Journal of Electroanalytical Chemistry 851 (2019) 113439

Contents lists available at ScienceDirect

Journal of Electroanalytical Chemistry

journal homepage: www.elsevier.com/locate/jelechem

Research paper

Optimizing noisy CNLS problems by using Nelder-Mead algorithm: A new method to compute simplex step efficiency

Mark Žic [a,*], Sergei Pereverzyev [b]

[a] Ruđer Bošković Institute, P.O. Box 180, 10000 Zagreb, Croatia
[b] Johann Radon Institute for Computational and Applied Mathematics, Altenbergerstrasse 69, A-4040 Linz, Austria

ARTICLE INFO

Article history:
Received 12 April 2019
Received in revised form
29 August 2019
Accepted 30 August 2019
Available online 31 August 2019

Keywords:
EIS
CNLS
Nelder-Mead
Adaptive and standard parameters

ABSTRACT

Nelder-Mead "simplex" algorithm (NMA) is a derivative-free algorithm that can be used to solve complex nonlinear least-squared (CNLS) problems. NMA can fit equivalent electrical circuit models to noisy electrochemical impedance spectroscopy (EIS) data. The ability of NMA simplex to adapt itself onto the local landscape of mathematical functions is governed by the simplex steps (*reflection*, *expansion*, *inside* and *outside* contraction and *shrink* steps). However, according to EIS literature, a method to compute the simplex step efficiency has not been reported yet.

Herein, we provide and recommend a new method to compute the simplex step efficiency. The new method was used to evaluate the *adaptive* and *standard* NMA modifications (SNMA and ANMA). Be advised that by using the new method we detected an unknown property of a more successful ANMA (vs. SNMA), i.e. the application of adaptive parameters decreased only the inside contraction step efficiency. However, the aforementioned drawback was resolved by modifying and boosting the existing ANMA, which was a process that demonstrated the usefulness of the new method. Thereby, SNMA, ANMA, and the modified ANMA fitting engines were embedded in the free (MIT licensed) software by using Python programming language.

1. Introduction

The Nelder-Mead algorithm (NMA) [1] is one of the most used derivative-free algorithms [2,3], which is inspired by the "simplex" method proposed by Spendley et al. [4]. The NMA simplex adapts itself to the local landscape of mathematical functions [1,5,6] in order to move in the local downhill direction. This adaptation process is enabled by using *reflection*, *expansion*, *inside contraction*, *outside contraction* and *shrink* operations (i.e. steps). However, as these operations do not require the objective function derivatives [2,7], NMA based algorithms are not sensitive to noise.

Furthermore, NMA-based fitting engines (e.g. [8–10]) are especially suitable to fit[1] equivalent electrical circuit (EEC) models to noisy electrochemical impedance spectroscopy (EIS) data [11]. For that reason, Dellis et al. [8] developed the first NMA-based software for EIS data fitting, which was tested by solving noisy complex nonlinear least square (CNLS) problems [11]. The authors applied the original NMA [1] with the *standard* choice of parameters (SNMA). However, SNMA efficiency is low when the number of parameters is higher >2 [1,5,12,13], which makes it unsuitable for impedance data fitting.

Due to a low SNMA efficiency problem at higher dimensions, several authors focused their efforts on developing enhanced NMA modifications [12,14–18]. Interestingly, one of the NMA modifications [15] evolves a NMA-like algorithm using genetic programming, which is an action that boosts algorithm's efficiency in the higher dimensions. On the other hand, Musafer et al. [17] developed a modification which does not depend on a number of parameters. Furthermore, a recent publication by Fajfar et al. [18] showed that SNMA efficiency can be improved by decreasing the angle between the search direction and direction of the local downhill gradient at the worst vertex. However, up-to-date EIS literature does not propose a method to compute simplex step efficiency which would facilitate a development and evaluation process of diverse new NMA modifications.

In this paper, a new method (which uses the simplex distortion data) to compute simplex step efficiency is proposed. In order to test the new method, we used SNMA and one superior NMA modification with a known impact on simplex distortions. From a vast number of the available enhanced NMA modifications [12,14–18], we selected the adaptive NMA (ANMA) modification [12] in which simplex distortions are alleviated. In Section 2, SNMA and ANMA are briefly reviewed, whereas a procedure to prepare

* Corresponding author.
E-mail addresses: mzic@irb.hr (M. Žic), sergei.pereverzyev@oeaw.ac.at (S. Pereverzyev).

[1] See, e.g.: https://www.mathworks.com/discovery/data-fitting.html.

https://doi.org/10.1016/j.jelechem.2019.113439
1572-6657/

differently polluted CNLS problems is presented in Section 3. Furthermore, in Section 4.1, a design of SNMA and ANMA fitting engines is given and then in Sections 4.2–4.3 we solve differently polluted CNLS problems to determine which algorithm is more superior. Next, an approach to monitor simplex distortions is described in Section 4.4, whilst in Section 4.5 we examine whether this approach yields data with a physical meaning. In Section 4.6 a new method which calculates the simplex step efficiency is offered. And finally, we put an effort to boost the existing ANMA modification by using the simplex step efficiency data (Section 4.7), a task that will affirm the applicability of the proposed method.

The aim of the present work is i) to design a new contemporary NMA-based fitting engine, ii) to propose a new method to compute simplex step efficiency, and iii) to boost the existing ANMA by using the data obtained from the proposed method.

2. Theoretical background

2.1. Nelder-Mead algorithm

The Nelder-Mead algorithm (NMA) is an iterative algorithm [12] commonly used to minimize real-valued functions:

$$O(\mathbf{x}),$$

where $O:\mathbb{R}^n \rightarrow \mathbb{R}$ is called the objective function of n designed parameters ($\mathbf{x}=[x_1,x_2,\ldots,x_n]^T$) and n is usually referred to as the dimension. At any stage of the iteration process the algorithm keeps track of $n+1$ points of interest (i.e. $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_{n+1}$), which form vertices of a simplex [3]. The simplex in n dimensions is a geometric shape that comprises $n+1$ vertices [19], i.e. the simplex in two dimensions is a triangle. As explained in the original Nelder and Mead paper [1], in each stage of the iteration process the *worst* vertex which yields the *highest* $O(\mathbf{x})$ value is replaced by a new vertex.

2.2. Problems solvable by Nelder-Mead algorithm

Generally, the algorithm is used to solve unconstrained problems of the form:

$$minO(\mathbf{x})$$

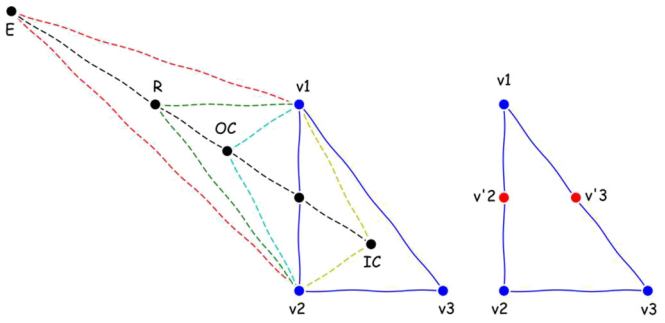
In order to extract, e.g. parameters x_1, x_2 and x_3 of the $EEC_{R(CR)}$ model from impedance data, the following function given by Sheppard et al. [20] was utilized in this work:

$$O(\omega_i,x)=\sum_{i=1}^{N}\left(w_i(Re(Y_i)-Re(y_i))^2+w_i(Im(Y_i)-Im(y_i))^2\right),y_i = \mathrm{EEC}_{\mathrm{R(CR)}}(\omega_i,x), \quad (1)$$

$$w_i=\frac{1}{Re(Y_i)^2+Im\,(Y_i)^2}, \quad (2)$$

$$\mathrm{EEC}_{\mathrm{R(CR)}}(\omega_i,x)=x_1+\frac{1}{i\omega_i x_2+\frac{1}{x_3}}=R_1+\frac{1}{i\omega_i C_2+\frac{1}{R_3}}, \quad (3)$$

where N, Y_i, y_i, w_i and ω_i are the number of data points, the i^{th} value of experimental impedance data, the i^{th} value of computed, e.g. $EEC_{R(CR)}$ impedance data, the "weighting" modulus factor associated with the i^{th} data point and the angular frequency associated with the i^{th} data point. For more comprehensive study on meaning of R, C and Q symbols, the readers can refer to Table 1 of [21]. Interestingly, when studying properties of a new Nelder-Mead modifications, it is common to use different objective functions, as demonstrated by, e.g. Fajfar et al. [15,18]. However, when investigating new EIS fitting engines properties (see, e.g. [21,22]), one usually applies Eq. (1) and tries several different EECs. An interesting analysis which deals with fitting process in EIS can be found here [23].



Scheme 1. Schematic representation[a] of the Nelder-Mead algorithm steps when the number of parameters (n) is 2. Symbol reference: $\mathbf{v}_1$-best vertex, $\mathbf{v}_2$-next-to-worst, $\mathbf{v}_3$-worst vertex, E-expansion, R-reflection, OC-outside and IC-inside contraction and v_2'/v_3'-shrink.
[a]Prepared in sketch-style drawing mode in Matplotlib (see: https://matplotlib.org/api/_as_gen/matplotlib.pyplot.xkcd.html).

2.3. Nelder-Mead algorithm with standard choice of parameters (SNMA)

As the original Nelder and Mead paper [1] contains several ambiguities, a standard ambiguity-free procedure given by Lagarias et al. [5] is reviewed herein. First, to describe Nelder-Mead algorithm, it is necessary to a priori define the four *standard* scalar parameters:

$$\alpha=1,\beta=2,\gamma=\delta=0.5. \quad (4)$$

The source of the standard parameters can be tracked down to the original paper [1] and the values should satisfy the following criteria:

$$\alpha>0,\beta>1,0<\gamma<1,0<\delta<1. \quad (5)$$

Furthermore, the values given in Eq. (4) govern the four possible simplex operations/steps (Scheme 1[2]) that occur during the iteration process. Although Spendley et al. [4] intention was to use simplex in the optimization of the mathematical functions, Nelder and Mead made the procedure broadly applicable by enabling simplex to exhibit a sequence of different steps: *reflection*, *expansion*, *contraction* or *shrinkage* [24]. By using the aforementioned steps, the simplex moves and each transformation is associated with α, β, γ and δ [25]. However, it should be accentuated that only one of the steps occurs in each iteration [5,12,26].

Second, the following procedure represents one NMA iteration, which is presented here in a generally accepted form (see, e.g. [5,12,17]):

1. Order/sort vertices in simplex, $O(\mathbf{x}_1)<O(\mathbf{x}_2)<\cdots<O(\mathbf{x}_{n+1})$, where $\mathbf{x}_1$ can be referred to as the *best* vertex, whilst $\mathbf{x}_{n+1}$ is denoted as the *worst* vertex.
2. Compute the *reflection* point $\mathbf{x}_r$ and evaluate $O(\mathbf{x}_r)$:

$$\mathbf{x}_r=\overline{\mathbf{x}}+\alpha(\overline{\mathbf{x}}-\mathbf{x}_{n+1}), \quad (6)$$

where $\overline{\mathbf{x}}$ is the centroid calculated by ignoring the worst point $\mathbf{x}_{n+1}$:

[2] Scheme was inspired by the Colloquium Talk given by Lixing Han (see: http://homepages.umflint.edu/~lxhan/NelderMead2013.pdf).

$$\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x_i}. \quad (7)$$

3. If $O(\mathbf{x}_1) \le O(\mathbf{x}_r) < O(\mathbf{x}_n)$ then accept $\mathbf{x}_r$ and terminate iteration step.
4. If $O(\mathbf{x}_r) < O(\mathbf{x}_1)$ then determine the *expansion* point $\mathbf{x}_e$ and evaluate $O(\mathbf{x}_e)$:

$$\mathbf{x}_e = \overline{\mathbf{x}} + \beta(\mathbf{x}_r - \overline{\mathbf{x}}) \quad (8)$$

If $O(\mathbf{x}_e) < O(\mathbf{x}_r)$ then accept $\mathbf{x}_e$ and terminate iteration step, otherwise (if $O(\mathbf{x}_e) \ge O(\mathbf{x}_r)$) accept $\mathbf{x}_r$ and terminate iteration step.

5. If $O(\mathbf{x}_r) \ge O(\mathbf{x}_n)$ then perform a contraction:
   a) If $O(\mathbf{x}_n) \le O(\mathbf{x}_r) < O(\mathbf{x}_{n+1})$ then perform an *outside contraction* and evaluate $O(\mathbf{x}_{oc})$:

$$\boldsymbol{x}_{oc} = \overline{\mathbf{x}} + \gamma(\mathbf{x}_r - \overline{\mathbf{x}}), \quad (9)$$

if $O(\mathbf{x}_{oc}) \le O(\mathbf{x}_r)$ then accept $\mathbf{x}_{oc}$ and terminate iteration step; otherwise, perform a shrink.

   b) If $O(\mathbf{x}_r) \ge O(\mathbf{x}_{n+1})$ then perform *inside contraction*, compute $\mathbf{x}_{ic}$ and evaluate $O(\mathbf{x}_{ic})$:

$$\mathbf{x}_{ic} = \overline{\mathbf{x}} - \gamma(\overline{\mathbf{x}} - \mathbf{x}_{n+1}) \quad (10)$$

If $O(\mathbf{x}_{ic}) < O(\mathbf{x}_{n+1})$ then accept $\mathbf{x}_{ic}$ and terminate iteration step.

6. Perform a *shrink* step, i.e. compute n points:

$$\mathbf{v}_i = \mathbf{x}_1 + \delta(\mathbf{x_i} - \mathbf{x}_1), i = 2, \ldots, n+1, \quad (11)$$

and form simplex (for next iteration) consisting of $\mathbf{x}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_{n+1}$ vertices.

The steps presented in the above procedure for $n = 2$ are shown in Scheme 1. The Nelder-Mead procedure given by Lagarias et al. [5] with a *standard* choice of parameters will be addressed here as SNMA.

2.4. Nelder-Mead algorithm with adaptive choice of parameters (ANMA)

The Nelder-Mead algorithm is designed to have an ability to adapt its simplex to the local landscape of mathematical functions [1,5,6]. Nevertheless, SNMA cannot adapt its simplex to the size of the CNLS problem (i.e. to the number of EEC parameters (n)). Therefore, to boost the descent property of SNMA when the number of parameters is ≥2, Gao et al. [12] proposed an application of *adaptive* choice of parameters (see Table 1):

Table 1
Impact of the CNLS problem size on the adaptive parameter values (see under ANMA) that govern four simplex operations.

Simplex operation	Parameters	SNMA	ANMA		
			$CNLS_{R(CR)}$	$CNLS_{R(CR)(CR)}$	$CNLS_{R(QR)(QR)}$
			$n = 3$	$n = 5$	$n = 7$
Reflection	α	1.00	1.00	1.00	1.00
Expansion	β	2.00	1.67	1.40	1.29
Contraction	γ	0.50	0.58	0.65	0.68
Shrink	δ	0.50	0.67	0.80	0.86

SNMA: Nelder-Mead algorithm with the standard choice of parameters; ANMA: Nelder-Mead algorithm with the adaptive choice of parameters.

$$\alpha = 1, \beta = 1 + \frac{2}{n}, \gamma = 0.75 - \frac{1}{2n}, \delta = 1 - \frac{1}{n}. \quad (12)$$

The values of the adaptive parameters choice obtained for different size of CNLS problems (i.e. $n = 3$, 5 and 7) are given in Table 1.

Please note that the aforementioned parameters (α, β, γ and δ) do not change during the iteration process and they should not be mistaken for the EEC parameters. In continuation of this work, the modification proposed by Gao et al. [12] with the *adaptive* choice of parameters (Eq. (12)) will be commented on in terms of ANMA.

2.5. Formation of the initial simplex

The Nelder-Mead algorithm requires a starting point ($\mathbf{x}_0$), which is used to form an initial simplex. The first vertex ($\mathbf{x}_1$) of the initial simplex is set to be equal to ($\mathbf{x}_0$):

$$\mathbf{x}_1 = \mathbf{x}_0. \quad (13)$$

According to Gao et al. [12], the remaining n vertices of the initial simplex can be computed by using:

$$\mathbf{x}_{k+1} = \mathbf{x}_0 + \tau_k \mathbf{e}_k, k = 1, \ldots, n, \quad (14)$$

where $\mathbf{e}_k$ is the unit vector with k^{th} component 1 and other components 0 and τ_k is a priori chosen as:

$$\tau_k = \begin{cases} 0.05 \text{ if } (\mathbf{x}_0)_k \neq 0, \\ 0.00025 \text{ if } (\mathbf{x}_0)_k = 0. \end{cases} \quad (15)$$

2.6. Iteration stopping criteria applied in the new fitting engine

The iteration procedure (see Section 2.3) continues until it is interrupted by stopping criteria. Dennis and Woods [27] gave a short discussion related to the application of several different stopping criteria in the Nelder-Mead algorithm. In line with the authors, the selection of proper stopping criterion is an important issue, since it is required to simultaneously focus on both objection function value and a size of the simplex. However, in this work a general approach was applied, i.e. iteration processes were stopped when both convergence tolerance on the objective function value (*TolFun*) and on the size of the step (*TolX*) were satisfied:

$$max_{2 \le k \le n+1} |O(\mathbf{x}_k) - O(\mathbf{x}_1)| \le TolFun, \quad (16)$$

and

$$max_{2 \le k \le n+1} \|\mathbf{x}_k - \mathbf{x}_1\|_\infty \le TolX. \quad (17)$$

Herein, the following value was used: $TolFun = TolX = 10^{-4}$.

3. Experimental

3.1. Polluted and non-polluted synthetic impedance data used in this work

The following electrical equivalent circuits $EEC_{R(CR)}$, $EEC_{R(CR)(CR)}$ and $EEC_{R(QR)(QR)}$ (Fig. 1) were used to prepare and fit the synthetic data ($Z_{synth}(\omega)$). The EEC parameters used to compute $Z_{synth}(\omega)$ are given in Tables 2–4 and the frequency (f) range varied from 0.01 Hz to 100 kHz, taking 10 points per decade.

The synthetic polluted ($Z_{poll}(\omega)$) data were prepared by adding noise to $Z_{synth}(\omega)$, which is a common approach when analyzing freshly developed fitting engines [8,21,22] and diverse algorithms [28,29]. Note that the noise intensity of $Z_{poll}(\omega)$ was gradually

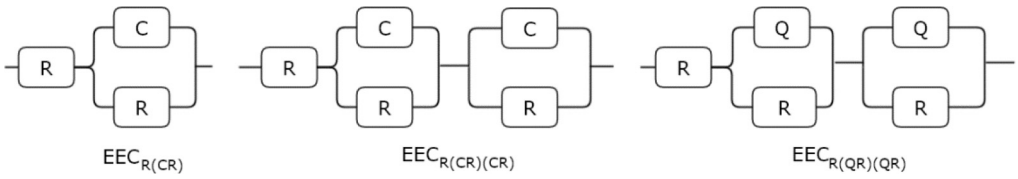


Fig. 1. Schematic presentation of equivalent electrical circuits (EECs) used in this work.

increased by multiplying noise with the noise factor (NF) values. It should also be noted that NF values are taken from arbitrarily chosen [0, 0.02] interval:

$$Z_{\text{poll}}(\omega) = Z_{\text{synt}}(\omega)\cdot(1+\text{NF}\cdot(\eta'+i\eta'')), \tag{18}$$

where η' and η'' are two independent normally distributed variables with zero mean and unit variance and ω is angular frequency, respectively. Although the application of, e.g. NF = 0.02, offers at least 2% noise, it should be mentioned that normally distributed noise (η' and η'') can take arbitrary large values (see Fig. 2).

3.2. CNLS problems solved in this work

EEC$_{R(CR)}$, EEC$_{R(CR)(CR)}$ and EEC$_{R(QR)(QR)}$ presented in Fig. 1 have different number of parameters (3, 5 and 7); and thus, corresponding CNLS$_{R(CR)}$, CNLS$_{R(CR)(CR)}$ and CNLS$_{R(QR)(QR)}$ problems are of different sizes (Table 5). Interestingly, when using starting fitting parameters from Tables 3–4, both EEC$_{R(CR)(CR)}$ and EEC$_{R(QR)(QR)}$ produce equal impedance data. Therefore, the same starting O-values were obtained when solving both CNLS$_{R(CR)(CR)}$ and CNLS$_{R(QR)(QR)}$ problems (Tables 3 and 4).

3.3. Open-source tools used in this work

The open-source Python programming language (v.2.7.15.) was used to develop a new NMA-based fitting engine. The following open-source Python modules were used:

- Numpy [30] v1.14.3,
- Matplotlib [31] v2.2.2,
- wxPython v3.0.2.0,
- Cython 0.23.3 and
- PyInstaller v3.3.1.

4. Results and discussion

4.1. Design of the new engine

The new engine was based on the procedure given in Section 2.5, although the engine was additionally modified to yield data that were used to monitor simplex distortions (see Section 4.4) and to compute simplex operation efficiency (see Section 4.6). It was decided to use larger edges of the initial simplex as they yield better

Table 2
Parameter values used to both prepare and solve polluted (NF = 0.02) CNLS$_{R(CR)}$ problem by using EEC$_{R(CR)}$.

Parameters	R_1 (Ω cm^2)	C_2 (F cm^{-2})	R_2 (Ω cm^2)	O-value	Number of iterations
Synthetic	10	1 10^{-4}	100		
Starting	1	0.001	60	42.62	
Final (SNMA)	9.953	1 10^{-4}	100.1	3.286 10^{-2}	128
Final (ANMA)	9.953	1 10^{-4}	100.1	3.286 10^{-2}	155
Final (M-ANMA[a])	9.953	1 10^{-4}	100.1	3.286 10^{-2}	162

[a] Modified ANMA.

Table 3
Parameter values used to both prepare and solve polluted (NF = 0.02) CNLS$_{R(CR)(CR)}$ problem by using EEC$_{R(CR)(CR)}$.

Parameters	R_1 (Ω cm^2)	C_2 (F cm^{-2})	R_2 (Ω cm^2)	C_3 (F cm^{-2})	R_3 (Ω cm^2)	O-value	Number of iterations
Synthetic	0.738	0.286	0.086	0.223	1723		
Starting	1	1	1	1	60	15.78	
Final (SNMA)	0.735	0.238	0.086	0.225	1688	3.389 10^{-2}	744
Final (ANMA)	0.735	0.238	0.086	0.225	1688	3.389 10^{-2}	1421
Final (M-ANMA[a])	0.735	0.238	0.086	0.225	1688	3.389 10^{-2}	624

[a] Modified ANMA.

Table 4
Parameter values used to both prepare and solve polluted (NF = 0.02) CNLS$_{R(QR)(QR)}$ problem by using EEC$_{R(QR)(QR)}$.

Parameters	R_1 (Ω cm^2)	Q_2 (S s^n cm^{-2})	n_2	R_2 (Ω cm^2)	Q_3 (S s^n cm^{-2})	n_3	R_3 (Ω cm^2)	O-value	Number of iterations
Synthetic	0.738	0.286	1	0.086	0.223	1	1723		
Starting	1	1	1	1	1	1	60	15.78	
Final (SNMA)	0.728	3.240	0.412	−47.98	0.236	1.02	1002	3.953 10^{-2}	1973
Final (ANMA)	0.735	0.355	0.901	0.093	0.224	1.00	1654	3.368 10^{-2}	2109
Final (M-ANMA[a])	0.735	0.355	0.901	0.093	0.224	1.00	1654	3.368 10^{-2}	1263

[a] Modified ANMA.

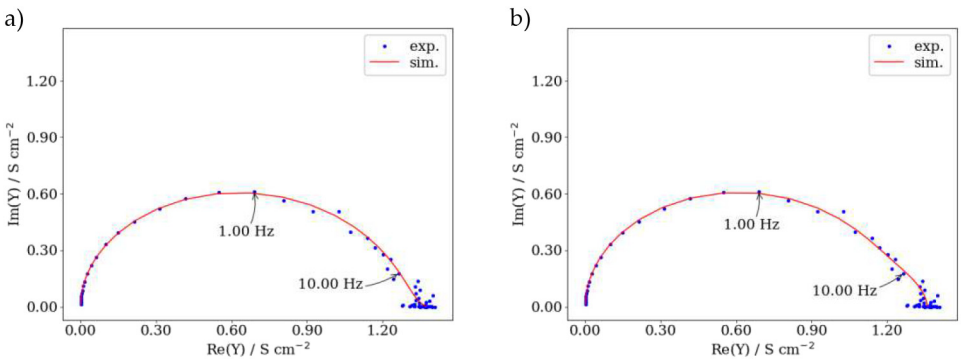


Fig. 2. The impedance spectra of two fits that were obtained by solving polluted (NF = 0.02) $CNLS_{R(QR)(QR)}$ problem by a) SNMA and b) ANMA. The symbol reference: the polluted synthetic (·) and simulated data (−).

performances when starting parameters are far from solution [12]. For the same reason, Dellis et al. [8] multiplied vertex coordinates by factor 1.1 (or 1.2):

$$\mathbf{x}_{k+1} = \mathbf{x}_0(1.1\mathbf{e}_k), k = 1, \ldots n. \tag{19}$$

Therefore, Eqs. (14), (19) were combined to form a more appropriate initial simplex:

$$\mathbf{x}_{k+1} = \mathbf{x}_0(1 + \tau_k\mathbf{e}_k), k = 1, \ldots n. \tag{20}$$

In order to facilitate computing tasks, data related to both **x** and $O(\mathbf{x})$ were used to form the following initial *simplex matrix* (SM):

$$\mathrm{SM} = \begin{bmatrix} x_{1,1} & \ldots & x_{1,n+1} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \ldots & x_{n,n+1} \\ O(\mathbf{x}_1) & \ldots & O(\mathbf{x}_{n+1}) \end{bmatrix} \tag{21}$$

The above data arrangement yields $n + 1$ by $n + 1$ matrix, which is straightforwardly handled in NumPy [30].

As NMA does not apply derivatives, it is incapable to produce a reliable information about errors in parameters values [9,10]. However, this drawback was resolved herein as the new engine approximates Hessian of the objective function (Eq. (1)) at the iteration's end. Consequently, Hessian matrix was used to compute the errors in EEC parameters values (not showed). This approach to compute error values by using Hessian is a common one when conducting data fitting (see, e.g. Section 3.2 in [21]). However, the Hessian matrix in EIS study has various applications, i.e. it can be used to obtain, e.g. optimal experiment design (OED) [32] (also see [33]).

4.2. Impact of the adaptive parameters choice when solving $CNLS_{R(QR)(QR)}$ problem

SNMA yields fair descent properties when optimizing problems with <3 variables [13,34]; and thus, $CNLS_{R(QR)(QR)}$ problem with 7 variables (Table 5) was solved in this section. Fig. 2a displays that SNMA fit failed as it yielded both a data mismatch between experimental and simulated data and a negative EEC parameter ($R_2 = -47.98\,\Omega\,\mathrm{cm}^2$) value (Table 4). On the other hand, ANMA fit was successful due to both a good data match and the absence of negative EEC values (Fig. 2b). It follows that the application of adaptive (vs. standard) parameters choice produced better simplex adaptation onto the local landscape of the $CNLS_{R(QR)(QR)}$ problem.

Table 5
EEC models used to prepare the synthetic impedance data and the corresponding synthetic CNLS problems.

EEC	Number of parameters (n)	Corresponding problem	Problem size
$EEC_{R(CR)}$	3	$CNLS_{R(CR)}$	3
$EEC_{R(CR)(CR)}$	5	$CNLS_{R(CR)(CR)}$	5
$EEC_{R(QR)(QR)}$	7	$CNLS_{R\ (QR)(QR)}$	7
$EEC_{R(QR)(QR)(QR)}$	10	$CNLS_{R\ (QR)(QR)(QR)}$	10

To extract additional information related to the simplex adaptation, $O_{SNMA/ANMA}$ vs. (Iteration number) data are given in Fig. 3. Fig. 3a shows that O_{SNMA}-value suddenly decreased (i.e. it declined) in 34th iteration, which was a result of the expansion step that elongated SNMA simplex (Scheme 1). Interestingly, Nelder and Mead enabled simplex to elongate itself in order to move through the local landscape [1,26]. However, newer insights indicate that the expansion step can yield some unwanted effects, i.e. it can induce simplex distortions [12] and it can cumulatively increase a *search angle* (an angle between the search direction and the direction of the local downhill gradient at the worst vertex [18]).

Furthermore, the ANMA expansion step did not induce any sudden jumps in O_{ANMA}-value, which suggests the absence of sudden simplex distortions (Fig. 3b). The aforementioned conclusions correspond to the literature [12], which claims that the application of the adaptive parameter choice generally alleviates simplex distortions. Still, the impact of the simplex step on both the search angle [18] and the simplex step efficiency has not been thoroughly explained yet. Therefore, in order to resolve the above issue, it was necessary to evaluate an impact of the simplex distortions onto algorithms' descent property (i.e. efficiency).

4.3. Impact of the simplex distortions on SNMA and ANMA descent properties

The simplex adapts itself to the local landscape of mathematical functions [1,5,6] through a series of distortions in shape/size. If distortions are moderate, they should not disturb algorithms' descent properties. These properties are usually tested in literature [15,17,18] by minimizing functions (from Moré et al. [35] dataset) with different mathematical landscapes. However, this engine was made for EIS data fitting; and thus, we decided to solve several different CNLS problems (Fig. 4). To be precise, the objective function was the same (Eq. (1)), whereas different EECs were used. Note that the above literature specifies that a superior NMA modification will yield lower O-value (Eq. (1)).

Fig. 4 represents $O_{SNMA/ANMA}$ values (vs. Noise factor) obtained by solving differently polluted CNLS problems (Table 5). It can be concluded that ANMA (vs. SNMA) was superior as it yielded lower O-value in 0,[3] 57.1 and 66.6% fitting attempts (Fig. 4a–c). These

[3] Equal SNMA and ANMA efficiency as the adaptive parameters are similar to the standard ones (Table 1).

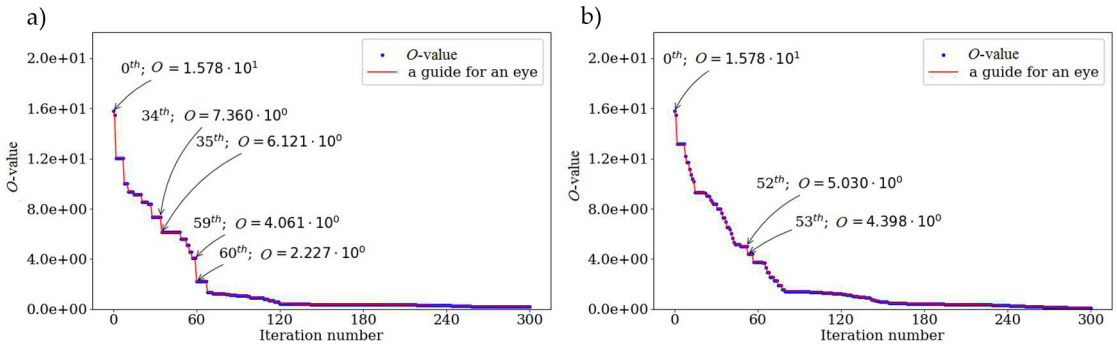


Fig. 3. The *O*-value vs. (Iteration number) data collected during solving polluted (NF = 0.02) $CNLS_{R(QR)(QR)}$ problem by using a) SNMA and b) ANMA. Only first 300 iterations were shown.

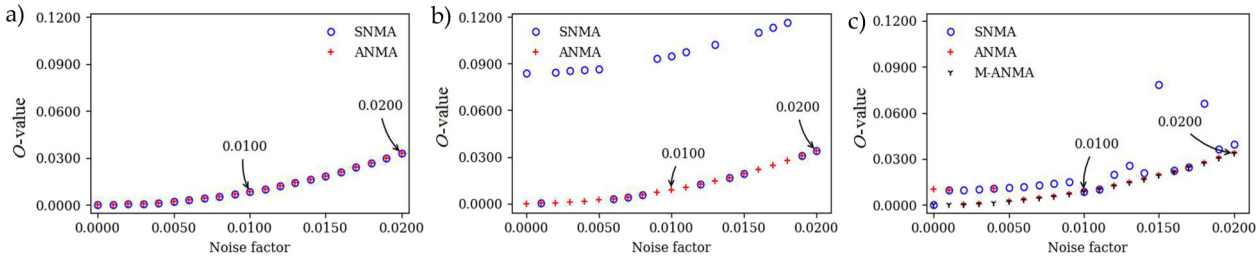


Fig. 4. The *O*-value vs. (Noise factor) data obtained when solving differently polluted a) $CNLS_{R(CR)}$, b) $CNLS_{R(CR)(CR)}$ and c) $CNLS_{R(QR)(QR)}$ problems by SNMA and ANMA. The symbol reference: data obtained by SNMA (○), ANMA (+), and by M-ANMA (▾).

results are in agreement with literature [15,17,18] that presented that ANMA generally outperforms SNMA. Therefore, according to [12], alleviated simplex distortions yielded a superior ANMA descent/fitting property. Please be advised that until now, the enhanced ANMA properties in EIS study have not been interpreted yet in the terms of the simplex step efficiency, which can be computed by using data related to simplex distortions.

4.4. Simplex distortion monitoring

The simplex size distortion occurs when the size of the simplex at the beginning of t^{th} and $t^{\text{th}}+1$ iterations is different (Scheme 2). Therefore, it should be motivating to detect the intensity of simplex distortion during the iteration process as they have an impact on descent property [12]. Be advised that Fajfar et al. [18] also monitored simplex-related properties during iterations, which resulted in a new scientific insight related to the simplex search direction.

Herein, it was proposed to monitor a simplex distortion by computing distortion in simplex size[4] (DSS):

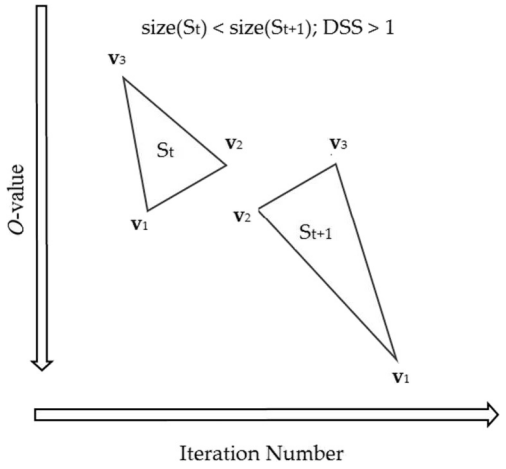
$$\text{DSS} = \frac{\text{size}(S_{t+1})}{\text{size}(S_t)}, \tag{22}$$

where S_t and S_{t+1} represents simplex (S) in t^{th} and $t^{\text{th}}+1$ iteration.

In order to enable a more comprehensive analysis, several issues related to DSS value should be clarified. First, the simplex in t^{th} iteration undergoes distortion in size only if one of the simplex steps (Table 1) produces DSS ≠ 1. This means that we can determine the number of simplex steps (e.g. expansion step) that produced DSS ≠ 1. Second, if the simplex size is increased, then DSS value with a physical meaning is >1. Oppositely, if the simplex size is decreased, then DSS value must be <1. Third, if the simplex distortions are more intense, then DSS values should deviate more from 1. Thus, the higher/lower DSS variance values indicate sudden/alleviated simplex distortions. And finally, DSS value can be used to compute the simplex step efficiency if this specific step decreased *O*-value.

4.5. Distortion in simplex size (DSS) when solving polluted $CNLS_{R(QR)(QR)}$ problem

In order to investigate whether computed DSS values possess a physical meaning, the polluted $CNLS_{R(QR)(QR)}$ problem was resolved and DSS values (≠ 1) obtained by the reflection, expansion, outside contraction and inside contraction steps are displayed in Fig. 5a–d. Data in Fig. 5 was processed and displayed in Table 6, which displays the number (%) of distortions (vs. total distortion number)



Scheme 2. Schematic representation of one simplex distortion induced by the expansion step. The size of S_{t+1} is $>S_t$ which yields DSS > 1. Note that the *O*-value of vertex $\mathbf{v}_1$ of S_{t+1} is lower than the *O*-value of vertex $\mathbf{v}_1$ of S_t. Symbol reference: $\mathbf{v}_1$-best vertex, $\mathbf{v}_2$-next-to-worst, $\mathbf{v}_3$-worst vertex.

[4] $\text{size(S)} = \frac{\max_{2\le i\le n+1} \|\mathbf{x}_i - \mathbf{x}_1\|}{\max(1, \|\mathbf{x}_1\|)}$; [36].

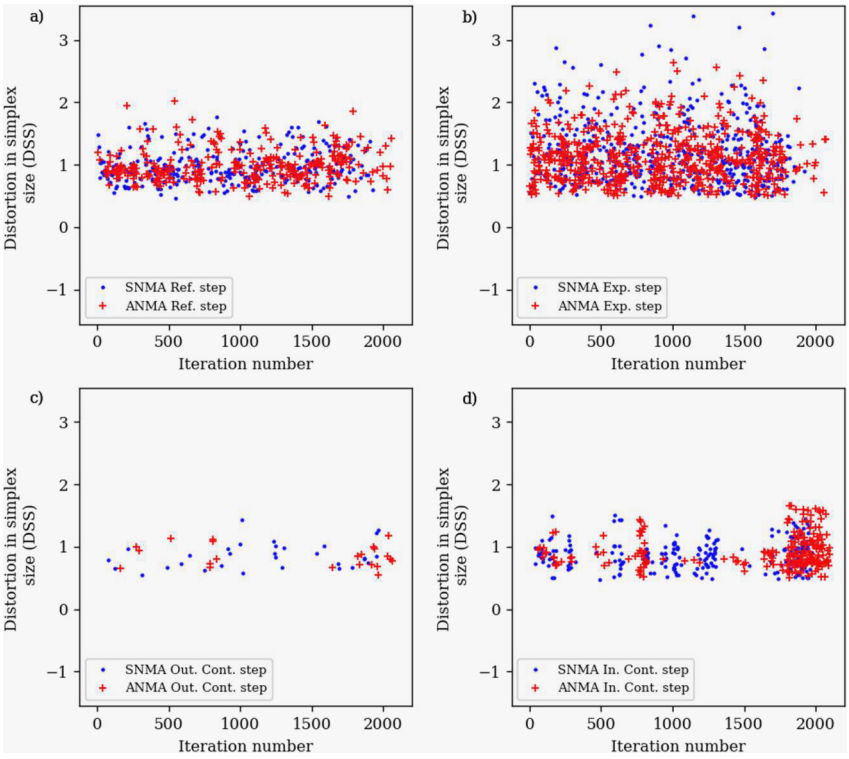


Fig. 5. DSS vs. (Iteration number) values obtained whilst solving polluted (NF = 0.02) $CNLS_{R(QR)(QR)}$ problem. The symbol reference: values obtained by SNMA (●) and by ANMA (+).

that was obtained by each simplex step. Note that Fig. 5 presents simplex distortions that have not necessarily produced a O-value decrease.

The first simplex operation in SNMA/ANMA is the reflection step, which reflects the worst vertex through the centroid (Scheme 1). This step regularly induced distortions (Fig. 5a) and it commonly yields a new vertex with a lower function value. However, in the case of a superior ANMA, the number of distortions induced by the reflection step was lower (26.45% < 28.44%, Table 6). This is expected as a frequent occurrence of this step [12,15,17] decreases SNMA descent property. Furthermore, the ANMA (vs. SNMA) reflection step produced lower DSS variance value (Table 7), which indicates alleviated simplex distortions (Scheme 3). This agrees with work by Gao et al. [12], which claims that ANMA simplex distortions are alleviated. Thus, the reflection step DSS values obtained in this work correspond to the literature [12,15,17].

Second simplex operation is the expansion step (Fig. 5b), which possesses a descent property [12]. The distortions induced by the ANMA (vs. SNMA) expansion step were more frequent (52.37% > 46.65%, Table 6), since the adaptive parameter choice

Table 6
Number (%) of simplex distortions induced by each step vs. total number of distortions. Data were obtained when solving polluted (NF = 0.02) $CNLS_{R(QR)(QR)}$ problem. Values in parentheses represent data obtained from the last 300 iterations.

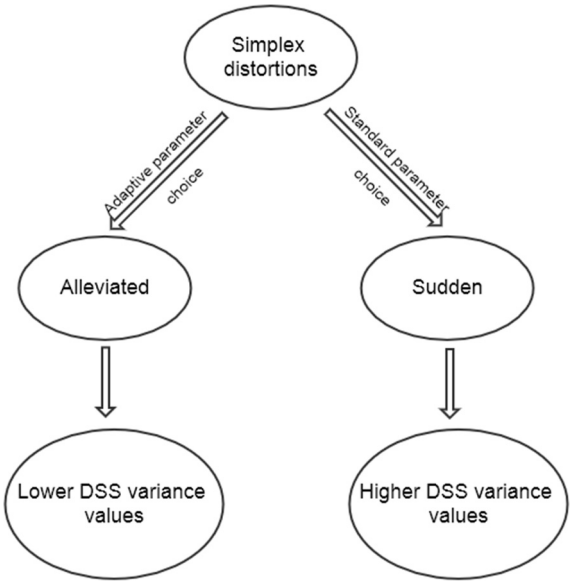
Simplex operation	Number of distortions in simplex size (%)	
	SNMA	ANMA
Reflection	28.44(17.33)	26.45(8.89)
Expansion	46.65(23.33)	52.37(7.22)
Outside contraction	2.88(6.00)	1.93(6.67)
Inside contraction	22.03(53.33)	18.80(74.44)

SNMA: Nelder-Mead algorithm with the standard choice of parameters; ANMA: Nelder-Mead algorithm with the adaptive choice of parameters.

Table 7
Mean of DSS values presented in Fig. 5a–d. Values in parenthesis represent variance.

Simplex operation	DSS	
	SNMA	ANMA
Reflection	0.9689(0.0647)	0.9779(0.0621)
Expansion	1.2068(0.2669)	1.1171(0.1576)
Out. contraction	0.8587(0.0422)	0.8533(0.0306)
Inside contraction	0.8675(0.0478)	0.9313(0.0695)

SNMA: Nelder-Mead algorithm with the standard choice of parameters; ANMA: Nelder-Mead algorithm with the adaptive choice of parameters.



Scheme 3. Schematic representation of impact of simplex distortions (alleviated or sudden) on DSS variance value.

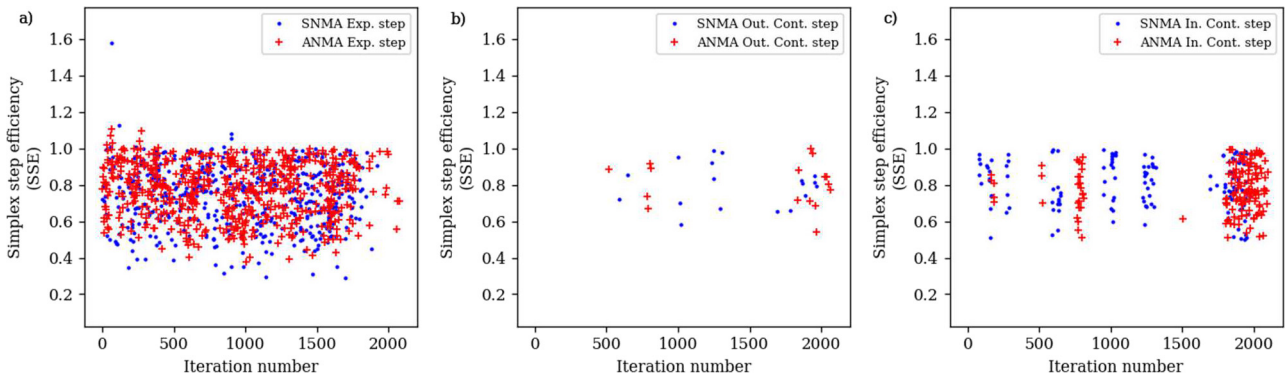


Fig. 6. SSE vs. (Iteration number) values obtained whilst solving polluted (NF = 0.02) $CNLS_{R(QR)(QR)}$ problem. The symbol reference: values obtained by SNMA (●) and by ANMA (+).

decreases manifestation of the reflection step [17]. As this step yields a more elongated simplex [26] (Schemes 1 and 2), several additional facts can be revealed. First, DSS mean values (1.2068 and 1.1171, Table 7) obtained by SNMA and ANMA expansion steps are >1, which confirms that this step increased the simplex size. Thus, DSS data related to the expansion step have a physical meaning. Second, the application of lower β_{ANMA} value (Table 1) resulted in lower DSS variance value (0.1576 < 0.2669, Table 7), which indicates that simplex distortions were alleviated [12] (Scheme 3). Overall, DSS values obtained by the expansion step have a physical meaning and they are in agreement with previously reported papers [12,17,26].

The third and fourth operations in SNMA/ANMA are the outside (Fig. 5c) and inside (Fig. 5d) contraction steps. In keeping with literature findings, these steps are the only ones that do not cumulatively increase the search angle [18] and they have descent properties [12]. Therefore, it is not a coincidence that a more successful ANMA (vs. SNMA) induced a higher number of simplex distortions (81.11% > 59.33%, Table 6) in the last 300 iteration. As these contraction steps generally decrease the relative simplex size (see [26] and Scheme 1), the corresponding DSS mean values (0.8587, 0.8675, 0.8533 and 0.9313 < 1, Table 7) have a physical meaning. Therefore, the aforementioned DSS data correspond to literature [12,26], which additionally supports the proposed approach to monitor simplex distortion data.

To summarize, the expansion and contraction steps yielded DSS mean values that are > and < 1, which confirms that simplex monitoring process yields DSS values with a physical meaning. What is more, the ANMA (vs. SNMA) expansion step DSS variance value were lower, which indicates that ANMA simplex distortions were more alleviated (Scheme 3). However, it still has not been clarified how the adaptive parameters choice improves ANMA descent properties. Therefore, this simplex distortions monitoring process should be further upgraded to yield the simplex step efficiency data.

4.6. Simplex step efficiency

If one of the simplex steps in t^{th} iteration induces both simplex distortion and reduction in O-value (see Scheme 2), then it would be reasonable to compute the simplex step efficiency (SSE) by:

$$\mathrm{SSE} = \frac{O_t/O_{t+1}}{max(\mathrm{size}(S_t), \mathrm{size}(S_{t+1}))/min(\mathrm{size}(S_t), \mathrm{size}(S_{t+1}))}, \quad (23)$$

where size (S_t) and size (S_{t+1}) represent simplex size in t^{th} and $t^{th}+1$ iterations and where O with indexes t and $t+1$ are the values of the minimized function in t^{th} and $t^{th}+1$ iterations. Therefore, a higher SSE value indicates a greater efficiency of the corresponding simplex step.

Fig. 6 displays SSE vs. (Iteration number) data obtained by the expansion, outside and inside contraction steps. According to Fig. 6a and Table 8, average/mean SSE value of ANMA (vs. SNMA) expansion step was higher (0.7725 vs 0.7472), which indicates that the application of lower β_{ANMA} value (Table 1) improved the average expansion step efficiency. This can be explained by the fact that even though the SNMA expansion step cumulatively increases the search angle [18], this 'negative' impact was somehow damped as lower β_{ANMA} value produced a restricted simplex elongation. Interestingly, the lower SSE variance value (0.0220 vs 0.0285) also supports the fact that ANMA (vs. SNMA) expansions step was restricted (Table 8).

The contraction steps (Fig. 6b and c) were the most effective steps in both SNMA and ANMA (Table 8), which is expected as they do not produce cumulative deviation from the downhill gradient direction [18]. The aforementioned observation corresponds to the literature which claims that contraction steps have descent property [12]. Furthermore, the application of γ_{ANMA} increased SSE mean of the outside contraction steps, whilst it decreased the SSE mean of the inside contraction step from 0.8075 to 0.7784 to (Table 8). This is rather unusual, as in the last 300 iterations the inside (vs. outside) contraction step occurred ≈ 10 times more frequently (Table 6). Therefore, an attempt should be made to increase the inside contraction step efficiency which can be accomplished by, e.g. modifying the existing ANMA.

To recapitulate, it turns out that the application of the adaptive parameters increased the average efficiency of the expansion and outside contraction steps (Table 8) which yielded better ANMA descent properties. However, at this stage of study, it is not clear why the adaptive parameter choice decreases the inside contraction step efficiency.

4.7. Application of the new method, i.e. boosting the existing ANMA

The proposed method demonstrated that greater γ_{ANMA} (vs. γ_{SNMA}) value decreased only the inside contraction step efficiency (Table 8).

Table 8
Mean and variance (in parenthesis) values of simplex step efficiency (SSE) data obtained when fitting presented in Fig. 6.

Simplex operation	Simplex step efficiency (SSE)		
	SNMA	ANMA	M-ANMA
Expansion	0.7472(0.0285)	0.7725(0.0220)	0.7832(0.0200)
Outside contraction	0.7980(0.0132)	0.8049(0.0139)	0.7886(0.0068)
Inside contraction	0.8075(0.0167)	0.7784(0.0156)	0.8216(0.0123)
	0.7842[a]	0.7852[a]	0.7977[a]

SNMA: Nelder-Mead algorithm with the standard choice of parameters; ANMA: Nelder-Mead algorithm with the adaptive choice of parameters; M-ANMA: modified ANMA.
[a] Mean SSE value.

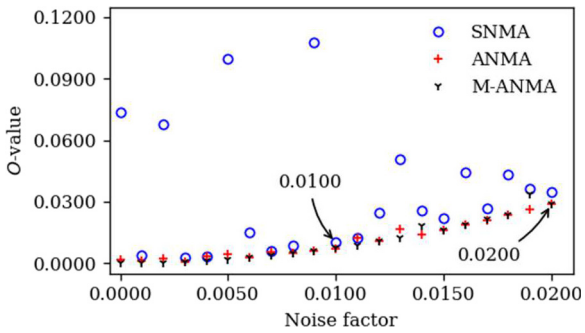


Fig. 7. The *O*-value vs. (Noise factor) data obtained when using $EEC_{R(QR)(QR)(QR)}$ model to fit differently polluted impedance data characterized by 3 different time constants (data not showed). The symbol reference: data obtained by SNMA (○), ANMA (+), and by M-ANMA (ϒ).

Table 9
Mean and variance (in parenthesis) values of simplex step efficiency (SSE) obtained when using $EEC_{R(QR)(QR)(QR)}$ model to fit polluted (NF = 0.02) impedance data characterized by 3 different time constants (data not showed) (see Fig. 7).

Simplex operation	Simplex step efficiency (SSE)		
	SNMA	ANMA	M-ANMA
Expansion	0.7760(0.0262)	0.8119(0.0184)	0.8018(0.0189)
Outside contraction	0.7805(0.0113)	0.7743(0.0097)	0.8607(0.0055)
Inside contraction	0.8390(0.0116)	0.8155(0.0154)	0.8242(0.0126)
	0.7985[a]	0.8005[a]	0.8288[a]

SNMA: Nelder-Mead algorithm with the standard choice of parameters; ANMA: Nelder-Mead algorithm with the adaptive choice of parameters; M-ANMA: modified ANMA.
[a] Mean SSE value.

This can be explained by the fact that *larger* γ_{ANMA} value induced formation of a new vertex which is *too close* to the worst vertex (see Scheme 1). This clearly indicates that γ_{ANMA} value in Eq. (10) should be decreased. Therefore, it was decided to modify the existing ANMA (M-ANMA) which was accomplished by using $0.95\gamma_{ANMA}$ instead of γ_{ANMA} in Eq. (10). Note that 0.95 value was chosen by experiment and its choice was not further elaborated herein.

Interestingly, the application of M-ANMA (vs. ANMA) improved the efficiency of both inside contraction step (from 0.7784 to 0.8216) and expansion step (from 0.7725 to 0.7832) (Table 8). On the other hand, only the M-ANMA outside contraction step efficiency was reduced. However, this is not an issue since the occurrence of this step was rather low (<1%). Furthermore, similar O_{M-ANMA} (not shown) and O_{ANMA} curves (Fig. 3b) in the first 300 iterations and different number of total iterations (1263 vs. 2109, Table 4) suggest that M-ANMA has a better descent property[5] in the final stage of the fit. This can be explained by the fact that the application of $0.95\gamma_{ANMA}$ improved a decrease in the simplex size, especially in the final stage of the fit.

Moreover, when the exercise in, e.g. Fig. 4c was reproduced by using a larger $EEC_{R(QR)(QR)(QR)}$ model (see Fig. 7) to analyze impedance data (not showed) characterized by 3 different time constants ($\Delta\tau = 1$ s), ANMA (vs. SNMA) was more superior in 89.47% fitting attempts. Again, M-ANMA (vs. ANMA) was more efficient as it yielded positive EEC parameters in 21 (vs. 17) fitting attempts. Furthermore, the application of a larger $EEC_{R(QR)(QR)(QR)}$ model generally increased the SSE values in Table 9 (vs. Table 8).

However, this particular event can be explained by the fact that the application of the higher *n* value (10 vs. 7) yielded both i) lower denominator values in Eq. (23) and ii) lower β value which additionally decreased the search angle. The lower denominator values in Eq. (23) can be attributed to a more deviated simplex shape at higher dimensions.

To recapitulate, M-ANMA (vs. ANMA) presented a better descent property and even when the number of parameters was increased to 10, they were more superior to SNMA. However, for some general conclusions, M-ANMA should be tested by using a plethora of functions from Moré et al. [35] dataset, which is beyond the scope of this manuscript. Although the application of $0.95\gamma_{ANMA}$ has not been mathematically justified, the intention of this work was to show that the simplex step efficiency values can be useful when investigating freshly developed NMA modifications.

This work also shows that a new tactic to define the simplex size (used in Eq. (23)) should be considered in EIS study as the simplex shape can drastically deviate due to the fact that EEC parameters can take a wide range of values (see final values in, e.g. Table 4). Although the application of ANMA (vs. SNMA) increased the simplex step efficiency, this study demonstrates that an increased number of dimensions is still a problem when analyzing EIS data. Therefore, one should consider the application of the NMA modification(s) with, e.g. a constant number of the simplex vertices [17].

5. Conclusion

It was presented that ANMA simplex was adapted with more success to the different landscape of noisy complex nonlinear least squares (CNLS) problems. This was accomplished by the application of the adaptive parameter choice that alleviates simplex distortions.

Herein, the distortions in simplex size (DSS) were monitored. The expansion step produced DSS mean values >1, which is expected as this step increases the simplex size. Oppositely, DSS mean values <1 were obtained by the contraction steps as they decrease the simplex size. It was explained that this monitoring approach is trustworthy, since it yields DSS data with a physical meaning.

The monitoring approach was additionally justified as ANMA (vs. SNMA) DSS variance values were lower, which indicates an alleviated ANMA simplex distortion. Therefore, by comparing DSS variance values, it was possible to monitor a simplex adaptation onto the local landscape.

It was elucidated that the ANMA (vs. SNMA) expansion step has a greater efficiency as the lower β_{ANMA} value restricted simplex elongation in the direction that deviates from the downhill gradient direction at the worst vertex. On the other hand, the larger γ_{ANMA} value increased the outside contraction step efficiency because it enabled the formation of the new vertex further away from the centroid.

The findings in this work showed that the ANMA inside contraction step efficiency was decreased as higher γ_{ANMA} value induced formation of a new vertex that is too close to the worst vector. This problem was avoided by modifying the existing ANMA (i.e. M-ANMA) by using $0.95\gamma_{ANMA}$ value. Consequently, it was presented that the application of $0.95\gamma_{ANMA}$ induced better M-ANMA (vs. ANMA) descent properties.

In the end, to enable a wider usage of SNMA, ANMA and M-ANMA fitting engines, they were embedded in MIT licensed[6] software by using Python programming language and hosted online (see [37]).

List of abbreviations and symbols

NMA	Nelder-Mead algorithm
SNMA	standard Nelder-Mead algorithm
ANMA	adaptive Nelder-Mead algorithm

[5] According to Fig. 4c, M-ANMA (vs. ANMA) yielded lower *O*-value in 14.4% fitting attempts.

[6] See https://opensource.org/licenses/MIT.

M-ANMA	modified adaptive Nelder-Mead algorithm
EIS	electrochemical impedance spectroscopy
EEC	electrical equivalent circuit
CNLS	complex nonlinear least square problem
DSS	distortion in simplex size
SSE	simplex step efficiency
OED	optimal experiment design
S	simplex
S_t	simplex in t^{th} iteration
S_{t+1}	simplex in $t^{th}+1$ iteration
R	resistor
C	capacitance
Q	constant phase element
$Z_{synth}(\omega)$	synthetic EIS data
$Z_{poll}(\omega)$	polluted synthetic EIS data
NF	noise factor
O	objective function used in EIS data fitting
n	number of dimensions, i.e. number of EEC parameters
x_n	n^{th} EEC parameter
f	frequency
N	number of impedance data points
ω	angular frequency ($\omega = 2\pi f$)
w	modulus weighting factor
Y	experimental EIS data
y	computed EIS data
α, β, γ, and δ	values associated with reflection, expansion, contraction (inside and outside) and shrink steps
$\mathbf{x}$	vertex of simplex
$\mathbf{x}_r$, $\mathbf{x}_e$, $\mathbf{x}_{ic}$ and $\mathbf{x}_{oc}$	reflection, expansion, inside contraction and outside contraction points (i.e. vertices)
$\overline{\mathbf{x}}$	centroid

Acknowledgments

The authors gratefully acknowledge the stimulation program "Joint Excellence in Science and Humanities" (JESH-2017) of the Austrian Academy of Sciences for providing supporting funds.

References

[1] J.A. Nelder, R. Mead, A simplex-method for function minimization, Comput. J. 7 (4) (1965) 308–313.
[2] C.T. Kelley, Iterative Methods for Optimization, SIAM, Philadelphia, 1999.
[3] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, New York, 1999.
[4] W. Spendley, G.R. Hext, F.R. Himsworth, Sequential application of simplex designs in optimisation and evolutionary operation, Technometrics 4(4) (1962) 441-&.
[5] J.C. Lagarias, J.A. Reeds, M.H. Wright, P.E. Wright, Convergence properties of the Nelder-Mead simplex method in low dimensions, SIAM J. Optim. 9 (1) (1998) 112–147.
[6] A.R. Conn, K. Scheinberg, L.N. Vicente, Introduction to Derivative-free Optimization Introduction, Introduction to Derivative-free Optimization vol. 8, 2009.
[7] W.H. Press, Numerical Recipes 3rd Edition: The Art of Scientific Computing, Cambridge University Press, New York, NY, USA, 2007.
[8] J.L. Dellis, J.L. Carpentier, Nelder and Mead algorithm in impedance spectra fitting, Solid State Ionics 62 (1–2) (1993) 119–123.
[9] F. James, M. Roos, Minuit - a system for function minimization and analysis of the parameter errors and correlations, Comput. Phys. Commun. 10 (6) (1975) 343–367.
[10] F. James, M. Winkler, Minuit User's Guide, 2004.
[11] E. Barsoukov, J.R. Macdonald, Impedance Spectroscopy: Theory, Experiment, and Applications, 2005.
[12] F.C. Gao, L.X. Han, Implementing the Nelder-Mead simplex algorithm with adaptive parameters, Comput. Optim. Appl. 51 (1) (2012) 259–277.
[13] K.I.M. McKinnon, Convergence of the Nelder-Mead simplex method to a nonstationary point, SIAM J. Optim. 9 (1) (1998) 148–158.
[14] N. Pham, B.M. Wilamowski, Improved Nelder Mead's simplex method and applications, J. Comput. 3 (2011) 55–63.
[15] I. Fajfar, J. Puhan, A. Burmen, Evolving a Nelder-Mead algorithm for optimization with genetic programming, Evol. Comput. 25 (3) (2017) 351–373.
[16] M.J. Blondin, J. Sanchis, P. Sicard, J.M. Herrero, New optimal controller tuning method for an AVR system using a simplified ant colony optimization with a new constrained Nelder–Mead algorithm, Appl. Soft Comput. 62 (2018) 216–229.
[17] H.A. Musafer, A. Mahmood, Dynamic Hassan Nelder Mead with simplex free selectivity for unconstrained optimization, IEEE Access 6 (2018) 39015–39026.
[18] I. Fajfar, Á. Bűrmen, J. Puhan, The Nelder–Mead simplex algorithm with perturbed centroid for high-dimensional function optimization, Optim. Lett. 13 (5) (2018) 1011–1025, https://link.springer.com/article/10.1007/s11590-018-1306-2.
[19] W.H. Press, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, Cambridge [Cambridgeshire]; New York, 1988.
[20] R.J. Sheppard, B.P. Jordan, E.H. Grant, Least squares analysis of complex data with applications to permittivity measurements, J. Phys. D. Appl. Phys. 3 (11) (1970) 1759.
[21] M. Zic, An alternative approach to solve complex nonlinear least-squares problems, J. Electroanal. Chem. 760 (2016) 85–96.
[22] M. Zic, Solving CNLS problems by using Levenberg-Marquardt algorithm: a new approach to avoid off-limits values during a fit, J. Electroanal. Chem. 799 (2017) 242–248.
[23] F. Ciucci, Modeling electrochemical impedance spectroscopy, Curr. Opin. Electrochem. 13 (2019) 132–139.
[24] D.M. Olsson, L.S. Nelson, Nelder-Mead simplex procedure for function minimization, Technometrics 17 (1) (1975) 45–51.
[25] S. Agrawal, D. Singh, Modified Nelder-Mead self organizing migrating algorithm for function optimization and its application, Appl. Soft Comput. J. 51 (2017) 341–350.
[26] M.H. Wright, Nelder, Mead, and the other simplex method, Doc. Math. Extra (2012) 271–276. ISMP.
[27] J.E. Dennis Jr., D.J. Woods, Optimization on microcomputers: the Nelder-Mead simplex algorithm, in: New Computing Environments: Microcomputers in Large-Scale Computing, 1987, pp. 116–122.
[28] M.B. Effat, F. Ciucci, Bayesian and hierarchical Bayesian based regularization for deconvolving the distribution of relaxation times from electrochemical impedance spectroscopy data, Electrochim. Acta 247 (2017) 1117–1129.
[29] T.H. Wan, M. Saccoccio, C. Chen, F. Ciucci, Influence of the discretization methods on the distribution of relaxation times deconvolution: implementing radial basis functions with DRTtools, Electrochim. Acta 184 (2015) 483–499.
[30] S. van der Walt, S.C. Colbert, G. Varoquaux, The NumPy Array: a structure for efficient numerical computation, Comput. Sci. Eng. 13 (2) (2011) 22–30.
[31] J.D. Hunter, Matplotlib: a 2D graphics environment, Comput. Sci. Eng. 9 (3) (2007) 90–95.
[32] F. Ciucci, T. Carraro, W.C. Chueh, W. Lai, Reducing error and measurement time in impedance spectroscopy using model based optimal experimental design, Electrochim. Acta 56 (15) (2011) 5416–5434.
[33] F. Ciucci, Revisiting parameter identification in electrochemical impedance spectroscopy: weighted least squares and optimal experimental design, Electrochim. Acta 87 (2013) 532–545.
[34] J.C. Lagarias, B. Poonen, M.H. Wright, Convergence of the restricted Nelder-Mead algorithm in two dimensions, SIAM J. Optim. 22 (2) (2012) 501–532.
[35] J.J. More, B.S. Garbow, K.E. Hillstrom, Testing unconstrained optimization software, ACM Trans. Math. Softw. 7 (1) (1981) 17–41.
[36] D.J. Woods, An Interactive Approach for Solving Multi-objective Optimization Problems, An Interactive Approach for Solving Multi-objective Optimization Problems, 1985.
[37] EisPy v.4.02 is hosted at. https://goo.gl/Hd9eUN.