf{y})^\top J(\mathbf{y}) + \kappa_H^3 \|\mathbf{m}(\mathbf{y})\| I & \text{if } \|\mathbf{g}_\phi(\mathbf{y})\| < \kappa_H^1 \text{ and} \\ & \quad c_\phi(\mathbf{y}) < \kappa_H^2 \|\mathbf{g}_\phi(\mathbf{y})\|, \\ J(\mathbf{y})^\top J(\mathbf{y}) + \sum_{i=1}^m m_i(\mathbf{y}) \nabla^2 m_i & \text{otherwise,} \end{cases}$$

$\mathbf{m}(\mathbf{y}) = (m_1(\mathbf{y}), m_2(\mathbf{y}), \ldots, m_m(\mathbf{y}))^\top$ and $J(\mathbf{y}) = (\nabla m_1(\mathbf{y}), \nabla m_2(\mathbf{y}), \ldots, \nabla m_m(\mathbf{y}))^\top$.

Here, I denotes the identity matrix, κ_H^1, κ_H^2 and κ_H^3 are positive constants, and $m_i(\cdot) \in \mathcal{P}_n^2$, $i = 1, \ldots, m$, are the polynomial interpolating models of $f_i(\cdot)$ on $\mathbf{Y}$.

For completeness and easier later reference, we repeat the description of the algorithm DFLS for least-squares minimization given in [20]. As explained there, this algorithm takes into account the problem structure, but otherwise it is close in spirit to the framework presented in [3], for global convergence, and the detailed algorithm applied in Powell's NEWUOA software [14], for practical efficiency. The details on

how to ensure the interpolating set $\mathbf{Y}$ remains Λ-poised are purposely omitted, since they are readily available from [3] and [4]. Throughout the algorithm stated below, we fix the number of points in the sampling set i.e. $|\mathbf{Y}_k| = N_P$, for all $k \geq 0$, where $N_P \in [C^1_{n+1}, C^2_{n+2}]$ is an integer constant. We denote the resulting iterates by x_k where k is the iteration number.

A Derivative-Free algorithm for Least-Squares minimization (DFLS)

Step 0 **(Initialization)** Choose the starting guess $\mathbf{x}_0$, $0 < \rho_0 \leq \bar{\Delta}_0 \leq \Delta_0 \leq \Delta_{max}$ and N_P, the number of sampling points, with $N_P \geq C^1_{n+1} = n + 1$. Choose an initial set of interpolation points, $\mathbf{Y}_0$, with $\mathbf{x}_0 \in \mathbf{Y}_0 \subset \mathcal{B}(\mathbf{x}_0, \bar{\Delta}_0)$. Choose $\epsilon_\beta \in (0, 1)$ and $\beta > 0$. Set $k = 0$.

Step 1 **(Criticality step)** Choose a base point $\mathbf{y}_k \in \mathbf{Y}_k$ and calculate

$$\mathbf{g}_{\phi_k} = J(\mathbf{y}_k)^{\mathsf{T}} \mathbf{m}(\mathbf{y}_k) + H_\phi(\mathbf{y}_k)(\mathbf{x}_k - \mathbf{y}_k), \tag{2.4}$$

where $\mathbf{y}_k \in \mathbf{Y}_k$. If $\|\mathbf{g}_{\phi_k}\| \leq \epsilon_\beta$, let $\bar{\Delta}^{(0)}_k = \bar{\Delta}_k$; possibly modifying $\mathbf{Y}_k$ as needed to make sure $\mathbf{Y}_k$ is Λ-poised in $\mathcal{B}(\mathbf{x}_k, \bar{\Delta}_k)$, where $\bar{\Delta}_k = \min\{\bar{\Delta}^{(0)}_k, \beta\|\mathbf{g}_{\phi_k}\|\}$, and $\mathbf{g}_{\phi_k}$ is recalculated using (2.4) with the new $\mathbf{Y}_k$ if $\mathbf{Y}_k$ has changed. We will have the determined interpolation model $\mathbf{m}(\mathbf{y}_k)$. It is shown in [2], Lemma 7.3, that unless y_k is a first-order stationary point, Λ-poisedness can be achieved in a finite number of steps.

Step 2 **(Step calculation)** Solve the following trust region subproblem:

$$\begin{aligned} \min \quad & \phi_k(\mathbf{d}) \\ \text{s.t.} \quad & \|\mathbf{d}\| \leq \Delta_k, \end{aligned} \tag{2.5}$$

where $\phi_k(\mathbf{d}) = \phi(\mathbf{y}_k, (\mathbf{x}_k - \mathbf{y}_k) + \mathbf{d})$ with $\phi(\cdot, \cdot)$ defined by (2.3), to obtain the step[1] $\mathbf{d}_k$.

Step 3 **(Safety step)** This step applies only when $\|\mathbf{d}_k\| < \frac{1}{2}\rho_k$ and $\|\mathbf{g}_{\phi_k}\| > \epsilon_\beta$.

3.1 Let $i = 0$, $\Delta^{(0)}_k = \Delta_k$.

3.2 Choose $\bar{\Delta}^{(i)}_k \in [\rho_k, \Delta^{(i)}_k]$.

3.3 If $\bar{\Delta}^{(i)}_k > \rho_k$, then

If $\mathbf{Y}_k$ is Λ-poised in $\mathcal{B}(\mathbf{x}_k, \bar{\Delta}^{(i)}_k)$, then
Let $\Delta^{(i+1)}_k = \max\{\Delta^{(i)}_k/10, \rho_k\}$. $i = i + 1$, go to 3.2.
Else
Let $\Delta_{k+1} = \bar{\Delta}^{(i)}_k$, $\rho_{k+1} = \rho_k$.
Endif

Else (i.e. $\bar{\Delta}^{(i)}_k = \rho_k$)

If $\mathbf{Y}_k$ is Λ-poised in $\mathcal{B}(\mathbf{x}_k, \bar{\Delta}^{(i)}_k)$, then

[1] In other words we build the model at y_k but shift our center to x_k.

Let $\Delta_{k+1} = \rho_k/2$, $\rho_{k+1} = \rho_k/10$.
Else
Let $\Delta_{k+1} = \bar{\Delta}_k^{(i)}$, $\rho_{k+1} = \rho_k$.
Endif

Endif
Let $\mathbf{x}_{k+1} = \mathbf{x}_k$ and choose $\bar{\Delta}_{k+1} \in [\rho_{k+1}, \Delta_{k+1}]$.
Modify $\mathbf{Y}_k$ to form $\mathbf{Y}_{k+1}$ such that $\mathbf{Y}_{k+1}$ is Λ-poised in $\mathcal{B}(\mathbf{x}_{k+1}, \bar{\Delta}_{k+1})$.
Set $k = k + 1$, go to Step 1.

Step 4 **(Acceptance of the trial step)** This step applies only when $\|\mathbf{d}_k\| \geq \frac{1}{2}\rho_k$ or $\|\mathbf{g}_{\phi_k}\| \leq \epsilon_\beta$ (so that Step 3 is not invoked).
Compute $\Phi(\mathbf{x}_k + \mathbf{d}_k)$ and $r_k := Ared_k/Pred_k$, where the actual reduction is defined by

$$Ared_k = \Phi(\mathbf{x}_k) - \Phi(\mathbf{x}_k + \mathbf{d}_k), \tag{2.6}$$

while the predicted reduction is defined by

$$Pred_k = \phi_k(\mathbf{0}) - \phi_k(\mathbf{d}_k). \tag{2.7}$$

If $r_k > 0$, then $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k$; otherwise, $\mathbf{x}_{k+1} = \mathbf{x}_k$.

Step 5 **(Trust region radius update)** Set[2]

$$\tilde{\Delta}_{k+1} = \begin{cases} \frac{1}{2}\|\mathbf{d}_k\| & \text{if } r_k < 0.1, \\ \max\{\frac{1}{2}\Delta_k, \|\mathbf{d}_k\|\} & \text{if } 0.1 \leq r_k < 0.7, \\ \max\{\Delta_k, 2\|\mathbf{d}_k\|\} & \text{if } r_k \geq 0.7, \end{cases} \tag{2.8}$$

and

$$\Delta_{k+1} = \min\{\max\{\tilde{\Delta}_{k+1}, \rho_k\}, \Delta_{max}\}. \tag{2.9}$$

If $r_k \geq 0.1$, then

Let $\rho_{k+1} = \rho_k$ and choose $\bar{\Delta}_{k+1} \in [\rho_k, \Delta_{k+1}]$.
The interpolating points set $\mathbf{Y}_k$ is updated to take into consideration the new point $\mathbf{x}_{k+1}$ to form $\mathbf{Y}_{k+1} \in \mathcal{B}(\mathbf{x}_{k+1}, \bar{\Delta}_{k+1})$.
Set $k = k + 1$, go to Step 1.

Endif

Step 6 **(Model improvement)** This step applies only when $r_k < 0.1$.
If $\mathbf{Y}_k$ is Λ-poised in $\mathcal{B}(\mathbf{x}_k, \Delta_k)$ and $\Delta_k = \rho_k$, then

Let $\Delta_{k+1} = \rho_k/2$ and $\rho_{k+1} = \rho_k/10$. Choose $\bar{\Delta}_{k+1} \in [\rho_{k+1}, \Delta_{k+1}]$.
The interpolating points set $\mathbf{Y}_k$ is updated to take into consideration the new point $\mathbf{x}_{k+1}$ to form $\mathbf{Y}_{k+1} \in \mathcal{B}(\mathbf{x}_{k+1}, \bar{\Delta}_{k+1})$.

Else

[2]Note that the values 0.1 and 0.7 are somewhat arbitrary, except that the first must be less than the second and usually the first is much smaller than the second.

Let $\rho_{k+1} = \rho_k$. Choose $\bar{\Delta}_{k+1} \in [\rho_{k+1}, \Delta_{k+1}]$. Possibly modification of $\mathbf{Y}_k$ is needed to form $\mathbf{Y}_{k+1}$ such that $\mathbf{Y}_{k+1}$ is Λ-poised in $\mathcal{B}(\mathbf{x}_{k+1}, \bar{\Delta}_{k+1})$. $\Delta_{k+1} = \Delta_k$

Endif
Set $k = k + 1$, go to Step 1.

To show local quadratic convergence, we need the following lemmas and assumptions, which were given in [20]. We restate them for convenience. The following lemma shows that when the sampling set $\mathbf{Y}$ is Λ-poised (by either definition), an interpolating polynomial on $\mathbf{Y}$ would be at least a local fully linear model. This lemma follows directly from Theorem 5.4 in [4].

Lemma 2.3 *Given any $\Delta > 0$, $\mathbf{z} \in \mathbb{R}^n$ and $\mathbf{Y} = \{\mathbf{y}^0, \mathbf{y}^1, \ldots, \mathbf{y}^p\} \subset \mathcal{B}(\mathbf{z}, \Delta)$ Λ-poised in $\mathcal{B}(\mathbf{z}, \Delta)$ with $C^1_{n+1} \le |\mathbf{Y}| \le C^2_{n+2}$, let $m(\cdot) \in \mathcal{P}^2_n$ be an interpolating polynomial of f on $\mathbf{Y}$, i.e.*

$$m(\mathbf{y}^i) = f(\mathbf{y}^i), \quad i = 1, \ldots, |\mathbf{Y}|.$$

If $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and ∇f is Lipschitz continuous, with Lipschitz constant L, in an open set containing $\mathcal{B}(\mathbf{z}, \Delta)$, then for any $\mathbf{s} \in \mathcal{B}(\mathbf{0}, \Delta)$ we have

$$\|\nabla f(\mathbf{z}+\mathbf{s}) - \nabla m(\mathbf{z}+\mathbf{s})\| \le \hat{\kappa}_{eg}(n, \Lambda)(\|\nabla^2 m\| + L)\Delta,$$
$$|f(\mathbf{z}+\mathbf{s}) - m(\mathbf{z}+\mathbf{s})| \le \hat{\kappa}_{ef}(n, \Lambda)(\|\nabla^2 m\| + L)\Delta^2,$$

where $\hat{\kappa}_{eg}$ and $\hat{\kappa}_{ef}$ are positive constants depending only on n and Λ.

Given a Λ-poised set $\mathbf{Y} \subset \mathcal{B}(\mathbf{z}, \Delta)$ with $C^1_{n+1} \le |\mathbf{Y}| \le C^2_{n+2}$, for each $i = 1, \ldots, m$, suppose $m_i(\mathbf{x}) \in \mathcal{P}^2_n$ is a polynomial interpolating model of $f_i(\mathbf{x})$ on $\mathbf{Y}$. Then, based on the above lemma, there exist positive constants κ_{eg} and κ_{ef} such that for any $\mathbf{s} \in \mathcal{B}(\mathbf{0}, \Delta)$,

$$\|\nabla f_i(\mathbf{z}+\mathbf{s}) - \nabla m_i(\mathbf{z}+\mathbf{s})\| \le \kappa_{eg}\Delta, \tag{2.10}$$
$$|f_i(\mathbf{z}+\mathbf{s}) - m_i(\mathbf{z}+\mathbf{s})| \le \kappa_{ef}\Delta^2, \tag{2.11}$$

for all $i = 1, \ldots, m$, where κ_{eg} and κ_{ef} are positive constants depending only on n, Λ, F and $\max\{\|\nabla^2 m_i\|, i = 1, \ldots, m\}$. In particular, κ_{eg} and κ_{ef} depend neither on $\mathbf{z}$ nor Δ.

Now, define $conv(\mathcal{L}_{enl}(\mathbf{x}_0))$ to be the convex hull of $\mathcal{L}_{enl}(\mathbf{x}_0)$ with

$$\mathcal{L}_{enl}(\mathbf{x}_0) = \bigcup_{\mathbf{x} \in \mathcal{L}(\mathbf{x}_0)} \mathcal{B}(\mathbf{x}, \Delta_{max}) \quad \text{and} \quad \mathcal{L}(\mathbf{x}_0) = \{\mathbf{x} \in \mathbb{R}^n : \Phi(\mathbf{x}) \le \Phi(\mathbf{x}_0)\},$$

where Φ is defined in (1.1). Throughout this paper we also need the following two assumptions.

Assumption 2.1 The level set $\mathcal{L}(\mathbf{x}_0) = \{\mathbf{x} \in \mathbb{R}^n : \Phi(\mathbf{x}) \le \Phi(\mathbf{x}_0)\}$ is bounded.

Assumption 2.2 There exists a constant κ_H, independent of the iteration number k in the DFLS algorithm, such that if m_i, $i = 1, \ldots, m$, is the polynomial interpolating model of f_i on a Λ-poised sampling set $\mathbf{Y}_k$ constructed as in the *DFLS* algorithm, then

$$\|\nabla^2 m_i\| \leq \kappa_H, \tag{2.12}$$

for all $i = 1, \ldots, m$.

Based on Assumption 2.1 and the fact that F is assumed twice continuously differentiable, although none of their first order or second order derivatives are explicitly available, we have the following lemmas.

Lemma 2.4 *Under Assumption 2.1, there exist positive constants L_0, L_1 and L_2, such that*

$$\|F(\mathbf{x})\| \leq L_0, \qquad \|F(\mathbf{x}) - F(\mathbf{y})\| \leq L_1 \|\mathbf{x} - \mathbf{y}\| \quad \text{and} \quad \|\nabla F(\mathbf{x})\| \leq L_1,$$
$$\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq L_2 \|\mathbf{x} - \mathbf{y}\| \quad \text{and} \quad \|\nabla^2 f_i(\mathbf{x})\| \leq L_2, \quad i = 1, \ldots, m,$$

for any $\mathbf{x}, \mathbf{y} \in conv(\mathcal{L}_{enl}(\mathbf{x}_0))$.

With Assumptions 2.1 and 2.2, by (2.10) and (2.11) the following bounds can be established for the models used in the DFLS algorithm.

Lemma 2.5 *Under Assumptions 2.1 and 2.2, there exist constants κ_m and κ_g, independent of k, such that if m_i, $i = 1, \ldots, m$, is the polynomial interpolating model of f_i on a Λ-posed sampling set $\mathbf{Y}_k$ constructed as in the DFLS algorithm, then for any $\mathbf{y} \in \mathbf{Y}_k$,*

$$|m_i(\mathbf{y})| \leq \kappa_m \quad \text{and} \quad \|\nabla m_i(\mathbf{y})\| \leq \kappa_g, \tag{2.13}$$

for all $i = 1, \ldots, m$. Hence, there exists a constant κ_H^ϕ such that

$$\|H_\phi(\mathbf{y})\| \leq \kappa_H^\phi, \tag{2.14}$$

for any $\mathbf{y} \in \mathbf{Y}_k$.

3 Local convergence

In this section, we discuss the local convergence properties of the DFLS algorithm for zero residual problems under a certain type of nonsingularity condition. Consequently, in this section we have the following additional assumptions on the objective function (1.1).

Assumption 3.1 (I) We assume (1.1) is a zero residual problem, i.e.

$$\Phi(\mathbf{x}^*) = 0, \quad \text{for any } \mathbf{x}^* \in \mathbf{X}^*,$$

where $\mathbf{X}^* \in \mathbb{R}^n$ is the solution set of (1.1).

(II) $\|F(\mathbf{x})\|$ provides a local error bound on $\mathbf{X}^*$, i.e. there exist positive constants α and ρ such that

$$\|F(\mathbf{x})\| \geq \alpha \; dist(\mathbf{x}, \mathbf{X}^*), \quad \text{whenever} \quad dist(\mathbf{x}, \mathbf{X}^*) \leq \rho, \tag{3.1}$$

where $dist(\mathbf{x}, \mathbf{X}^*) = \inf_{\mathbf{y} \in \mathbf{X}^*} \|\mathbf{x} - \mathbf{y}\|$.

Since $\Phi(\mathbf{x})$ is continuous, we know $\mathbf{X}^*$ is closed. Hence, it follows from $\mathbf{X}^*$ being nonempty that for any $\mathbf{x} \in \mathbb{R}^n$, there exists a $\bar{\mathbf{x}} \in \mathbf{X}^*$ such that

$$dist(\mathbf{x}, \mathbf{X}^*) = \min_{\mathbf{y} \in \mathbf{X}^*} \|\mathbf{x} - \mathbf{y}\| = \|\bar{\mathbf{x}} - \mathbf{x}\|. \tag{3.2}$$

In Assumption 3.1, the local error bound condition (II), first proposed in [17, 18], is a particular generalization of the nonsingularity assumption of the Jacobian at the solution, as is readily seen by linearising F about x^*. Similar conditions have subsequently been widely used and studied in [7–10, 19]. When $\nabla F(\mathbf{x}^*)$ is nonsingular at a solution $\mathbf{x}^*$, $\mathbf{x}^*$ is an isolated solution. Hence, $\|F(\mathbf{x})\|$ provides a local error bound at $\mathbf{x}^*$. However, $\nabla F(\mathbf{x}^*)$ might be singular but nevertheless $\|F(\mathbf{x})\|$ may provide a local error bound at $\mathbf{x}^*$. One can refer to the examples provided in [6] and [19]. It is well known that the Levenberg-Marquardt method has local quadratic convergence when the Jacobian at $\mathbf{x}^*$ is nonsingular. In the following we will first show, under proper conditions and assumptions, that a certain class of DFLS algorithms will eventually reduce to a "regularized Levenberg-Marquardt-type" algorithm and then we can show the local convergence rate is quadratic, even without the nonsingularity assumption of the Jacobian at the solution.

For different choices of the base point $\mathbf{y}_k$ in the DFLS algorithm, we have different algorithms. In this section, with respect to local convergence properties, we consider a subclass of DFLS algorithms which makes a specific choice for the base points. Hence, in this section we make the following assumption on the choice of base points.

Assumption 3.2 For sufficiently large k, we choose the base point $\mathbf{y}_k \in \mathbf{Y}_k$ to be $\mathbf{x}_k$, the current iterate (which corresponds to the lowest objective function value seen to date), i.e. for k sufficiently large, we choose

$$\mathbf{y}_k = \mathbf{x}_k.$$

To discuss the local convergence properties of the DFLS algorithm, we first assume that the sequence $\{\mathbf{x}_k\}$ generated by the algorithm converges to the solution set $\mathbf{X}^*$. Then, we show it converges to a solution $\mathbf{x}^* \in \mathbf{X}^*$ quadratically. Thus, we have the following assumption.

Assumption 3.3 Assume the sequence of $\{\mathbf{x}_k\}$ generated by the DFLS algorithm converges to the solution set $\mathbf{X}^*$, i.e.

$$\lim_{k \to \infty} dist(\mathbf{x}_k, \mathbf{X}^*) = 0.$$

We begin the discussion with the following lemma.

Lemma 3.1 *Under Assumptions 2.1, 2.2, 3.1, 3.2, and 3.3, we have that*

$$\lim_{k\to\infty} \|\mathbf{g}_{\phi_k}\| = 0, \tag{3.3}$$

and there exists a constant $L_3 > 0$ such that

$$\begin{aligned} \|J(\mathbf{x}_k) - \nabla F(\mathbf{x}_k)\| &= \|\nabla \mathbf{m}(\mathbf{x}_k) - \nabla F(\mathbf{x}_k)\| \\ &\le \bar{M}\|\mathbf{g}_\phi(\mathbf{x}_k)\| \le L_3\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|, \end{aligned} \tag{3.4}$$

for all large k, where $\bar{M} := m\beta\kappa_{eg}$, with κ_{eg} and β defined in (2.10) and Step 0 of the algorithm, respectively. Here and in what follows $\bar{\mathbf{x}}_k$ is defined as in (3.2).

Proof By Assumptions 3.1 and 3.3, we have

$$\lim_{k\to\infty} \|F(\mathbf{x}_k)\| = 0. \tag{3.5}$$

Since $\mathbf{y}_k = \mathbf{x}_k$ for all sufficiently large k, by Assumption 3.2, it follows from Lemma 2.5 that

$$\|\mathbf{g}_{\phi_k}\| = \|\mathbf{g}_\phi(\mathbf{x}_k)\| = \|J(\mathbf{x}_k)^{\mathsf{T}}\mathbf{m}(\mathbf{x}_k)\| \le m\kappa_g\|F(\mathbf{x}_k)\|. \tag{3.6}$$

Hence, by (3.5), we have (3.3) holds.

Now, by Assumption 3.3, it follows from Step 1 (Criticality step) of the DFLS algorithm that $\mathbf{Y}_k$ is Λ-poised in $\mathcal{B}(\mathbf{x}_k, \beta\|\mathbf{g}_{\phi_k}\|)$. Since $\mathbf{g}_{\phi_k} = \mathbf{g}_\phi(\mathbf{x}_k) = J(\mathbf{x}_k)^{\mathsf{T}}\mathbf{m}(\mathbf{x}_k)$, by (2.10), (3.6) and Lemma 2.4, we have

$$\begin{aligned} \|J(\mathbf{x}_k) - \nabla F(\mathbf{x}_k)\| &= \|\nabla \mathbf{m}(\mathbf{x}_k) - \nabla F(\mathbf{x}_k)\| \\ &\le \sum_{i=1}^{m} \|\nabla m_i(\mathbf{x}_k) - \nabla f_i(\mathbf{x}_k)\| \\ &\le m\kappa_{eg}\beta\|\mathbf{g}_{\phi_k}\| \le m^2\kappa_{eg}\beta\kappa_g\|F(\mathbf{x}_k)\| \\ &\le m^2\kappa_{eg}\beta\kappa_g L_1\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| := L_3\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|, \end{aligned}$$

where $L_3 = m^2\kappa_{eg}\beta\kappa_g L_1$ and β is defined in Step 0 of the algorithm. □

The following lemma shows that, in the zero residual case, the regularized Hessian (i.e. the middle term in the definition of H_ϕ) will eventually be chosen by the DFLS algorithm for building the trust region subproblem (2.5).

Lemma 3.2 *Under Assumptions 2.1, 2.2, 3.1, 3.2, and 3.3, for any $\kappa_H^2 > 0$, if k is sufficiently large, then*

$$c_\phi(\mathbf{x}_k) \le \kappa_H^2\|\mathbf{g}_\phi(\mathbf{x}_k)\|, \tag{3.7}$$

and for $\phi_k(\cdot)$ defined by (2.5) and $\phi(\cdot,\cdot)$ defined by (2.3), we have

$$\begin{aligned}\phi_k(\mathbf{d}) = \phi(\mathbf{x}_k, \mathbf{d}) &= \frac{1}{2}\|\mathbf{m}(\mathbf{x}_k) + J(\mathbf{x}_k)\mathbf{d}\|^2 + \lambda_k\|\mathbf{d}\|^2 \\ &= \frac{1}{2}\|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}\|^2 + \lambda_k\|\mathbf{d}\|^2, \end{aligned} \tag{3.8}$$

where $\lambda_k = \frac{1}{2}\kappa_H^3\|\mathbf{m}(\mathbf{x}_k)\|$, with κ_H^2 and κ_H^3 given as in the definition of H_ϕ.

Proof First, we assume k is large enough that Lemma 3.1 holds and $\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| = \delta$ is sufficiently small.

Since $\|F(\mathbf{x})\|$ provides a local error bound on $\mathbf{X}^*$, it follows from (3.1) that for sufficiently large k, or equivalently δ sufficiently small, $\|F(\mathbf{x}_k)\| \geq \alpha\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|$ for some $\alpha > 0$. Therefore, noticing $\Phi(\bar{\mathbf{x}}_k) = 0$ and $\nabla\Phi(\bar{\mathbf{x}}_k) = \mathbf{0}$, we have

$$\begin{aligned}\frac{1}{2}\alpha^2\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|^2 \leq \frac{1}{2}\|F(\mathbf{x}_k)\|^2 &= \Phi(\mathbf{x}_k) \\ &= \frac{1}{2}(\mathbf{x}_k - \bar{\mathbf{x}}_k)^{\mathsf{T}}\bar{H}(\mathbf{x}_k - \bar{\mathbf{x}}_k) + R_2(\mathbf{x}_k, \bar{\mathbf{x}}_k), \end{aligned} \tag{3.9}$$

where $\bar{H} = \nabla^2\Phi(\bar{\mathbf{x}}_k)$ is the Hessian at $\bar{\mathbf{x}}_k$ and R_2 is the remainder term. By choosing δ smaller if necessary, we know

$$|R_2(\mathbf{x}_k, \bar{\mathbf{x}}_k)| \leq \frac{\alpha^2}{3}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|^2.$$

So, in this case, (3.9) gives

$$\frac{\alpha^2}{3}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|^2 \leq (\mathbf{x}_k - \bar{\mathbf{x}}_k)^{\mathsf{T}}\bar{H}(\mathbf{x}_k - \bar{\mathbf{x}}_k) \leq \|\mathbf{x}_k - \bar{\mathbf{x}}_k\|\,\|\bar{H}(\mathbf{x}_k - \bar{\mathbf{x}}_k)\|.$$

Hence, we have

$$\frac{\alpha^2}{3}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| \leq \|\bar{H}(\mathbf{x}_k - \bar{\mathbf{x}}_k)\|. \tag{3.10}$$

On the other hand,

$$\nabla\Phi(\mathbf{x}_k) = \nabla\Phi(\mathbf{x}_k) - \nabla\Phi(\bar{\mathbf{x}}_k) = \bar{H}(\mathbf{x}_k - \bar{\mathbf{x}}_k) + R_1(\mathbf{x}_k, \bar{\mathbf{x}}_k), \tag{3.11}$$

where R_1 is the remainder term. Choose k sufficiently large so that

$$|R_1(\mathbf{x}_k, \bar{\mathbf{x}}_k)| \leq \frac{\alpha^2}{6}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|. \tag{3.12}$$

Combining (3.10)–(3.12), we have

$$\|\nabla F(\mathbf{x}_k)^{\mathsf{T}}F(\mathbf{x}_k)\| = \|\nabla\Phi(\mathbf{x}_k)\| \geq \frac{\alpha^2}{6}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|. \tag{3.13}$$

Thus, by Assumption 2.2, Lemma 2.4 and (3.13), for δ small,

$$\begin{aligned}\frac{c_\phi(\mathbf{x}_k)}{\|\nabla F(\mathbf{x}_k)^\mathsf{T} F(\mathbf{x}_k)\|} &= \frac{1}{2}\frac{\mathbf{m}(\mathbf{x}_k)^\mathsf{T}\mathbf{m}(\mathbf{x}_k)}{\|\nabla F(\mathbf{x}_k)^\mathsf{T} F(\mathbf{x}_k)\|}\\ &= \frac{1}{2}\frac{F(\mathbf{x}_k)^\mathsf{T} F(\mathbf{x}_k)}{\|\nabla F(\mathbf{x}_k)^\mathsf{T} F(\mathbf{x}_k)\|}\\ &\le \frac{L_1^2\|\mathbf{x}_k-\bar{\mathbf{x}}_k\|^2}{2\|\nabla F(\mathbf{x}_k)^\mathsf{T} F(\mathbf{x}_k)\|}\\ &\le \frac{3L_1^2}{\alpha^2}\|\mathbf{x}_k-\bar{\mathbf{x}}_k\|. \end{aligned} \tag{3.14}$$

Now, by (3.4), we have

$$\begin{aligned}\frac{\|\nabla F(\mathbf{x}_k)^\mathsf{T} F(\mathbf{x}_k) - J(\mathbf{x_k})^\mathsf{T}\mathbf{m}(\mathbf{x}_k)\|}{\|J(\mathbf{x_k})^\mathsf{T}\mathbf{m}(\mathbf{x}_k)\|} &= \frac{\|(\nabla F(\mathbf{x}_k) - J(\mathbf{x_k}))^\mathsf{T} F(\mathbf{x}_k)\|}{\|J(\mathbf{x_k})^\mathsf{T}\mathbf{m}(\mathbf{x}_k)\|}\\ &\le \frac{\bar{M}\|\mathbf{g}_\phi(\mathbf{x}_k)\|\|F(\mathbf{x}_k)\|}{\|\mathbf{g}_\phi(\mathbf{x}_k)\|}\\ &= \bar{M}\|F(\mathbf{x}_k)\| \le \bar{M}L_1\|\mathbf{x}_k-\bar{\mathbf{x}}_k\|,\end{aligned}$$

where $\bar{M}$ is the constant defined in (3.4). Hence,

$$1-\bar{M}L_1\|\mathbf{x}_k-\bar{\mathbf{x}}_k\| \le \frac{\|\nabla F(\mathbf{x}_k)^\mathsf{T} F(\mathbf{x}_k)\|}{\|J(\mathbf{x_k})^\mathsf{T}\mathbf{m}(\mathbf{x}_k)\|} \le 1+\bar{M}L_1\|\mathbf{x}_k-\bar{\mathbf{x}}_k\|. \tag{3.15}$$

Therefore, by (3.14) and (3.15), for δ small, we have

$$c_\phi(\mathbf{x}_k) \le c\delta\|\mathbf{g}_\phi(\mathbf{x}_k)\|, \tag{3.16}$$

where $c>0$ is some constant. Hence, for any $\kappa_H^2>0$, if k is sufficiently large, δ will be sufficiently small, and therefore (3.7) holds. Noticing, $\mathbf{g}_{\phi_k}=\mathbf{g}_\phi(\mathbf{x}_k)$ and $\|\mathbf{g}_{\phi_k}\| \le \kappa_H^1$ for all large k, by (3.7) and the definition of $\phi(\cdot,\cdot)$, (3.8) follows immediately. □

Now, for the trust region step size, we have the following lemma.

Lemma 3.3 *Under Assumptions 2.1, 2.2, 3.1, 3.2, and 3.3, we have*

$$\|\mathbf{d}_k\| \le 2\,dist(\mathbf{x}_k,\mathbf{X}^*) = 2\|\bar{\mathbf{x}}_k-\mathbf{x}_k\|, \tag{3.17}$$

for k sufficiently large.

Proof First, we assume k is large enough that Lemma 3.1 and Lemma 3.2 hold, and $\|\mathbf{x}_k-\bar{\mathbf{x}}_k\|$ is sufficiently small.

We only need to consider the case $\|\bar{\mathbf{x}}_k-\mathbf{x}_k\| \le \|\mathbf{d}_k\|$. In this case, $\bar{\mathbf{x}}_k-\mathbf{x}_k$ is a feasible point of the trust region subproblem (2.5). Hence, denoting $\mathbf{s}_k=\bar{\mathbf{x}}_k-\mathbf{x}_k$, by

Lemma 2.4, (3.1), (3.4) and (3.8), choosing δ small if necessary, we have

$$\begin{aligned}
\|\mathbf{d}_k\|^2 &\le \frac{2}{\kappa_H^3\|\mathbf{m}(\mathbf{x}_k)\|}\phi_k(\bar{\mathbf{x}}_k-\mathbf{x}_k)\\
&= \frac{1}{\kappa_H^3\|F(\mathbf{x}_k)\|}\|F(\mathbf{x}_k)+\nabla\mathbf{m}(\mathbf{x}_k)\mathbf{s}_k\|^2+\|\mathbf{s}_k\|^2\\
&\le \frac{1}{\kappa_H^3\|F(\mathbf{x}_k)\|}(\|F(\mathbf{x}_k)+\nabla F(\mathbf{x}_k)\mathbf{s}_k\|+\|(\nabla\mathbf{m}(\mathbf{x}_k)-\nabla F(\mathbf{x}_k))\mathbf{s}_k\|)^2+\|\mathbf{s}_k\|^2\\
&\le \frac{1}{\kappa_H^3\|F(\mathbf{x}_k)\|}\left(\frac{m}{2}L_2\|\mathbf{s}_k\|^2+L_3\|\mathbf{s}_k\|^2\right)^2+\|\mathbf{s}_k\|^2\\
&\le \frac{1}{\kappa_H^3\alpha\|\mathbf{s}_k\|}\left(\frac{m}{2}L_2\|\mathbf{s}_k\|^2+L_3\|\mathbf{s}_k\|^2\right)^2+\|\mathbf{s}_k\|^2\\
&= \frac{(\frac{m}{2}L_2+L_3)^2}{\kappa_H^3\alpha}\|\mathbf{s}_k\|^3+\|\mathbf{s}_k\|^2\\
&\le 2\|\mathbf{s}_k\|^2=2\|\bar{\mathbf{x}}_k-\mathbf{x}_k\|^2.
\end{aligned}$$

[Note that for the third inequality, we have used $0=f_i(\bar{\mathbf{x}}_k)=f_i(x_k)+\nabla f_i(x_k)^\top s_k+\frac{1}{2}s_k^\top\nabla^2 f_i(\xi_k)s_k$ and thus $\|f_i(x_k)+\nabla f_i(x_k)^\top s_k\|\le\frac{1}{2}L_2\|s_k\|^2$.] □

Lemma 3.4 *Under Assumptions 2.1, 2.2, 3.1, 3.2, and 3.3, for any $\epsilon>0$, if k is sufficiently large, then*

$$|r_k-1|\le\epsilon. \tag{3.18}$$

In addition, there exists a constant $\bar{\Delta}>0$ such that

$$\Delta_k\ge\bar{\Delta},\quad \textit{for all } k\ge 0. \tag{3.19}$$

Proof First, we assume k is large enough that Lemma 3.1, Lemma 3.2 and Lemma 3.3 hold, and $\|\mathbf{x}_k-\bar{\mathbf{x}}_k\|$ is sufficiently small. Let $\mathbf{s}_k=\bar{\mathbf{x}}_k-\mathbf{x}_k$.

Now, we first prove

$$\|\mathbf{m}(\mathbf{x}_k)\|-\|\mathbf{m}(\mathbf{x}_k)+\nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\|\ge c_3\|\mathbf{d}_k\|, \tag{3.20}$$

when k is sufficiently large, where $c_3>0$ is a constant.

Case I, $\|\mathbf{s}_k\|\le\|\mathbf{d}_k\|$. In this case, $\mathbf{s}_k$ is a feasible point of (2.5) since $\|\mathbf{s}_k\|\le\|\mathbf{d}_k\|$. Because the trust region subproblem has the form (2.5), $\mathbf{d}_k$ is a solution of this subproblem, (3.8) holds for k sufficiently large, and $\mathbf{s}_k$ is feasible, we have

$$\frac{1}{2}\|\mathbf{m}(\mathbf{x}_k)+\nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\|^2+\lambda_k\|\mathbf{d}_k\|^2\le\frac{1}{2}\|\mathbf{m}(\mathbf{x}_k)+\nabla\mathbf{m}(\mathbf{x}_k)\mathbf{s}_k\|^2+\lambda_k\|\mathbf{s}_k\|^2.$$

So, it follows from $\|\mathbf{s}_k\|\le\|\mathbf{d}_k\|$ that $\|\mathbf{m}(\mathbf{x}_k)+\nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\|\le\|\mathbf{m}(\mathbf{x}_k)+\nabla\mathbf{m}(\mathbf{x}_k)\mathbf{s}_k\|$. Now, it follows from the above inequality, (3.1), (3.4) and (3.17) and $\|\mathbf{s}_k\|$ sufficiently

small that

$$\begin{aligned}\|\mathbf{m}(\mathbf{x}_k)\| - \|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\| &\geq \|\mathbf{m}(\mathbf{x}_k)\| - \|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{s}_k\| \\ &\geq \|\mathbf{m}(\mathbf{x}_k)\| - \|\mathbf{m}(\mathbf{x}_k) + \nabla F(\mathbf{x}_k)\mathbf{s}_k\| \\ &\quad - \|(\nabla\mathbf{m}(\mathbf{x}_k) - \nabla F(\mathbf{x}_k))\mathbf{s}_k\| \\ &\geq \|\mathbf{m}(\mathbf{x}_k)\| - \frac{m}{2}L_2\|\mathbf{s}_k\|^2 - L_3\|\mathbf{s}_k\|^2 \\ &\geq \alpha\|\mathbf{s}_k\| - \left(\frac{m}{2}L_2 + L_3\right)\|\mathbf{s}_k\|^2 \\ &\geq \frac{\alpha}{2}\|\mathbf{s}_k\| \geq \frac{\alpha}{4}\|\mathbf{d}_k\|,\end{aligned}$$

where the third inequality again used the note in the proof of Lemma 3.3 and the last inequality used (3.17).

Case II, $\|\mathbf{s}_k\| > \|\mathbf{d}_k\|$. In this case, $\mathbf{t}_k := (\|\mathbf{d}_k\|/\|\mathbf{s}_k\|)\mathbf{s}_k$ is a feasible point of (2.5) and then by (3.1) and (3.4), we have that

$$\begin{aligned}\|\mathbf{m}(\mathbf{x}_k)\| - \|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\| &\geq \|\mathbf{m}(\mathbf{x}_k)\| - \|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{t}_k\| \\ &\geq \frac{\|\mathbf{d}_k\|}{\|\mathbf{s}_k\|}(\|F(\mathbf{x}_k)\| - \|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{s}_k\|) \\ &\geq \frac{\|\mathbf{d}_k\|}{\|\mathbf{s}_k\|}\left(\alpha\|\mathbf{s}_k\| - \left(\frac{m}{2}L_2 + L_3\right)\|\mathbf{s}_k\|^2\right) \\ &\geq \frac{\alpha}{2}\|\mathbf{d}_k\|,\end{aligned}$$

for k sufficiently large, where we used the same argument as in Case I to establish the penultimate inequality.

From Case I and II, we know (3.20) holds with $c_3 = \alpha/4$. Now, by (2.7), (3.8), (3.17) and (3.20), we have

$$\begin{aligned}Pred_k &= \phi_k(\mathbf{0}) - \phi_k(\mathbf{d}_k) \\ &= \frac{1}{2}(\|\mathbf{m}(\mathbf{x}_k)\|^2 - \|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\|^2 - \kappa_H^3\|\mathbf{m}(\mathbf{x}_k)\|\|\mathbf{d}_k\|^2) \\ &\geq \frac{1}{2}\|F(\mathbf{x}_k)\|(\|F(\mathbf{x}_k)\| - \|F(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\| - \kappa_H^3\|\mathbf{d}_k\|^2) \\ &\geq \frac{1}{2}\|F(\mathbf{x}_k)\|\left(\frac{\alpha}{4}\|\mathbf{d}_k\| - \kappa_H^3\|\mathbf{d}_k\|^2\right) \\ &\geq \frac{\alpha}{16}\|F(\mathbf{x}_k)\|\|\mathbf{d}_k\|, \qquad (3.21)\end{aligned}$$

for k sufficiently large, where we used $\|F(\mathbf{x_k})\| \geq \|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\|$ (using (3.20)) for the first inequality. Now, by (3.4), we have

$$\begin{aligned}&\|(\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k) - (F(\mathbf{x}_k) + \nabla F(\mathbf{x}_k)\mathbf{d}_k)\| \\ &\quad \leq \|\nabla\mathbf{m}(\mathbf{x}_k) - \nabla F(\mathbf{x}_k)\|\|\mathbf{d}_k\| \leq L_3\|\mathbf{s}_k\|\|\mathbf{d}_k\|.\end{aligned}$$

By Lemma 2.4, we have that

$$\|F(\mathbf{x}_k + \mathbf{d}_k) - (F(\mathbf{x}_k) + \nabla F(\mathbf{x}_k)\mathbf{d}_k)\| \leq \frac{L_2\sqrt{m}}{2}\|\mathbf{d}_k\|^2. \tag{3.22}$$

Hence, we obtain

$$\begin{aligned}&\|(\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k) + (F(\mathbf{x}_k) + \nabla F(\mathbf{x}_k)\mathbf{d}_k)\| \\ &\quad \leq 2\|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\| + L_3\|\mathbf{s}_k\|\|\mathbf{d}_k\|,\end{aligned}$$

and

$$\begin{aligned}&\|F(\mathbf{x}_k + \mathbf{d}_k) + (F(\mathbf{x}_k) + \nabla F(\mathbf{x}_k)\mathbf{d}_k)\| \\ &\quad \leq 2\|F(\mathbf{x}_k) + \nabla F(\mathbf{x}_k)\mathbf{d}_k\| + \frac{L_2\sqrt{m}}{2}\|\mathbf{d}_k\|^2 \\ &\quad \leq 2\|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\| + 2L_3\|\mathbf{s}_k\|\|\mathbf{d}_k\| + \frac{L_2\sqrt{m}}{2}\|\mathbf{d}_k\|^2.\end{aligned}$$

Therefore using $\|\mathbf{d}_k\| \leq 2\|\mathbf{s}_k\|$,

$$\begin{aligned}&\left|\|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\|^2 - \|F(\mathbf{x}_k + \mathbf{d}_k)\|^2\right| \\ &\leq \left|\|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\|^2 - \|F(\mathbf{x}_k) + \nabla F(\mathbf{x}_k)\mathbf{d}_k\|^2\right| \\ &\quad + \left|\|F(\mathbf{x}_k) + \nabla F(\mathbf{x}_k)\mathbf{d}_k\|^2 - \|F(\mathbf{x}_k + \mathbf{d}_k)\|^2\right| \\ &\leq (2\|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\| + L_3\|\mathbf{s}_k\|\|\mathbf{d}_k\|)(L_3\|\mathbf{s}_k\|\|\mathbf{d}_k\|) \\ &\quad + \left(2\|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\| + 2L_3\|\mathbf{s}_k\|\|\mathbf{d}_k\| + \frac{L_2\sqrt{m}}{2}\|\mathbf{d}_k\|^2\right) \\ &\quad \times \left(\frac{L_2\sqrt{m}}{2}\|\mathbf{d}_k\|^2\right) \\ &= (2L_3\|\mathbf{s}_k\| + L_2\sqrt{m}\|\mathbf{d}_k\|)\|\mathbf{d}_k\|\|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\| \\ &\quad + (L_3^2\|\mathbf{s}_k\|\|\mathbf{d}_k\| + L_2L_3\sqrt{m}\|\mathbf{d}_k\|^2)\|\mathbf{s}_k\|\|\mathbf{d}_k\| + \frac{L_2^2m}{4}\|\mathbf{d}_k\|^4.\end{aligned}$$

By the above inequality and using $\|F(\mathbf{x_k})\| \geq \|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\|$ again, we have

$$\begin{aligned}|Ared_k - Pred_k| = \frac{1}{2}\Big|&\|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\|^2 - \|F(\mathbf{x_k} + \mathbf{d}_k)\|^2 \\ &+ \kappa_H^3\|\mathbf{m}(\mathbf{x}_k)\|\|\mathbf{d}_k\|^2\Big|\end{aligned}$$

$$\leq (2L_3\|\mathbf{s}_k\| + (L_2\sqrt{m} + \kappa_H^3)\|\mathbf{d}_k\|)\|F(\mathbf{x}_k)\|\|\mathbf{d}_k\| + (L_3^2\|\mathbf{s}_k\|\|\mathbf{d}_k\| + L_2L_3\sqrt{m}\|\mathbf{d}_k\|^2)\|\mathbf{s}_k\|\|\mathbf{d}_k\| + \frac{L_2^2 m}{4}\|\mathbf{d}_k\|^4, \tag{3.23}$$

for k sufficiently large. Hence, by (3.21) and (3.23), we have

$$|r_k - 1| \leq \left|\frac{Ared_k - Pred_k}{Pred_k}\right| \leq \frac{32L_3}{\alpha}\|\mathbf{s}_k\| + \frac{16(L_2\sqrt{m} + \kappa_H^3)}{\alpha}\|\mathbf{d}_k\| + \frac{16(L_3^2\|\mathbf{s}_k\|\|\mathbf{d}_k\| + L_2L_3\sqrt{m}\|\mathbf{d}_k\|^2)}{\alpha}\frac{\|\mathbf{s}_k\|}{\|F(\mathbf{x}_k)\|} + \frac{4L_2^2 m}{\alpha}\frac{\|\mathbf{d}_k\|^3}{\|F(\mathbf{x}_k)\|}.$$

The above inequality together with Assumption 3.3, (3.1) and (3.17), implies (3.18) holds when k is sufficiently large. Hence, if we choose $\epsilon = 0.3$ in (3.18), by (3.3) and (3.18), Step 3 and Step 6 in the DFLS algorithm will not be invoked for large k. Therefore, from the DFLS algorithm and $r_k \geq 0.7$ for all large k, (3.19) holds. □

The lemma we just proved states that asymptotically the trust region radius will be bounded away from zero and the model will be a very good one. We can see this implies that the trust region constraint in the DFLS algorithm is eventually inactive. Hence, the algorithm will ultimately reduce to a "regularized Levenberg-Marquardt-type" method. Using a similar approach to that proposed in [8] we are able to show that the iterates generated by our algorithm converge to a local solution quadratically. We start with showing that the iterates converge to a solution superlinearly.

Theorem 3.5 *Under Assumptions 2.1, 2.2, 3.1, 3.2, and 3.3, the sequence* $\{\mathbf{x}_k\}$ *generated by the DFLS algorithm converges to a point* $\mathbf{x}^* \in \mathbf{X}^*$ *superlinearly.*

Proof First, we choose $\bar{k}$ sufficiently large that Lemma 3.1, Lemma 3.3, Lemma 3.4 and the local error bound condition (3.1) hold for all $k \geq \bar{k}$.

Since $\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| \to 0$ as $k \to \infty$, by Assumption 3.3, it follows from (3.19) that there exists an integer, denoted as $\bar{k}$, such that

$$\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| \leq \Delta_k, \quad \text{for all } k \geq \bar{k}.$$

Hence, denoting $\mathbf{s}_k = \bar{\mathbf{x}}_k - \mathbf{x}_k$, for all $k \geq \bar{k}$ large enough, we have

$$\begin{aligned}
\phi_k(\mathbf{d}_k) \leq \phi_k(\mathbf{s}_k) &\leq \|\mathbf{m}(\mathbf{x}_k) + \nabla\mathbf{m}(\mathbf{x}_k)\mathbf{s}_k\|^2 + \frac{\kappa_H^3}{2}\|\mathbf{m}(\mathbf{x}_k)\|\|\mathbf{s}_k\|^2 \\
&\leq (\|F(\mathbf{x}_k) + \nabla F(\mathbf{x}_k)\mathbf{s}_k\| + L_3\|\mathbf{s}_k\|^2)^2 + \frac{\kappa_H^3}{2}\|F(\mathbf{x}_k)\|\|\mathbf{s}_k\|^2 \quad \text{(using (3.4))} \\
&\leq \left(\frac{m}{2}L_2\|\mathbf{s}_k\|^2 + L_3\|\mathbf{s}_k\|^2\right)^2 + \frac{\kappa_H^3}{2}\|F(\mathbf{x}_k)\|\|\mathbf{s}_k\|^2 \\
&\qquad \text{(using the fifth inequality in Lemma 2.4)} \\
&\leq \kappa_H^3 L_1\|\mathbf{s}_k\|^3 \quad \text{(using the second inequality in Lemma 2.4).}
\end{aligned}$$

From the above inequality, Lemma 2.4 and (3.17), for $k \geq \bar{k}$ large,

$$\begin{aligned}\|F(\mathbf{x}_{k+1})\| = \|F(\mathbf{x}_k + \mathbf{d}_k)\| &\leq \|F(\mathbf{x}_k) + \nabla F(\mathbf{x}_k)\mathbf{d}_k\| + \frac{\sqrt{m}}{2} L_2 \|\mathbf{d}_k\|^2 \\ &\leq \sqrt{2\phi_k(\mathbf{d}_k)} + \frac{\sqrt{m}}{2} L_2 \|\mathbf{d}_k\|^2 \\ &\leq 2\sqrt{\kappa_H^3 L_1}\|\mathbf{s}_k\|^{3/2}.\end{aligned}$$

Hence, by the above inequality and (3.1), we have

$$\alpha\|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| \leq \|F(\mathbf{x}_{k+1})\| \leq 2\sqrt{\kappa_H^3 L_1}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|^{3/2} \tag{3.24}$$

for all large k. Consequently,

$$\sum_{k=0}^{\infty} \|\mathbf{x}_k - \bar{\mathbf{x}}_k\| \leq \infty.$$

Since (3.17) holds for all sufficiently large k, it follows from the above inequality that

$$\sum_{k=0}^{\infty} \|\mathbf{d}_k\| \leq \infty.$$

So, $\mathbf{x}_k$ converges to some point $\mathbf{x}^* \in \mathbf{X}^*$. In addition, it follows from (3.24), the definition of $\bar{x}_k$ and

$$\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| \leq \|\mathbf{x}_k - \bar{\mathbf{x}}_{k+1}\| \leq \|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| + \|\mathbf{d}_k\|$$

that $\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| \leq 2\|\mathbf{d}_k\|$ for large k. This together with (3.17) and (3.24) imply

$$\|\mathbf{d}_{k+1}\| \leq \frac{16\sqrt{\kappa_H^3 L_1}}{\alpha}(\|\mathbf{d}_k\|)^{3/2}.$$

This implies $\mathbf{x}_k$ converges to the point $\mathbf{x}^*$ superlinearly. □

In the following we can apply the same singular value decomposition (SVD) technique as is used in [8] to show that $\mathbf{x}_k$ converges to the point $\mathbf{x}^* \in X^*$ quadratically. Suppose that the $rank(\nabla F(\mathbf{x}^*)) = r \geq 0$. Using the singular value decomposition, we can write $\nabla F(\mathbf{x}^*)$ in the following form with the appropriate dimensions:

$$\nabla F(\mathbf{x}^*) = U^* \Sigma^* (V^*)^{\mathsf{T}},$$

where $\Sigma^* = diag(\sigma_1^*, \sigma_2^*, \ldots, \sigma_r^*)$, $(U^*)^{\mathsf{T}}U^* = I_{r\times r}$, $(V^*)^{\mathsf{T}}V^* = I_{r\times r}$ and $\sigma_1^* \geq \sigma_2^* \geq \cdots \geq \sigma_r^* > 0$. Since $\mathbf{x}_k \to \mathbf{x}^*$, it follows from (3.4) and the continuity properties of the SVD that, for k sufficiently large, we can also write $\nabla \mathbf{m}(\mathbf{x}_k)$ as:

$$\nabla \mathbf{m}(\mathbf{x}_k) = U_k \Sigma_k V_k^{\mathsf{T}} \tag{3.25}$$

where $U_k = (U_{k,1}, U_{k,2}, U_{k,3})$, $U_k^{\mathsf{T}} U_k = I_{m\times m}$, $V_k = (V_{k,1}, V_{k,2}, V_{k,3})$, $V_k^{\mathsf{T}} V_k = I_{n\times n}$, $\Sigma_k = diag(\Sigma_{k,1}, \Sigma_{k,2}, \Sigma_{k,3})$, $rank(\Sigma_{k,1}) = r$, $rank(\Sigma_{k,2}) = rank(\Sigma_k) - r$ and $\Sigma_{k,3} = 0$, with

$$\lim_{k\to\infty} \Sigma_{k,1} = \Sigma^* \quad \text{and} \quad \lim_{k\to\infty} \Sigma_{k,2} = 0. \tag{3.26}$$

(Note that we allow the possibility that $\Sigma_{k,2}$ and $\Sigma_{k,3}$ are empty.)

Because of (3.4), Lemma 2.4, $\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| \le \|\mathbf{x}_k - \mathbf{x}^*\|$ and the perturbation properties of the SVD, we have that

$$\begin{aligned}\|diag(\Sigma_{k,1} - \Sigma_1^*, \Sigma_{k,2}, 0\| &\le \|\nabla \mathbf{m}(\mathbf{x}_k) - \nabla F(\mathbf{x}^*)\| \\ &\le \|\nabla \mathbf{m}(\mathbf{x}_k) - \nabla F(\mathbf{x}_k)\| + \|\nabla F(\mathbf{x}_k) - \nabla F(\mathbf{x}^*)\| \\ &\le L_3\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| + L_2\|\mathbf{x}_k - \mathbf{x}^*\| \le (L_3 + L_2)\|\mathbf{x}_k - \mathbf{x}^*\|,\end{aligned}$$

for all large k. Then, the following lemma can be obtained directly by applying the technique used in Lemma 2.3 in [8]. We omit restating its proof here.

Lemma 3.6 *Under Assumptions 2.1, 2.2, 3.1, 3.2, and 3.3, for k sufficiently large, we have*

(a) $\|U_{k,1}U_{k,1}^{\mathsf{T}} F(\mathbf{x}_k)\| \le L_1\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|$
(b) $\|U_{k,2}U_{k,2}^{\mathsf{T}} F(\mathbf{x}_k)\| \le 2(L_2 + L_3)\|\mathbf{x}_k - \mathbf{x}^*\|^2$
(c) $\|U_{k,3}U_{k,3}^{\mathsf{T}} F(\mathbf{x}_k)\| \le (L_2 + L_3)\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|^2$.

Now, we have the local quadratic convergence theorem.

Theorem 3.7 *Under Assumptions 2.1, 2.2, 3.1, 3.2, and 3.3, the sequence $\{\mathbf{x}_k\}$ generated by the DFLS algorithm converges to a point $\mathbf{x}^* \in \mathbf{X}^*$ quadratically.*

Proof By Theorem 3.5, we know the sequence $\{\mathbf{x}_k\}$ converges to a point $\mathbf{x}^* \in \mathbf{X}^*$ superlinearly. Hence,

$$\lim_{k\to\infty} \|\mathbf{d}_k\| = 0.$$

This together with (3.19) implies that

$$\|\mathbf{d}_k\| < \Delta_k,$$

for all sufficiently large k. Hence, the trust region constraint in the DFLS algorithm is ultimately inactive. Therefore, from the SVD of $\nabla \mathbf{m}(\mathbf{x}_k)$, we have that

$$\begin{aligned}\mathbf{d}_k &= -V_{k,1}(\Sigma_{k,1}^2 + \kappa_H^3\|\mathbf{m}(\mathbf{x}_k)\|I)^{-1}\Sigma_{k,1}U_{k,1}^{\mathsf{T}} F(\mathbf{x}_k) \\ &\quad - V_{k,2}(\Sigma_{k,2}^2 + \kappa_H^3\|\mathbf{m}(\mathbf{x}_k)\|I)^{-1}\Sigma_{k,2}U_{k,2}^{\mathsf{T}} F(\mathbf{x}_k) \\ &= -V_{k,1}(\Sigma_{k,1}^2 + \kappa_H^3\|F(\mathbf{x}_k)\|I)^{-1}\Sigma_{k,1}U_{k,1}^{\mathsf{T}} F(\mathbf{x}_k) \\ &\quad - V_{k,2}(\Sigma_{k,2}^2 + \kappa_H^3\|F(\mathbf{x}_k)\|I)^{-1}\Sigma_{k,2}U_{k,2}^{\mathsf{T}} F(\mathbf{x}_k),\end{aligned} \tag{3.27}$$

for sufficiently large k. Then, the result that the sequence $\{\mathbf{x}_k\}$ converges to $\mathbf{x}^*$ quadratically follows directly from the above equality, Lemma 3.6 and the same approach used for Theorem 2.2 in [8]. But for completeness and convenience of the reader, we still provide the proof as follows.

First, by (3.25) and (3.27), we have

$$\begin{aligned} F(\mathbf{x}_k) + \nabla \mathbf{m}(\mathbf{x}_k)\mathbf{d}_k &= F(\mathbf{x}_k) - U_{k,1}\Sigma_{k,1}(\Sigma_{k,1}^2 + \kappa_H^3\|F(\mathbf{x}_k)\|I)^{-1}\Sigma_{k,1}U_{k,1}^{\mathsf{T}}F(\mathbf{x}_k) \\ &\quad - U_{k,2}\Sigma_{k,2}(\Sigma_{k,2}^2 + \kappa_H^3\|F(\mathbf{x}_k)\|I)^{-1}\Sigma_{k,2}U_{k,2}^{\mathsf{T}}F(\mathbf{x}_k) \\ &= \kappa_H^3\|F(\mathbf{x}_k)\|U_{k,1}(\Sigma_{k,1}^2 + \kappa_H^3\|F(\mathbf{x}_k)\|I)^{-1}U_{k,1}^{\mathsf{T}}F(\mathbf{x}_k) \\ &\quad + \kappa_H^3\|F(\mathbf{x}_k)\|U_{k,2}(\Sigma_{k,2}^2 + \kappa_H^3\|F(\mathbf{x}_k)\|I)^{-1}U_{k,2}^{\mathsf{T}}F(\mathbf{x}_k) \\ &\quad + U_{k,3}U_{k,3}^{\mathsf{T}}F(\mathbf{x}_k). \end{aligned} \tag{3.28}$$

By Theorem 3.5, $F(\mathbf{x}^*) = \mathbf{0}$ and (3.26) implies that, for k sufficiently large, we have

$$\|(\Sigma_{k,1}^2 + \kappa_H^3\|F(\mathbf{x}_k)\|I)^{-1}\| \le \|\Sigma_{k,1}^{-2}\| \le \frac{2}{(\sigma_r^*)^2}$$

and

$$\|(\Sigma_{k,2}^2 + \kappa_H^3\|F(\mathbf{x}_k)\|I)^{-1}\| \le \frac{1}{\|F(\mathbf{x}_k)\|\,\kappa_H^3}.$$

Hence, it follows from the above two inequalities, the proof of (3.4), (3.28), Lemma 2.4 and Lemma 3.6 that

$$\begin{aligned} &\|F(\mathbf{x}_k) + \nabla \mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\| \\ &\quad \le \kappa_H^3\|F(\mathbf{x}_k)\|\left(\frac{2L_1}{(\sigma_r^*)^2}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| + \frac{2(L_2+L_3)}{\kappa_H^3\|F(\mathbf{x}_k)\|}\|\mathbf{x}_k - \mathbf{x}^*\|^2\right) \\ &\quad + (L_2+L_3)\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|^2 \\ &\quad \le \left(\frac{2L_1^2\kappa_H^3}{(\sigma_r^*)^2} + 3(L_2+L_3)\right)\|\mathbf{x}_k - \mathbf{x}^*\|^2 = \mathcal{O}(\|\mathbf{x}_k - \mathbf{x}^*\|^2). \end{aligned} \tag{3.29}$$

From Lemma 3.5, we know $\mathbf{x}_k$ converges to $\mathbf{x}^*$ superlinearly, which implies

$$\lim_{k\to\infty}\frac{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} = \lim_{k\to\infty}\frac{\|\mathbf{d}_k\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} = 1. \tag{3.30}$$

Then, it follows, using the above equality, Assumption 3.1, Lemma 3.3, Lemma 3.1, (3.22) and (3.29) that

$$\begin{aligned} \|\mathbf{d}_{k+1}\| &\le 2\,dist(\mathbf{x}_{k+1}, \mathbf{X}^*) \le \frac{2}{\alpha}F(\mathbf{x}_{k+1}) = \frac{2}{\alpha}F(\mathbf{x}_k + \mathbf{d}_k) \\ &\le \|F(\mathbf{x}_k) + \nabla F(\mathbf{x}_k)\mathbf{d}_k\| + \frac{L_2\sqrt{m}}{2}\|\mathbf{d}_k\|^2 \end{aligned}$$

$$\leq \|F(\mathbf{x}_k) + \nabla \mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\| + L_3\|\mathbf{x}_k - \bar{x}_k\|\|\mathbf{d}_k\| + \frac{L_2\sqrt{m}}{2}\|\mathbf{d}_k\|^2$$

$$\leq \|F(\mathbf{x}_k) + \nabla \mathbf{m}(\mathbf{x}_k)\mathbf{d}_k\| + L_3\|\mathbf{x}_k - \mathbf{x}^*\|\|\mathbf{d}_k\| + \frac{L_2\sqrt{m}}{2}\|\mathbf{d}_k\|^2$$

$$= \mathcal{O}(\|\mathbf{d}_k\|^2),$$

which by (3.30) implies $\mathbf{x}_k$ converges to $\mathbf{x}^*$ quadratically. □

4 Numerical experiments

In this section, we give numerical comparisons of the long-term performances of DFLS for achieving high accuracy solutions relative to the performances of the following codes:

- LMDIF [12]: A version, developed in 1980 by Garbow, Hillstrom and Moré of the Levenberg-Marquardt algorithm that uses finite (forward) differences for approximating the gradients of the sum of the squares objective function.[3]
- NEWUOA [14]: Software for unconstrained optimization without derivatives developed by Powell in 2004.

LMDIF and NEWUOA belong to two fundamentally different classes of algorithms that do not use explicit derivatives. One class of methods use finite difference to approximate the derivative whereas the other uses polynomial interpolations to build approximate models of the problem. NEWUOA is derivative-free software that was initially developed for general unconstrained optimization. Hence, NEWUOA does not make use of the least-squares problem structure. An extension to accommodate simple bounds, BOBYQA, has been development by Powell [13]. In [20] we demonstrated the generally best performance of DFLS compared with LMDIF and NEWUOA within a limited computational budget. Here, our purpose of using LMDIF and NEWUOA again as comparison codes is to indicate the superior long-term asymptotic performances of DFLS by taking advantage of the problem's structure as well as by taking the model based approach developed in [14].

Our implementation of the DFLS algorithm is based on the same framework as that provided by NEWUOA. This is partly because the details in NEWUOA are carefully and intelligently chosen and partly because it makes the comparison between it and DFLS more meaningful. In our runs of DFLS, we chose $N_P = 2n + 1$ sampling points and applied the same polynomial interpolation techniques as those implemented in NEWUOA, on each of the nonlinear functions f_i, $i = 1, \ldots, m$. Since the sampling points for all the functions f_i, $i = 1, \ldots, m$, are the same, the amount of work for calculating the updates of the coefficients of all the models can

[3] The Levenberg-Marquardt algorithm is essentially a modified Gauss-Newton algorithm where the modification can be thought of as a regularization of $J^\top J$ with a parameter scaled identity matrix. Although the motivation is rather different, if one associates the parameter with a trust region radius it is essentially a trust-region Gauss-Newton method.

be kept within $\mathcal{O}((m(N_P + n)) \approx \mathcal{O}(mn)$ [16] and therefore, discounting the occasional origin shifts, the work of building the models per iteration can be kept within $\mathcal{O}(mn^2)$ (see [15] for details). However, DFLS requires more memory to store individual models $m_i(\mathbf{x})$, for $i = 1, \ldots, m$, rather than one single model. This leads to $(m - 1)(1 + n + n(n + 1)/2) \approx \mathcal{O}(mn^2)$ more storage compared with NEWUOA, which is the cost of exploiting the structure. NEWUOA includes excellent software for selecting the sampling points and for updating the models accurately and efficiently. Many elegant and useful techniques for controlling the numerical rounding errors and stability issues are addressed in [14, 15] and the references therein, and the reader is encouraged to obtain more details there.

All three codes were written in Fortran and compiled with F77 on an IBM ThinkPad T40 laptop computer. In LMDIF, it is admissible for the variables to be scaled internally. All the other parameters are set to the default values except for the difference step values for LMDIF in the case of noisy functions. In NEWUOA, we set the number of interpolating points to the default number recommended by the code, namely $2n + 1$. In both NEWUOA and DFLS, the initial trust region radius is set to 1.0. All algorithms stop when the required stopping condition are satisfied or the algorithms stop internally for numerical reasons. All the solvers are tested using four different sets of test problems, including both zero and nonzero residual problems, both problems with and without noise. Since in this paper we are interested in investigating the long-term asymptotic behavior of different solvers, we assume there are effectively no constraints on the computational budget and allow the solvers to explore the test problems as much as they reasonably can. Hence, in all three codes, we set the maximum number of function evaluations to be 10^5, which is rather large for derivative-free optimization. Our numerical results are listed from Tables 1 to 4. In these tables, "n" means the number of variables, i.e. the dimension of the primal variable, "m" means the number of nonlinear functions in (1.1). "NF" stands for the number of function evaluations performed by each code. We put "$*$" in the table, if the code converged to a different local optimum from the global optimum.

Our first set of test problems consist of 25 zero residual problems. The first 15 problems are the problems with equality constraints in the Hock and Schittkowski [11] test problem library. They are reformulated as least-squares problems by taking the objective function minus its optimum value as the first nonlinear function and taking all the equality constraints as the other nonlinear functions in (1.1). The remaining problems were collected from a variety of places during the development of the DFLS code. The stopping condition for all the codes was

$$\Phi(\mathbf{x}_k) \leq \max\{10^{-12}, 10^{-20}\Phi(\mathbf{x}_0)\}, \tag{4.1}$$

where $\mathbf{x}_0$ is the initial point, which is rather stringent. The numerical results are listed in Table 1. All the codes stop either because the stopping condition (4.1) was satisfied or the internal conditions in the code determined that no further improvement could be made, perhaps due to the numerical rounding errors, or the number of function evaluations reached its maximum allowable value 10^5. For these tests, the finite difference parameter in LMDIF was set to be 10^{-7}. If the optimal stopping condition was not met, we list in the column F_Error the best cost function value finally returned by the code, along with the number of function evaluations. From Table 1,

Table 1 Numerical comparisons (zero residuals)

Problem name	n	m	LMDIF NF/F_Error	NEWUOA NF/F_Error	DFLS NF/F_Error
HS26	3	2	3631/1.25e−11	3706	378
HS27	3	2	313	405	338
HS28	3	2	104	247	48
HS39	4	3	81	585	53
HS40	4	4	36	344	30
HS42	4	3	93	655	48
HS46	5	3	502/8.76e−9	7060	1076
HS47	5	4	5328	2151	263
HS49	5	3	10^5/4.06e−7	2546	2395
HS56	7	5	228	1851	168
HS60	3	2	57	415	65
HS77	5	3	86/3.06e−3	1439	145
HS78	5	4	79	727	48
HS81	5	4	202	2262	107
HS111	10	4	63477/6.35e−7	10^5/2.32e−8	2193/7.85e−12
CONN	20	20	880	639	113
CHROSEN	20	38	148	635	96
CHROSEN	80	158	568	3094	346
TRIGSSQS	10	10	56	490	81
TRIGSSQS	10	20	45	236	49
TRIGSSQS	20	20	149	2267	145
BUCKLEY21	6	13	204	2076	171
HEART	6	6	*	10^5/1.21e−2	863
MORE25	20	22	232	10359	198
MORE20	31	31	296	10^5/1.23e−8	431/8.00e−12

we can see that for the zero residual case DFLS is very stable and normally uses between $1/5$ to $1/10$ of the number of function evaluations of NEWUOA. Generally the LMDIF code performs reasonably well too, although not as good as DFLS on average. In some cases, LMDIF could not reach the stopping condition (4.1). Based on these small number of numerical experiments, we believe the approach taken by DFLS is both promising and robust as concerns its long-term behavior for zero residual least-squares problems, although a more challenging set of test problems should be tested to confirm the promise.

Our second test set consists of 22 nonzero residual problems. The "HS" problems in the second test set are reformulated as least-squares problems by simply taking the objective function as the first nonlinear function and taking the equality constraints for the remaining nonlinear functions in (1.1). To ensure that the least-squares residual is nonzero, if the original objective optimum value is zero, the first nonlinear function is taken as the objective function plus ten. The "TRIGSSQS" problems are extracted from [14] and the remaining problems are intrinsically nonzero residual problems. Again, we ran all the codes until either the internal conditions in the code

Table 2 Numerical comparisons (nonzero residuals)

Problem name	n	m	LMDIF NF/F_ReErr	NEWUOA NF/F_ReErr	DFLS NF/F_ReErr
HS26	3	2	106/1.12e−8	189	155
HS27	3	2	322/3.43e−6	248	108
HS28	3	2	1626	107	143
HS39	4	3	114/2.33e−8	245	100
HS40	4	4	64/7.18e−9	125	76
HS42	4	3	166/4.20e−8	84	59
HS46	5	3	15990/5.11e−7	600/1.99e−12	586
HS47	5	4	*	92	92
HS49	5	3	10^5/1.10e−4	832/3.51e−12	516
HS56	7	5	165/4.11e−7	234	160
HS60	3	2	132/3.58e−7	217	81
HS77	5	3	2551/2.31e−5	655	162
HS78	5	4	93/6.69e−9	158	105
HS81	5	4	115/1.68e−4	949	214
HS111	10	4	3807/1.13e−8	10562	673
DENNIS	4	20	1759/5.82e−7	254	130
SPHRPTS	20	45	*	2263	1406
TRIGSSQS	10	10	309/7.10e−9	241	227
TRIGSSQS	10	20	286/6.73e−8	217	241
TRIGSSQS	20	20	426/7.41e−7	580	576
PENALTY1	20	21	183	9855	433
PENALTY2	20	40	482/1.87e−11	1982	336

determined that no further improvement could be made, or the number of function evaluations reached its maximum allowable value of 10^5. The finite difference parameter in LMDIF was again set to 10^{-7}. For each method we calculate

$$\text{F\_ReErr} = \frac{\Phi_{end} - \Phi^*}{\Phi^*},$$

where Φ_{end} is the final cost function value returned by the particular code and Φ^* is the lowest one among all the final (nonzero) cost function values. We list F_ReErr and the number of function evaluations whenever F_ReErr $\geq 10^{-12}$. The numerical results are shown in Table 2. We can again see that for the nonzero residual case, DFLS still performs significantly better than NEWUOA when comparing their long-term behavior. The difference is not as large as in the zero residual case, which is what one might expect since the structure to be exploited is less apparent. On the other hand, with the nonzero residual results, as expected, we can see that LMDIF was unable to reach very high accuracy for most of the test problems.

In many real problems, the situation is exacerbated by the presence of noise. For this reason, we wanted to test how noise affects the long-term performance of each method. We first tested a perturbed version of the zero residual problem set by adding

Table 3 Numerical comparisons (zero residuals, random noise level 1.e−2)

Problem name	n	m	LMDIF NF/F_Error	NEWUOA NF/F_Error	DFLS NF/F_Error
HS26	3	2	69.1/8.53e−5	87.4/1.80e−5	39.7/9.19e−5
HS27	3	2	F	94.3/2.49e−5	43.0/2.71e−5
HS28	3	2	66.0/1.43e−4	85.1/1.29e−5	38.8/6.09e−6
HS39	4	3	72.5/1.47e−4	99.4/1.28e−6	40.7/7.19e−6
HS40	4	4	54.2/6.11e−6	95.4/2.32e−5	30.5/3.14e−5
HS42	4	3	57.0/6.47e−6	93.8/3.55e−3	31.7/3.95e−5
HS46	5	3	89.7/8.03e−5	133.8/5.65e−5	55.6/1.09e−4
HS47	5	4	94.9/2.05e−3	136.3/6.69e−5	57.6/6.53e−5
HS49	5	3	101.5/1.47e−3	192.0/6.05e−4	112.0/1.47e−4
HS56	7	5	93.2/3.82e−4	168.0/2.64e−4	63.0/5.04e−5
HS60	3	2	66.8/2.64e−5	85.7/1.91e−5	37.0/2.09e−6
HS77	5	3	78.0/1.21e−2	149.4/1.95e−3	70.4/1.45e−3
HS78	5	4	75.5/1.36e−5	116.7/2.14e−5	43.8/6.78e−5
HS81	5	4	87.4/3.52e−4	115.4/1.34e−4	37.9/7.44e−5
HS111	10	4	201.5/4.11e−2	235.3/1.32e−4	97.7/2.10e−4
CONN	20	20	F	415.5/7.96e−5	150.5/1.20e−4
CHROSEN	20	38	F	385.4/1.14e−4	144.7/2.08e−4
CHROSEN	80	158	F	2227.3/4.22e−1	599.6/4.26e−4
TRIGSSQS	10	10	139.7/1.47e−4	205.0/1.02e−4	62.8/1.57e−4
TRIGSSQS	10	20	96.9/5.82e−5	158.0/3.65e−5	39.5/9.39e−5
TRIGSSQS	20	20	315.0/2.44e−4	375.7/1.61e−4	112.1/5.27e−4
BUCKLEY21	6	13	138.4/2.10e−4	168.7/3.94e−5	82.8/1.16e−4
HEART	6	6	198.4/3.71e−1	196.6/4.20e−1	129.3/2.00e−1
MORE25	20	22	F	450.2/2.28e−4	201.9/3.90e−4
MORE20	31	31	378.3/4.32e−3	588.1/1.78e−4	200.8/3.87e−5

some random noise to the nonlinear functions in (1.1). More specifically, we let

$$F(\mathbf{x}) = F_{true}(\mathbf{x}) + 10^{-2}E,$$

where $E \in \mathbb{R}^m$ is a random vector with a normal distribution in $[-0.5, 0.5]$. For each problem, we ran the code until the internal conditions terminated the iterations. The ending trust region radius of NEWUOA and DFLS, and the finite difference parameter of LIMDIF are all set to 10^{-2} to be comparable to the noise level. The numerical results are shown in Table 3, where "NF" means the average number of function evaluations during ten runs. Here, we let F_Error be the best true cost function value, i.e. the lowest cost function value without noise, returned by the code during the 10 runs. Since the original problems without noise are zero residual problems, we list F_Error in its column if F_Error ≤ 1.0; otherwise, we list "F" in its column to indicate that the code failed on this problem. We can see DFLS again performs best among these three codes. We remark that both NEWUOA and DFLS are relatively insensitive to

Table 4 Numerical comparisons (nonzero residuals, relative random noise level 1.e−2)

Problem name	n	m	LMDIF NF/F_Error	NEWUOA NF/F_Error	DFLS NF/F_Error
HS26	3	2	51.7/7.38e−3	26.8/1.12e−2	28.5/9.93e−4
HS27	3	2	F	F	68.6/4.95e−3
HS28	3	2	40.8/1.05e−2	24.7/4.86e−2	29.3/1.98e−3
HS39	4	3	91.1/4.69e−3	62.3/3.17e−2	49.5/1.85e−3
HS40	4	4	53.4/3.73e−4	57.9/3.20e−3	47.7/1.12e−3
HS42	4	3	63.7/1.58e−2	23.7/1.61e−3	28.7/2.14e−3
HS46	5	3	62.6/5.73e−3	35.0/2.96e−2	36.3/1.27e−2
HS47	5	4	*	22.0/1.67e−1	22.0/3.92e−2
HS49	5	3	75.3/2.91e−2	101.6/3.70e−1	65.7/5.12e−1
HS56	7	5	86.6/7.55e−3	76.8/1.91e−2	69.4/3.57e−3
HS60	3	2	79.9/2.92e−4	71.8/2.51e−2	63.6/8.43e−4
HS77	5	3	96.1/1.17e−2	F	99.2/3.11e−2
HS78	5	4	56.3/1.07e−3	69.1/9.17e−3	39.7/8.64e−3
HS81	5	4	97.5/3.91e−1	F	75.9/6.34e−3
HS111	10	4	182.2/1.99e−1	95.1/2.63e−1	104.6/4.01e−3
DENNIS	4	20	F	56.4/3.17e−3	40.1/1.03e−3
SPHRPTS	20	45	92.9/7.14e−1	290.7/8.18e−3	249.4/8.66e−3
TRIGSSQS	10	10	108.2/2.31e−2	76.7/5.38e−3	88.0/2.97e−3
TRIGSSQS	10	20	93.8/7.69e−2	90.8/3.05e−3	94.4/2.19e−3
TRIGSSQS	20	20	171.8/1.23e−1	175.6/7.28e−3	181.6/5.68e−3
PENALTY1	20	21	F	446.9/1.51e−1	297.1/8.60e−3
PENALTY2	20	40	248.2/7.71e−3	110.7/4.87e−3	115.1/2.72e−3

the noise in their long-term behavior. LMDIF is more affected by the noise because of using difference approximations for the derivatives.

Secondly, we tested the effect of noise for each method in the case of the nonzero residual problems. In this instance we let

$$F(\mathbf{x}) = F_{true}(\mathbf{x}) + 10^{-2} E \circ |F(\mathbf{x})|.$$

$E \in \mathbb{R}^m$ is again a random vector with normal distribution in $[-0.5, 0.5]$ and $|F(\mathbf{x})|$ is the vector in $\mathbb{R}^m$ with its components the absolute value of the corresponding components of $F(\mathbf{x})$. Here "$\circ$" means componentwise product. For each problem, we again ran each code 10 times until the internal conditions terminated the iterations. The final trust region radius of NEWUOA and DFLS, and the finite difference parameter of LIMDIF are all set to $10^{-2}\sqrt{\Phi^*}$, where Φ^* is the optimal function value of the problem without noise. Then, for each method we calculated

$$\text{F\_ReErr} = \frac{\Phi_{best} - \Phi^*}{\Phi^*},$$

where Φ_{best} is the best true (i.e. the lowest without the added noise) cost function value returned by the codes. The numerical results are shown in Table 4. Here, "NF"

again means the average number of function evaluations during the ten runs. We list F_ReErr, along with the number of function evaluations if F_ReErr ≤ 1.0; otherwise, we list "F" in its column to indicate that the code failed on this problem. From Table 4, DFLS again generally uses the smallest number of function evaluations and returns the best cost function values comparable to the noise level. NEWUOA and LMDIF can give reasonable solutions for some of the problems. Compared with DFLS, LMDIF is again more sensitive to the noise. In addition, we observe that because of not exploiting the problem structure, NEWUOA also starts to be affected by the noise for these nonzero residual problems.

5 Conclusion

In this paper, we have studied the long-term asymptotic convergence behavior of the DFLS algorithm, which is a derivative-free algorithm first presented in [20] for minimizing least-squares problem. More specifically, we have established the local quadratic convergence of the DFLS algorithm for zero residual problems under a certain type of non-singularity assumption, which is considerably weaker than a non-singularity assumption on the Jacobian. Preliminary numerical experiments have been carried out to compare the long-term behavior of DFLS with the other two benchmark algorithms, NEWUOA and LMDIF, on four set of test problems including both zero and nonzero residual problems and both problems with and without noises. We have observed that in all cases DFLS performs significantly better in both efficiency and reliability than the other two comparison codes. In general the polynomial interpolation based methods, NEWUOA and DFLS, are more reliable (for both problems with and without noise) when high accurate solutions are required. Consequently, for long-term, robust and accurate solutions, compared with finite difference based methods, polynomial interpolation based methods are recommended. This is somewhat different from what we have observed in [20] in the numerical performances of the same software within a limited computational budget. There, LMDIF was not as much affected by numerical instabilities and noise. This may be because LMDIF has made full use of the least-squares problem structure, while the polynomial interpolation based method NEWUOA may not build accurate models given a small computing budget. DFLS is not only based on models built by polynomial interpolations, but also takes full advantages of the least-square problem structure. Hence, it is reasonable to believe DFLS will perform more efficiently and will be more robust in both the short-term and the long-term compared with methods that only implement one of the two features in the above discussion. Our numerical results in [20] and [21] support this conclusion. Finally, very recently the bound constrained version of DFLS, called DFBOLS, has been developed in [21]. We will continue to study the global and local convergence behavior of DFBOLS, which again is able to exploit the special structure of the least-squares problem with, in this case, additional bound constraints.

Acknowledgements The authors gratefully acknowledge the comments and suggestions provided by Dr. Katya Scheinberg during this research.

References

1. Conn, A.R., Gould, N.I.M., Toint, Ph.L.: Trust-Region Methods. MPS-SIAM Series on Optimization. SIAM, Philadelphia (2000)
2. Conn, A.R., Scheinberg, K., Vicente, L.N.: Global convergence of general derivative-free trust-region algorithms to first and second order critical points. SIAM J. Optim. **20**, 387–415 (2009)
3. Conn, A.R., Scheinberg, K., Vicente, L.N.: Geometry of interpolation sets in derivative free optimization. Math. Program. **111**, 141–172 (2008)
4. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. MPS-SIAM Series on Optimization. SIAM, Philadelphia (2009)
5. Conn, A.R., Scheinberg, K., Vicente, L.N.: Geometry of sample sets in derivative free optimization: Polynomial regression and underdetermined interpolation. IMA J. Numer. Anal. **28**, 721–748 (2008)
6. Dan, H., Yamashita, N., Fukushima, M.: Convergence properties of the inexact Levenberg-Marquardt method under local error bound. Optim. Methods Softw. **17**, 605–626 (2002)
7. Fan, J.: Convergence properties of a self-adaptive Levenberg-Marquardt algorithm under local error bound condition. Comput. Optim. Appl. **34**, 47–62 (2006)
8. Fan, J., Yuan, Y.: On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption. Computing **74**, 23–39 (2005)
9. Hager, W.W., Zhang, H.: Self-adaptive inexact proximal point methods. Comput. Optim. Appl. **39**, 161–181 (2008)
10. Hager, W.W., Zhang, H.: Asymptotic convergence analysis of a new class of proximal point methods. SIAM J. Control Optim. **46**, 1683–1704 (2007)
11. Hock, W., Schittkowski, K.: Test examples for nonlinear programming codes. Lect. Notes Econ. Math. Syst. **187** (1981)
12. Moré, J.J.: The Levenberg-Marquardt algorithm, implementation and theory. In: Watson, G.A. (ed.) Numerical Analysis. Lecture Notes in Mathematics, vol. 630. Springer, Berlin (1977)
13. Powell, M.J.D.: Developments of NEWUOA for unconstrained minimization without derivatives. IMA J. Numer. Anal. **28**, 649–664 (2008)
14. Powell, M.J.D.: The NEWUOA software for unconstrained optimization without derivatives. DAMTP (2004)
15. Powell, M.J.D.: Least Frobenius norm updating of quadratic models that satisfy interpolation conditions. Math. Program., Ser. B **100**, 183–215 (2004)
16. Powell, M.J.D.: On trust region methods for unconstrained minimization without derivatives. Math. Program. **97**, 605–623 (2003)
17. Tseng, P.: Error bounds and superlinear convergence analysis of some Newton-type methods in optimization. In: Di Pillo, G., Giannessi, F. (eds.) Nonlinear Optimization and Related Topics, pp. 445–462. Kluwer Academic, Dordrecht (2000)
18. Yamashita, N., Fukushima, M.: The proximal point algorithm with genuine superlinear convergence for the monotone complementarity problem. SIAM J. Optim. **11**, 364–379 (2000)
19. Yamashita, N., Fukushima, M.: On the rate of convergence of the Levenberg-Marquardt method. Computing **15**, 237–249 (2001)
20. Zhang, H., Conn, A.R., Scheinberg, K.: A derivative-free algorithm for least-squares minimization. SIAM J. Optim. (2010, accepted)
21. Zhang, H., Conn, A.R., Scheinberg, K.: A derivative-free algorithm for least-squares minimization with bound constraints (2010, in preparation)