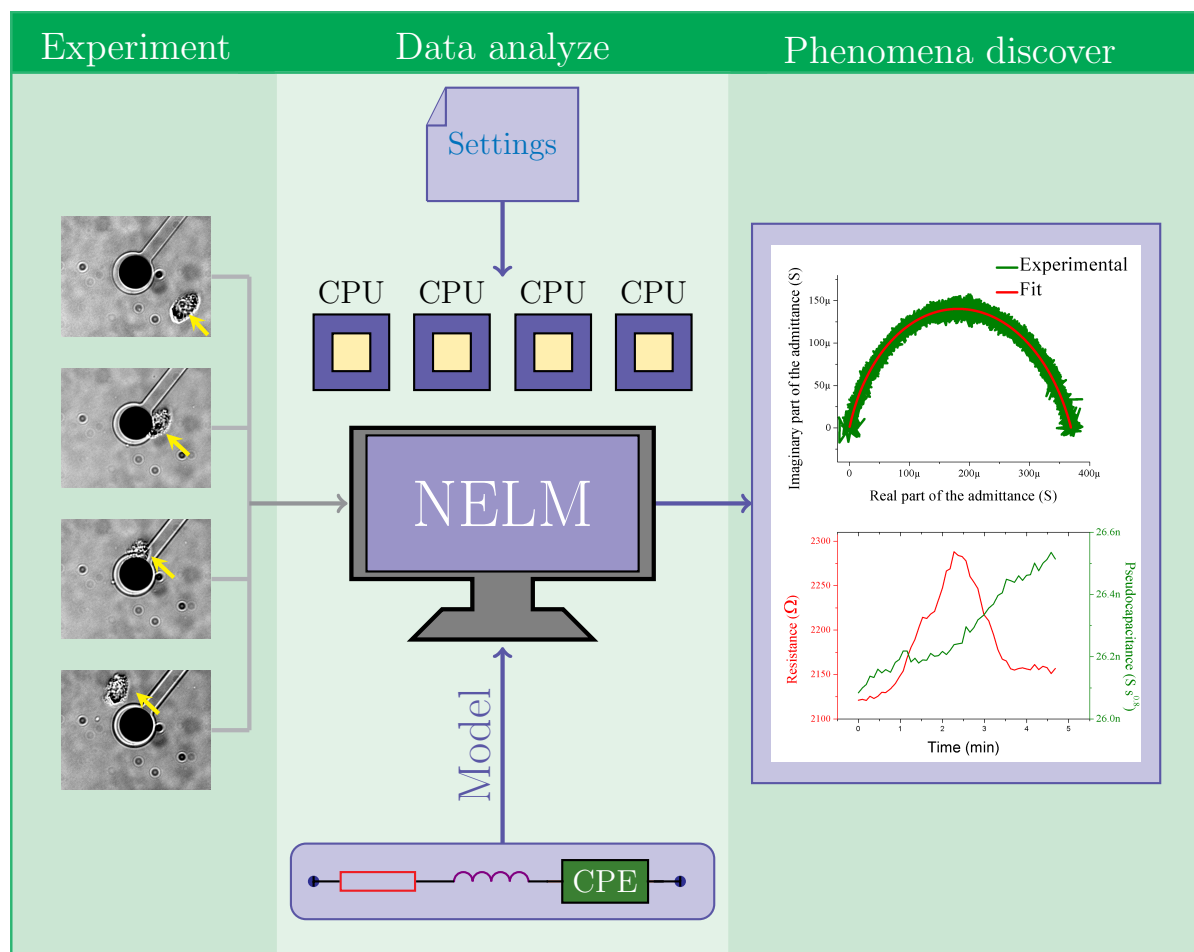


Graphical Abstract

NELM: Modern Open-Source Software for Multipurpose Impedance Spectra Analysis

Natalia A. Boitsova, Anna A. Abelit, Daniil D. Stupin



Highlights

NELM: Modern Open-Source Software for Multipurpose Impedance Spectra Analysis

Natalia A. Boitsova, Anna A. Abelit, Daniil D. Stupin

- Multipurpose open-source software NELM for impedance spectra analyzing was designed
- NELM was written in well-tuned MATLAB and is supported by the power of the parallel and symbolic computing
- Program efficiency was demonstrated on the different scientific tasks: from semiconductors to biology

NELM: Modern Open-Source Software for Multipurpose Impedance Spectra Analysis

Natalia A. Boitsova¹, Anna A. Abelit¹, Daniil D. Stupin¹

^a*Alferov University Khlopina st., 8/3, St. Petersburg, 194021, Russian Federation*

Abstract

Nowadays electrical impedance spectroscopy (EIS) has become an advanced experimental technique with a wide range of applications: from simple passive circuits diagnostics to semiconductor high-end device development and breakthrough technologies in bio-sensing. Although hardware for EIS today is well developed, the EIS analysis software is mainly custom, old fashioned, *i.e.* it is limited by features, does not utilize the progress in the modern computer science and hardware, and is usually implemented in close-source code or written on outdated programming languages, which causes slow progress in field of the EIS and complicates researchers attempts of development in EIS autonomous devices, such as implants. In this article, we introduce a free and open-sourced MATLAB/GNU OCTAVE package for EIS analysis called NELM, which provides powerful equipment tools for matching experimental impedance data with theoretical equivalent circuits. Our software has an user friendly interface and supports different formats of input data, fitting programs, and impedance models. In addition, we have developed NELM with implementation of the latest progress in computation science such as symbolic calculations, parallel computing, and artificial intelligence. The abilities of NELM were validated by its applications in the different fields of science, such as semiconductor studies, bioimpedance and electrochemistry, which demonstrated high-efficiency of the proposed software package and showed that it is a promising tool for solving actual problems in electronic industry, biosensorics, and healthcare technologies.

Keywords:

impedance spectroscopy, CNLS, biosensing, data analysis, adaptive filtering, semiconductor studies, equivalent circuit, cell research, multielectrode array, MatLab, GNU Octave

Email address: `natab2002@yandex.ru` (Natalia A. Boitsova)

Contents

1	Introduction	2
2	NELM structure, “phrase-book”, and “guidebook”	5
3	Step-by-step guide and program block-scheme	11
4	Overview of the NELM’s immittance analysis techniques	12
4.1	State-of-the-art CNLS processing	12
4.2	Monte-Carlo CNLS stabilization	16
4.3	Adaptive filtering approach	17
4.4	User-defined CNLS solvers and models	18
4.4.1	Adding new methods of minimization to NELM	18
4.4.2	Model interface description	18
5	Software design materials and methods	22
5.1	Hardware and software requirements	22
5.2	Structure of the program and operational principle	22
6	NELM application for real-life systems research	23
6.1	RLC circuit and semiconductor study	24
6.2	Cells research	26
6.3	AI-based adaptive filtering approach for noise combat	29
7	Conclusion and outlooks	32

1. Introduction

Typical experimental science practices include two essential steps: data collection and data processing [1–8]. The final aim of the last step usually is to find the relationships in the experimental data, which can be achieved by matching it to some *ab initio* theory or to appropriate phenomenological models. For impedance spectroscopy, both theoretic and phenomenological models can often be represented in terms of equivalent circuits (EC), electrical schemes that have the same impedance as the sample under test [9–11]. The advantage of such an approach consists in the possibility of separating in the observed spectra the phenomena with a distinct nature, because frequently different processes in the investigated samples can be attributed to certain elements in an equivalent circuit of the sample [1, 9]. Thus, fitting of the equivalent scheme parameters in order to match its impedance spectrum

to the experimentally obtained spectrum provides a promising and powerful tool for the investigation of complex systems, including biological objects [1, 12–15], chemical sensors [16–19], and semiconductor devices [19–21].

For implementation of this EIS processing type, there are three common approaches: algebraic, geometric, and complex non-linear least squares (CNLS) [1, 10, 11, 22–24]. The algebraic approach consists of the evaluation of EC parameters using spectrum features (minima, maxima, peak position values, *etc*). The geometric approach uses geometric manipulations with impedance spectra plots for estimation of EC parameters. The CNLS approach, introduced by J. Ross Macdonald in the 1960s [22], today is the most common and accurate method for EC reconstruction, because it can be applied to a wide range of EC topologies and because it uses the whole measured spectrum points in contrast to algebraic and geometric methods. Due to these reasons, the NELM software is basically developed for producing CNLS approximations of the experimental immittance spectra.

The key idea of the CNLS method, which we illustrate in the case of admittance representation, is to minimization of the functional

$$E^2 = \frac{1}{L} \sum_{\ell=1}^L |Y_m(\omega_\ell) - Y_{ex}(\omega_\ell)|^2 \times \mathcal{W}(\omega_\ell) = \min, \quad (1)$$

where $Y_{ex}(\omega_\ell)$ is experimentally obtained admittance at the frequency ω_ℓ , $Y_m(\omega_\ell)$ is the EC model admittance at the same frequency, L is the total spectrum measured points, and E^2 is the mean squared error, or so-called working function in the terms of optimization theory, $\mathcal{W}(\omega_\ell) \geq 0$ is the weight of the data-point at ω_ℓ frequency. Instead, the admittance Y in Eq. (1) can be used with any immittance value, such as impedance, impedance magnitude, *etc* (see Sec. 4.1). Because typically the EC impedance depends on the ECs' parameters in the indirect, non-linear manner, for solving Eq. (1) was used the numerical optimization methods, the most popular of which are gradient descent, Gauss-Newton, Levenberg-Marquardt, and Nelder-Mead [25]. The earliest known wide-spread software for providing CNLS impedance analysis is the famous LEVM program by J. Ross Macdonald, which is often used today, despite its limitations and the fact that it is technically outdated [26].

One of the commonly used today commercial software for EIS is EC-lab by BioLogic (France) [27], which is usually connected to a BioLogic device, allowing users to analyze data in the same program that they used to gather it. This program version 11.50 has a free demo shareware, and it is possible to transfer users' data from outside devices with a function import from ASCII format. This package also operates with a user-friendly interface that constructs equivalent circuits from different elements, but their selection is limited to parallel/series connections, *e.i.* bridge schemes not

supported by EC-Lab. In addition, the schemes in EC-Lab are constructed using lumped elements, and thus not all immittance phenomena can be taken into account during CNLS processing in this package. To be specific, *e.g.* it is impossible in EC-Lab add to immittance delay exponent multiplier $e^{-i\omega\tau}$, which appear, in particular, due to the time offset between current and voltage measurements using multichannel ADC or when a non-ideal cable is used [28]. In EC-Lab, users can choose a method of approximation from the five provided, regulate the number of iterations, and even use weights, but only as $\mathcal{W} = |Z|^{-1}$. We also found that EC-Lab has performed perfect approximation of model data of RLC circuit, but did not manage to succeed in the case of experimental data, possibly due to the noise and inability to add the mentioned above delay exponent to the model.

Another example of frequently used software, which deals with CNLS, is commercially available ZView by Scribner Associates Inc. ZView version 4.0 is partly based on LEVM and has a relatively modern, user-friendly graphical interface, and also provides a wide range of EC choosing [29]. It also includes an instant fit feature, which is likely based on the EC algebraic search method. However, since ZView is provided as a Windows executable file with closed source code, it has lower task flexibility compared to LEVM, source code of which can be modified for ongoing impedance research. In addition, unlike LEVM, ZView 4.0 supports batch processing of spectra, which, however, is implemented just in a pipelined manner, *e.g.* no refinements are used to initialize the CNLS for time series impedance data (cmp. with Sec. 4.2 and [30]). Unfortunately, ZView has a limitation in visualizing impedance spectra at high resolution (spectrum plots only display 108 data points, possibly due to optimization of video outputs), which makes preliminary analysis of spectra difficult. In contrast, LEVM does not support large, high-resolution spectra. Finally, EC-Lab, LEVM, and ZView, by default, are not compatible with parallel processing and modern high-speed impedance spectroscopy approaches such as adaptive filtering EIS and Fourier-EIS [1].

In recent years, several custom [31] and open-source programs for processing impedance spectra [32] have also appeared. In particular, the PYTHON packages are very popular today [32–44], however, they, like other open-source products in one combination or another have the same limitations as EC-Lab, Zview and LEVM, *e.g.* primary parallel/series EC type concept (usage the Circuit Description Code [45]), usage of only built-in optimization methods, like `lmfit` PYTHON package, only impedance (Z) fitting is released, no hardware acceleration by default, lack of time-domain EIS support, labor-intensive installation and use, *etc.*

Trying to eliminate the limitations of existing state-of-the-art programs for impedance phenomena analysis and inspired by the LEVM program, in this study, we have de-

veloped the open-source MATLAB/GNU OCTAVE-BASED software called NELM, which includes the advantages of the LEVM, EC-Lab, and ZView, and, in addition, provides rich and flexible opportunities for challenging CNLS-problems mentioned above. The alpha version of NELM (NELder-Mead acronym) was initially introduced in 2017 [12] as a request-available software, which our team has used for bioimpedance data analysis [1, 12, 46]. In this study, we have significantly improved and optimized its source code, which can be download via link [47] at GitHub. We have aimed to make an open-source software package with a clear user-friendly structure that allows people with minimal programming experience to be able to produce CNLS analysis with different equivalent schemes, minimization protocols, and miscellaneous data processing precisely for their needs. The purpose of the current article, which can be viewed as a manual, is to introduce our program to new potential users, guide them through their use of NELM and give some possible application of this package including analysis of the passive circuits, semiconductor devices, electrochemistry, and biosensing.

This paper is organized as follows. First, there is brief information about the CNLS approximation and the theory behind the minimization strategies. After that, there is an overview of the structure of the program and a short manual for users. Following that we introduce results of the different NELM performed approximations for various experiments, such as the RLC circuit identification, HeLa cell research using a multielectrode array, photodiode characterization, and EIS spectra noise canceling example. Finally, some outcomes and further development of NELM are presented and discussed.

2. NELM structure, “phrase-book”, and “guidebook”

Since most of the article shows interconnections between the NELM source code and EIS theory and experimental techniques, here we introduce a small vocabulary, provided in order to explain the meaning of NELM essential variables and routines. We have used in the text the **verbatim** font to indicate the programming variables of the source code file names.

In total, NELM consists of five parts: three interfaces for data manipulation and NELM tweaking, the core for data analysis, and the interface for result storage (Fig. 1). The key NELM variables are listed on the Tab. 1. Several NELM adjustments, which were created to give user an ability to customize (sound notification or batch processing mode), can be triggered by changing values of the variables, which are highlighted at the beginning of the main `NELM.m` file by

```
%---->>>>
```

Variable or routine	Description
<code>NELM.m</code>	The main script, by running which user starts process of approximation.
<code>NELM_Func.m</code>	The main routine, which carries out the CNLS approximation.
<code>options</code>	The structure which controls NELM core execution.
<code>N</code>	Total number of the Monte-Carlo iterations.
<code>Get_Spectrum_Func</code>	The data loading interface anonymous function.
<code>Channels</code>	The structure for storage NELM output.

Table 1: Description of the essential NELM variables and routines.

comment. These tweaks are devoted for user comfort, and are not critical for main data analysis. Contrary, the tweaks which are essential for NELM signal processing (model type, minimization algorithm, *etc.*) are stored in special m-files, which has file-name mask `Settings_*.m`. The important tweaks in these files are denoted by fish-like comment

`%>---]}`

The loading of the settings file is provided via a special graphical interface. We have used such separation of the tweaks to provide NELM flexibility, since the first of them allows to convert immittance data from arbitrary format to the format understandable for NELM, while the second part of tweaks are used to control the NELM CNLS data-processing. The settings essential for CNLS processing such as minimization methods, frequency and time range, *etc.* are stored in the `options` structure. Since NELM was initially developed for analyzing high-resolution multichannel EIS data, we introduce the structure `Channels`, which contains the information about NELM processing results in the standardized form. To provide the flexibility for the fitting model choice NELM program uses the anonymous function `Channels.Model` as an interface. It allows the user to specify the arbitrary model types in the format understandable by NELM. Thus, NELM operator can use the different processing techniques for different data types.

In summary, the NELM operates as follows: the input data is converted into the NELM-compatible format by `@Get_Spectrum_Func` interface, then the data is processed by `NELM_Func` core in accordance with `Channels.Model` and `options` interfaces, and then the NELM output is stored into `Channels` structure.

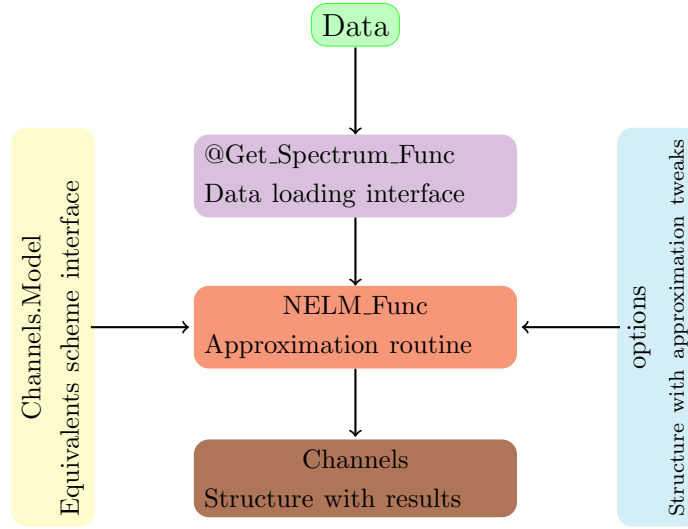
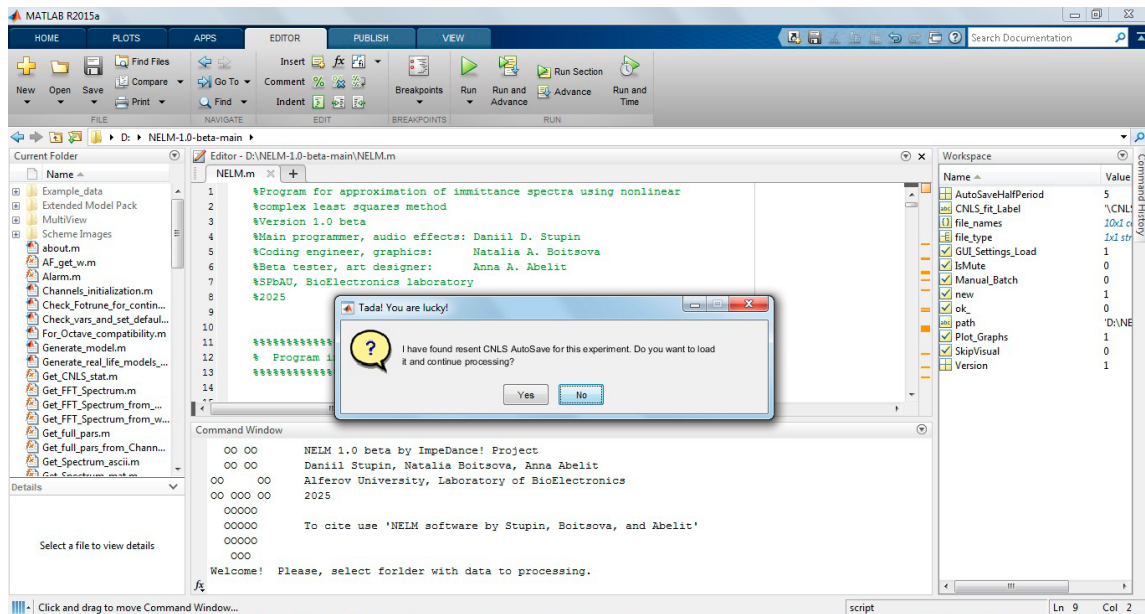
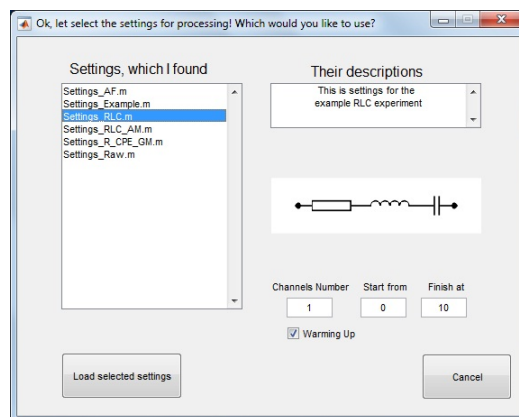


Figure 1: Architecture of the NELM engine. To provide the program flexibility NELM has: data loading interface, which convert the data in arbitrary format to the MATLAB complex column-array by anonymous function `@Get_Spectrum_Func`; model loading interface, which evaluate by anonymous function `Channels.Model` the EC immittance; the structure `options`, which define the approximations process settings; the main universal approximation core `NELM_Func`; the mentioned above structure `Channels`, which store approximation results, including EC parameters, model options, model type, data file names, *etc.*

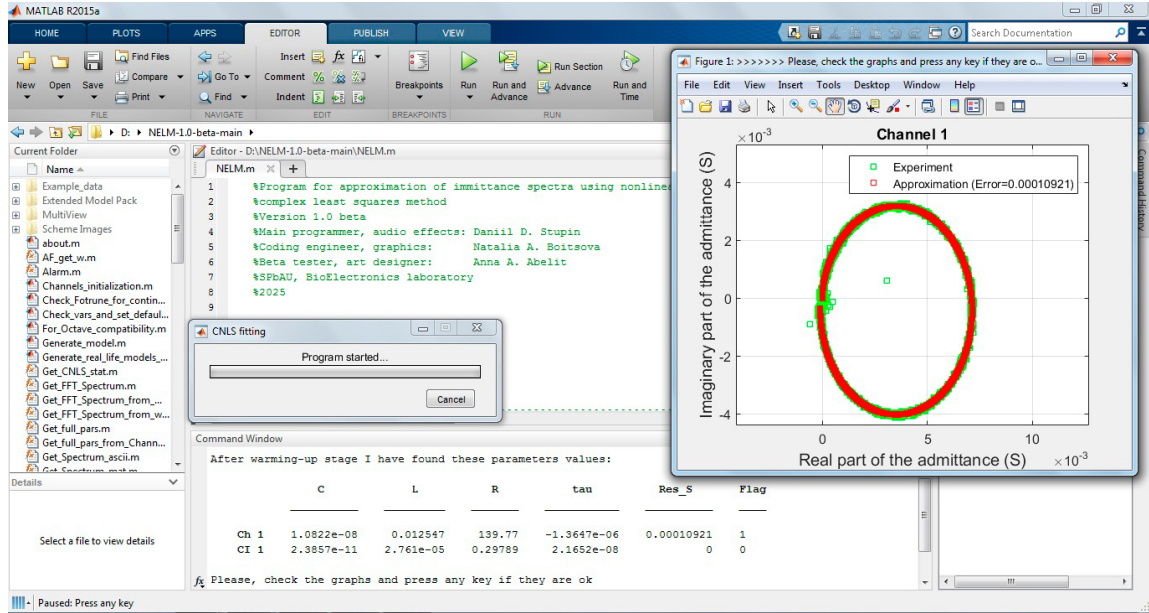


(a)

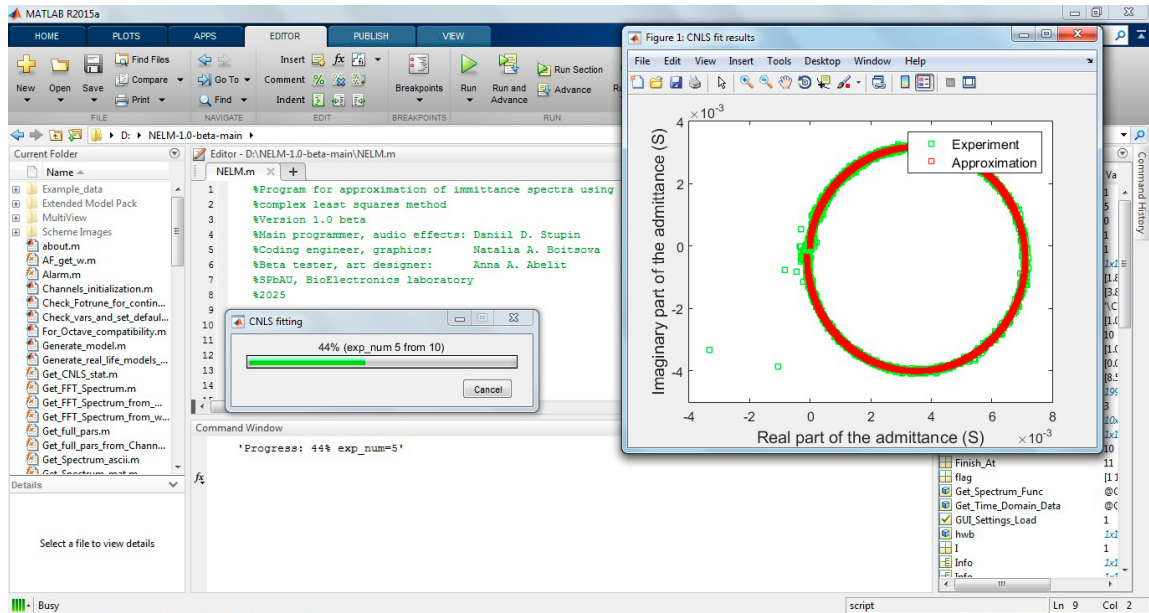


(b)

Figure 3: Screenshots of NELM software. (a) MATLAB editor window in which the NELM shell code is open, and a message prompting to continue the previously started CNLS processing of spectra; (b) window for selecting an equivalent circuit for data processing.



(a)



(b)

Figure 4: Screenshots of NELM software (continued). (a) MATLAB editor window after the warm-up stage. The screenshot shows a graph with the approximation results and a table with the EC elements ratings. Here CI are confidence intervals; (b) the result of CNLS processing of one of the RLC-admittance spectra and a window displaying the number of already processed spectra.

3. Step-by-step guide and program block-scheme

In this section we will present step-by-step instructions for NELM usage and simultaneously discuss NELM code. For this purpose we will describe each step in the text and refer to the NELM block-diagram Fig. 2 and NELM screenshots Fig. 3 and Fig. 4 to show how our package operates. So, to process impedance data with NELM the following workflow should be done (for video lessons see playlist on YouTube [48]).

1. At first, before running main script `NELM.m` user can choose the input batch processing type, data format, and initial path, where experimental data is stored. This can be done by using variables

`Manual_Batch, names, file_type`

respectively in the script beginning. In addition, since impedance data, especially in time representation, may take a large storage space, the user can connect to external network (cloud) where data are stored, by using, for example, VPN services. We have tested our software with Radmin VPN v1.4.4642.1 for Windows OS (Famatech, Tortola).

2. Then, after preliminary NELM configuration is done, the user should choose the folder with data of interest if batch mode is selected (`Manual_Batch=false`), or to choose data files manually otherwise.
3. Further, the NELM will show the graphical dialog window where settings can be selected for data processing [Fig. 3(b)].
4. After selecting the settings, the NELM starts to process the data (Fig. 2). If the warming-up mode is turned on (`Is_Warming_Up=true`) in the settings file, the NELM will try to find the best initial points for the CNLS analysis by using an increased number of iterations during solving Eq. (1), after which it will show to the user result in the form of table in the console and in the form of comparative plot in the figure windows [Fig. 4(a)], and will ask the user to check them. If obtained EC spectra match to the experimental data with low residual, and EC parameters are physically correct, then user can allow to NELM to continue processing by pressing any key.
5. During the data analysis, the NELM will show to user progress in console, and in window with progress bar, which also contain the “Cancel” button, by pressing which the emergence NELM shutdown can be made [Fig. 4(b)].
6. When all data is processed, the NELM alerts user about this event by sound alarm and by showing the message in console.

7. Finally, the NELM will save obtained data in the mat-file at the path with name `[path CNLS_fit_Label]`, where `path` is a directory with processed files and `CNLS_fit_Label` is a name for an output m-file. If string `CNLS_fit_Label` started with `\` then the output will be saved in the directory with experimental data, otherwise the output m-file will be saved in the parent directory with name, which includes the name of the folder with experimental data.

It should be noted that if some errors during CNLS treatment are occurring, the NELM does not crash, but alerts the user about them and starts processing again with the last used data file. This protection is done to prevent the NELM crashes due to network issues, for example, VPN shutting down. If such an error occurs, after network is repaired, the NELM will continue processing. If other types of error, the user can try to fix them, or they can stop NELM by pressing “cancel” button. By doing that, they require NELM to save workspace in its own folder and show more information about the problem in `Message` variable. For resuming CNLS processing from the last successful data sample after NELM emergence stop user can set the variable `new=false` at beginning of the `NELM.m` script.

After acquiring the starter pack, to make sure that the program was installed correctly, we recommend having a test run of the program with the data and settings provided for RLC circuit (use the pre-made `Settings_RLC.m` and data in folder `\Example_data\ascii\RLC 130 ohm 15 mH 10 nF\`, see video lesson [48]).

4. Overview of the NELM’s immittance analysis techniques

As discussed in the introduction, for impedance spectra analysis typically equivalent circuits concept is applied, and a goal of experimental impedance data analysis is estimation of the parameters of such equivalent circuits, which can be done, in particular, using CNLS approach [Eq. (1)]. For this purpose, in common cases numerical minimization algorithms are used [25]. This section is devoted to an overview of the strategies that were implemented in the NELM impedance spectra analysis, which we summarized for user convenience in Tab. 2.

4.1. State-of-the-art CNLS processing

In the NELM package, there are a few different numerical CNLS-strategies from which users can choose. The optimization solvers based on Nelder-Mead, Trust Region and Levenberg-Marquardt algorithms were already implemented in MATLAB, but some others, such as damped Gauss-Newton, gradient-descent search and coordinate descent search (and their combinations) were developed specifically for the

Method	Description
'Nelder-Mead'	Classical Nelder-Mead search [8, 25, 49].
'Levenberg-Marquardt'	Levenberg-Marquardt minimization [8, 25].
'Trust_Region'	Trust region algorithm [50].
'NMLM'	Combination of the Nelder-Mead algorithm with Levenberg-Marquardt method refinement.
'NMCD'	Coordinate descent algorithm, which used one-dimensional Nelder-Mead method for each coordinate [8, 25].
'NMSD'	Nelder-Mead algorithm with gradient descent method refinement.
'NM_Shake'	Nelder-Mead algorithm with local minimum jumping out feature (similar to [51]).
'NMGN'	Nelder-Mead algorithm with damped Gauss-Newton method refinement [8].
'Y_Homotopy'	Spectrum homotopy method for trying speeding up calculations.
'AF'	Adaptive filtering immittance approximation for noise canceling [1, 12].
'AM'	Algebraic method for calculation EC parameters [1, 12, 23].
'GM'	Geometric method for calculation EC parameters (compare with [24]).
'Raw'	Non-treated spectrum output (<i>e.g.</i> for Cell-Index usage) [1, 52, 53].

Table 2: Basic methods for treating immittance spectra that are implemented in the NELM package. In addition, the NELM package include a simple Kramres-Kronig test (`KK_score.m`) and distribution of relaxation times estimator (`Simple_DRT.m`).

NELM package. Since these methods were built from the ground up, a brief explanation of their implementation promises to be useful to NELM users.

Let us start with the damped Gauss-Newton iterative method [8, 25], which can be chosen in NELM by setting field `options.method='NMGN'` in setting file. It is based on the replacement of model immittance Y_m in the working function Eq. (1) by its linear Taylor series approximation, which converts problem Eq. (1) into solution of the normal equations system [54, 55]. The iterations for finding the parameter vector in this case look like

$$\vec{p}^{(n+1)} = \vec{p}^{(n)} - \lambda \Re \left(J^+ J \right)^{-1} \Re \left(J^+ \vec{F}[\vec{p}^{(n)}] \right), \quad (2)$$

where $J = \nabla_{\vec{p}} Y_m$ – Jacobian of the Gauss-Newton system,¹ n – iteration number, $\vec{F}_k = Y_m(\omega_k, \vec{p}^{(n)}) - Y_{ex}(\omega_k)$ – the residual between model and experiment [compare with Eq. (1)], $^+$ denotes hermitian conjugation, λ is so-called damping factor.

Its use allows us to expand the capabilities of the classical Gauss-Newton method, since varying the parameter λ allows the algorithm to work correctly not only near the minimum point. To be specific, in our version of the damped Gauss-Newton algorithm at each new iteration $\vec{p}^{(n)}$ uses the bisection algorithm to choose the appropriate value for λ in the range, *e.g.* $[0, 1]$, which gives the lowest value to the working function in the direction defined by Eq. (2). Such a strategy not only includes the classical Gauss-Newton $\lambda = 1$ value, but in the case if classical $\lambda = 1$ leads to the growth of the working function it allows us to make a shot at changing the working function value to something that is lower than it's value at $\vec{p}^{(n)}$. We found the application of this damped Gauss-Newton technique is very useful for refining the results of the Nelder-Mead solver, thus in the NELM package these algorithms preferred to be used in pairs, for which additionally `options.Use_NM` flag should be set to `true`. In other case, the NELM will solve Eq. (1) by using the damped Gauss-Newton method only.

It is fruitful to note that during the execution of the damped Gauss-Newton method the convergence λ to unity can be used as an indication whether the algorithm has reached the minimum point. Moreover, the iterative process almost always converges, since at each step, the algorithm moves to the lower value of the working function.

Let us now consider the gradient descent method. Contrary to the Gauss-Newton approach, this iteration technique solves equation $\nabla E^2 = 0$ without using approximation for model immittance Y_m . The idea for iterations design in gradient descent

¹In fact, this Jacobian J is a matrix, which columns corresponds to Y_m gradient components, calculated at different frequencies.

approach is based on the false-position rule, namely

$$\vec{p}^{(n+1)} = \vec{p}^{(n)} - \lambda \Re J^+ \vec{F}[\vec{p}^{(n)}], \quad (3)$$

where λ is a step, which mathematical meaning is similar to the Gauss-Newton damping factor, however, its choosing strategy is quite different. Contrary to the Gauss-Newton method, the steepest descent approach does not have any typical value for λ like unity, and the effective change in the $\vec{p}^{(n)}$ strongly depends on the magnitude of J . Thus, we have used some heuristic technique, which is based on the idea that the maximum relative change of the elements of $\vec{p}^{(n)}$ should not be greater than 1%. This strategy provides the convergence of the steepest descent method by blocking far away jumping from the $\vec{p}^{(n)}$ point, which we assume is located in relative proximity to the minimum point.

Let's now take a look at the coordinate descent method, which in fact is the sequential iterative application of the one-dimension minimization algorithms for each EC parameter. In NELM Nelder-Mead algorithm is used as a one-dimension solver, *i.e.* minimization of the working function Eq. (1) are provided via automatical fixing values of all EC parameters except one (compare with Sec. 4.4.2), for which the Nelder-Mead approach is applied. At the next iteration, the next variable is passed to the solver, while other parameters are freezed and so on. Such approach can be used when Jacobian matrix is ill-posed.

Finally, it should be emphasized that NELM allows any immittance representation to be used for CNLS fitting, since the admittance terms Y in the Eq. (1) can be replaced in the NELM code by an arbitrary immittance values via setting `options.data_type` field (Tab. 3) in the `Settings_*.m` file. Also, weight type \mathcal{W} in Eq. (1) can be selected by setting `W_type` variable (`unit`, `inverse_moduluse` weights are already implemented in NELM, as is the `custom` weight, which can be *e.g.* loaded from a file).

<code>options.data_type</code>	Immittance used in Eq. (1) instead Y	Domain
<code>'Y'</code>	Admittance (default)	Complex
<code>'Z'</code>	Impedance	Complex
<code>'abs_Y'</code>	Admittance magnitude	Real
<code>'abs_Z'</code>	Impedance magnitude	Real
<code>'real_Y'</code> or <code>'imag_Y'</code>	Real or imaginary part of the admittance	Real
<code>'real_Z'</code> or <code>'imag_Z'</code>	Real or imaginary part of the impedance	Real
<code>'angle_Y'</code>	Phase of the admittance (experimental)	Real

Table 3: Immittance types available in NELM for CNLS processing. Users can add their own types (*e.g.* $\log_{10}|Y|$) to the `Residual_LMS.m` script.

4.2. Monte-Carlo CNLS stabilization

Since CNLS is a non-linear problem, local (false) minima points of the working function can exist. To our best knowledge, today there is no universal algorithms for finding global minimum developed, but some heuristic approaches still can be utilized for overcoming local minimum issues. One such promising technique can be realized by using Monte-Carlo concept, *e.g.* by probing working function via starting minimization algorithms from different initial points with further choosing the result with minimal residual, so called Multistart approach [56]. The mentioned above starting points for other factors being equal can be chosen randomly in some reasonable area in the EC parameters space. This strategy is implemented in NELM by utilization of the `parfor` loop type for increasing the speed of the CNLS calculations. Moreover, to stabilize the operation of the NELM software for each spectrum in batch processing of the time-depend, non-stationary immittance data samples, we are not only using the best result of the previous approximation as a starting point for the next one, but also a certain set of points, obtained randomly in a small vicinity of mentioned above starting points (local MC mode) or near appropriate “universal” starting point (global MC mode), which does not change during approximation process. The random selection of these points was carried out using the `rand` or `randn` functions of the MATLAB software, and their number in total was usually equal to the number of processors on the computer, which made possible for us to speed up the Monte-Carlo calculation process due to parallel processing of data carried out using the `parfor` loop. In this case, for each spectrum, the best result of the Monte-Carlo experiment (with the smallest error), as well as the average value of the parameters of the equivalent circuit and their confidence intervals obtained with its help, were stored in the computer memory in the `Best`, `Mean`, and `CI` fields of the `Channels` structure, respectively. This approach allows us to discard the results of the CNLS approximation with a large error and thereby solve the above-described

problem of destabilization of the approximation process due to the appearance of unsuccessful CNLS fits. The obtained statistical data can be thought of as a computationally inexpensive measure of the working function surface smoothness, which can be used for measurements quality control. It should be noted, that we does not use build into MATLAB method `MultiStart` since NELM was developed as multipurpose investigating CNLS platform and thus we try to make code flexible for users challenging problems, for example for statistical analysis of the working function mentioned above. The on/off switching of MC mode is provided via setting the value of the `N` variable in the `Settings_*.m` file, which is equal to the total number of the MC iterations: if `N=1`, then MC mode is switched off, otherwise the MC mode is turned on. The type of the MC mode (local or global) is achieved by setting field `options.Monte_Carlo_Mode` to `'local'` or `'global'`, respectively.

Another Monte Carlo application for local minimum fighting could be done by addition to a solver's result \vec{p} by some random vector \vec{r} and by starting solver again using $\vec{p} + \vec{r}$ as initial point with the hope of jumping out from the possible local minimum. This strategy is implemented in `NM_Shake` approach in NELM.

NELM also supports estimation of confidence intervals for CNLS fitting that is not based on Monte-Carlo trials, but on the Jacobian usage (Sec. 4.1) instead. Namely, after the fitting procedure, NELM provides a working function (1) Taylor linearization and applies the standard Student's approach [57, 58] to the resulting system of linear equations to obtain confidence intervals, which are stored in `Channels.Best_CI` field.

4.3. Adaptive filtering approach

The NELM software development was motivated by the absence of immittance analysis programs, which were able to work with high resolution spectra, generated mainly by FFT impedance spectroscopy [59–61]. Unfortunately, the FFT approach is not robust to noise and inferences contrary to the AI-based adaptive filtering approach, which also produces high-resolution spectra [1, 12, 62]. In order to give to users the possibility to work with noisy data, we have also integrate into the NELM the adaptive filtering processing. Using a slight modification of the convolution theorem [63] we also increase the AF-based approach speed, making it comparable to the FFT approach. To call the adaptive filtering method, the user should set in the field `options.method='AF'`, and declare the filtering order in the fields `options.ell_n` (for voltage) and `options.ell_d` (for current).

4.4. User-defined CNLS solvers and models

4.4.1. Adding new methods of minimization to NELM

One of the key features of the NELM package is the possibility to add and modify data processing protocols. Specifically, to create a new immittance analyzing method, the user can create in `NELM_Func.m` subroutine the new case in the `switch options.method` environment, and then select it for data processing by changing `options.method` field the `Settings_*.m` file. The data processing code can be arbitrary, but should return the immittance treatment results in the `Result` row-array. Also, users' data processing code should return in the `Time` variable the time, when experimental data were captured, in the `info` structure it should return the information about data analysis performance (fields `iterations`, `funcCount`, `algorithm`, and `message`), in the `flag` user's code should return the machine understandable numeric pseudonym for success or unsuccessfulness of the data processing treatment, and also user's code should return `Misc` structure with additional information, *e.g.* it can contain the level of the offset voltage, temperature, *etc.* The values of the `flag` and `Time` variables and `info` and `Misc` fields are not essential for NELM operating, and in the simplest case their values can be set to zeros or empty strings as some kind of stub (compare with case 'AF' in the `NELM_Func.m`). In the new case for data processing, the user can load the experimental data by using anonymous function `Get_Spectrum_Func`, as well as they can access the model for data fitting by calling anonymous function `Model` (which is a copy of `Channels.Model` in the `NELM_func` environment). Both of these anonymous routines should be specified in the `Setting_*.m` file. For CNLS processing in the NELM there exists routine `Residual_LMS`, which calculates the least mean squares (LMS) error between a experimental and modeling data. This routine via wrapper, typically denoted in NELM as `Target`, can be sent to any LMS solver, as it is done in the current NELM version for 'Nelder-Mead' or 'Levenberg-Marquardt' methods. All already existing cases have comments and can be used as guidance in order to create new methods of approximation.

4.4.2. Model interface description

The NELM package version 1.0 beta already contains mostly used in physics, chemistry, and biology EC, which are listed in Tab. 5. For software flexibility, each model in NELM is represented as a distinct overload m-function file, with mask `M_*.m`, the output mode of which is controlled by its input format. For obtaining as output the model admittance values at specified frequencies the model m-function should be called with three arguments, namely the frequencies vector, at which immittance should be calculated, the parameters of the EC, which can be

tuned for fitting process, and `Model_Options` structure, which controls the immittance evaluation process. In this case, of three arguments input mode if flag-field `Model_Options.get_J` is equal to true, then the model m-function will return two values: the admittance and it's exact Jacobian calculated at frequencies input vector. Here we emphasize that by default the model functions should return the admittance and its Jacobian. For CNLS processing, Jacobian calculations, and plotting with other immittance types, the user can select the required type from Tab. 3 using `options.data_type` variable in `Settings_*.m` file. Along with the obvious application of the Jacobian evaluation feature for the minimization of the working function Eq. (1), its purpose also includes the analysis of the conditionality of the CNLS problem and the search for redundant parameters of the equivalent circuit. If the model m-function is called with no arguments or with one argument `Model_Options` it returns the information about the model in the form of structure, the common fields of which are listed in Tab. 4. The last output mode is very essential for NELM execution, since model information is frequently used in the main script, *e.g.* for fitted parameters parsing.

The immittance evaluation inside the model m-function can be performed using all MATLAB computation power, including pipeline calculations (`.*`, `./`, `.^` notations). In this context, it should be stressed that NELM is not limited to parallel and series connection of the circuits, but is also able to work with bridge-like circuits, which is known for its inability to be represented by these types of connection. The simplest example of such circuit types was added to the NELM package as the `M_Bridge_r1.m` model (see Tab. 5).

Moreover, the number of the EC presented in Tab. 5 is much higher than its row number, since we also have taken on board the EC constructing technique from the original LEVM program. More specifically, the user can fix certain EC parameters at preferred values by using `Model_Options.fix_pars` 2D-array field, therefore, they will create a new EC. To achieve this, the user should in the first line of the `fix_pars` array specify the indexes of the parameters that should be fixed, and in the second line they should type the corresponding parameter values. For example, option

```
Model_Options.fix_pars=[1, 7; 8 11]
```

will fix the EC first parameter value at 8, and the seventh EC parameter value at 11. Also for converting some elements into the break-circuit or short-circuit the user can set their impedance to infinite or zero, respectively, by choosing appropriate elements parameters values. For example, for creating parallel RC-circuit EC, the user can take as the basis the `M_Fricke_Morse_r1.m` model, and set its CPE exponent value to 1, and set its series resistor rating to zero. It should be stressed that after

Field	Description
Elements_names	The cells-array, containing the names of the EC parameters.
Pars_N	Total number of the EC parameters.
Elements_Num	Number of the non-fixed EC parameters.
non_fix_pars_idx	The indexes of the non-fixed parameters.
non_negative_pars	The indexes of the non-negative parameters with respect to the non-fixed parameters order.

Table 4: The fields of the information structure, which return NELM impedance-model routines, when they are called with one argument. These info-structures are essential for NELM functioning.

parameters values are fixed in the input initial parameters vector **x0_init** the values that correspond to fixed parameters should be deleted.

Along with the LEVM concept for constructing an EC, the user can define their own EC by using the **Make_Model** routine. In order to use it, users only need to pass into this function admittance of the new model as symbolic expression (**syms** type in MATLAB), name of the variable, which corresponds to frequency (in format $i\omega$, *e.i.* it must be a single symbolic expression denoting the product of the imaginary unit and the angular frequency), file name of EC model m-function, and names of variables that can become negative. For example, to define EC for the single CPE element model, the user should calculate its admittance manually (*e.i.* $Y=W*(i\omega)^\alpha$) and call function

Make_Model(Y, iw, 'CPE', alpha,0)

with additional parameters **iw** (frequency name), **'CPE'** (function file name) and **alpha** (parameters that can drop below zero), after that MATLAB will automatically create CPE.m function file, containing both EC admittance calculation part and Jacobian calculation part implemented in m-code. Here we want to stress that **Make_Model** function for Jacobian evaluation uses MATLAB low-level code, which is obtained by converting the Jacobian symbolic expression to algebraic MATLAB pipeline routines. This allows the user to create Jacobian automatically and significantly increase the speed of Jacobian calculations.

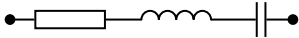

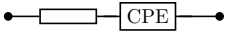
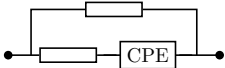

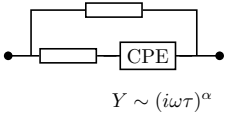
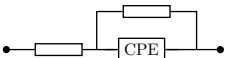
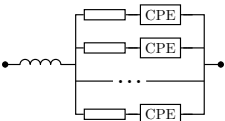
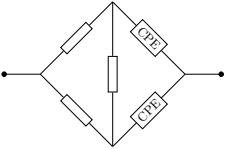
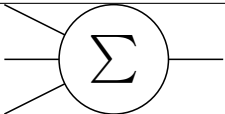
m-file	Circuit	Description	Applications
M_RLC_rl.m		Simple series RLC-circuit.	Resonance systems (<i>e.g.</i> quartz resonators), capacitive systems (metal/electrolyte interfaces, pn-junctions) with parasitic inductance.
M_RL_CPE_rl.m		Series RL-CPE circuit.	Electrochemical or non-ideal interfaces with parasitic inductance.
M_R_CPE_rl.m		Series R-CPE circuit.	Diffusion, rough, non-ideal electrochemical electrodes, complex distributed systems (<i>e.g.</i> tissues).
M_Fricke_Morse_rl.m		Series R-CPE circuit with resistive leakage also known as Cole-Cole model or Fricke and Morse circuit	Tissues and cells suspensions, electrochemical systems with Faradic currents, systems with resistive parasitic leakage.
M_Randles_rl.m		Often used Randles circuit	Electrochemical interfaces.
M_Cole_Cole_FM_rl.m		Fricke and Morse topology scheme with Cole-Cole τ -representation for CPE	Measuring time-constant for tissue or electrochemical interfaces.
M_Lapicque_rl.m		Lapicque topology scheme with CPE	Electrochemical interfaces or pn-junctions with defects, and tissue studying.
M_Multiple_R_CPE_rl.m		Parallel connection of the series R-CPE circuits.	Unhomogeneous or hierarchical systems (<i>e.g.</i> embedded into electrolyte electrode with two or more surface regions with different properties).
M_Bridge_rl.m		Famous circuit, which does not reduce to parallel and series connections	Electrochemical interfaces with two different regions, between which current can flow.
M_AF.m		Artificial intelligence adaptive filtering model	Systems which are investigated under noise environment <i>e.g.</i> biosensors.

Table 5: Build-in NELM models presented in version 1.0 beta. Although it may seem that some of the models, for example, `M_Fricke_Morse_rl.m` and `M_Cole_Cole_FM_rl.m`, are nearly identical, they both were added to spare some user's time and to avoid numerical issues which appear with different Jacobian computation. The `'_rl'` (real life) suffix is used for models that include series parasitic inductance [12] (not shown in the pictures) and current-voltage delay (see Sec. 1).

5. Software design materials and methods

5.1. Hardware and software requirements

The NELM-software was created in MATLAB language using the Open-Source concept and for this reason it was provided with detailed comments. In the process of developing NELM software, we used libraries built into MATLAB, *i.e.* routines created by third party developers were not involved. Graphical interfaces were created with the support of the `guide` constructor built into MATLAB.

Primarily, the software was tested on three computers with the following characteristics: an Asus laptop with an Intel Corei3 2330M 2.2 GHz processor, 2 cores, 3 GB of RAM (Asus, Taiwan), a Huawei laptop with a 12th Gen Intel(R) Core(TM) i5-1240P processor 1.70 GHz, 12 cores, 16 GB RAM (Huawei, China), Key desktop computer with Core i5-4570 3.2 GHz processor, 4 cores, 4 GB (Key, Russia). The highest performance was achieved on a Huawei laptop.

Before starting work with NELM, users need to make sure that their computer meets all necessary requirements. To operate, NELM software requires a preinstalled application package MATLAB version not lower than 2015a or GNU OCTAVE version no lower than 9.3.0. The NELM 1.0 beta distribution requires 37 MB of free disk space (2 MB without example data). To fully use NELM, it is desirable to have a multicore processor compatible with the Parallel ToolBox package, at least 1 GB of free RAM, and access to the Internet for cloud computing.

If the user's device meets all the criteria mentioned above, they need to download the NELM from GitHub using link [47].

5.2. Structure of the program and operational principle

General steps, on which NELM software approximation is based, are presented (Fig. 2). Firstly, the user is required to select the folder with input data files if flag `Manual_Batch` is set to `true` or select manually the files of interest if flag `Manual_Batch` is set to `false`.

Secondly, the operator, using a graphical interface, selects a settings file, which contains information about the method of data processing, frequency and time ranges, the model for approximation and the initial values of parameters, parameters that can change during the approximation process, and ones that should have fixed values, *etc.*

After user-controlled initialization of the NELM software settings is completed, the computer begins batch CNLS processing of the spectra. Since the best initial approximation for the parameters of the equivalent circuit is not always precisely known, the program can optionally begin with a warming-up mode (if flag

`Is_Warming_Up=1`), the purpose of which is to increase the number of iterations by a factor `WUC` to help in more accurately finding the values of the elements of the equivalent circuit for the first measured spectrum. For the remaining spectra, the initial point of approximation can be chosen based on the assumption that during the time between measurements of neighboring spectra the properties of the system under study change insignificantly. This allows us to use the result of the approximation of the $N - 1$ spectrum as an initial approximation for the N spectrum. However, at the same time, one should not lose sight of the existence of a situation in which the approximation on the $N - 1$ spectrum turns out to be unsuccessful, since in this case, approximations for all subsequent spectra may also fail with a high probability. In order to deal with this problem the 'global' mode for Monte-Carlo multistart can be chosen (see [30] and Sec. 4.2).

After successful completion of the approximation process, the program saves the results in the form of a mat-file and notifies the operator about end of its work through the console and a sound signal.

In addition to the scripts described above, the NELM package includes a number of routines that facilitate the processing and analysis of spectra. In particular, such utilities include the `MultiViewGUI` script, which allows the user to visualize both the experimentally measured spectra themselves and their CNLS approximation, script `KK_score.m` for performing the Kramers-Kronig test [64, 65], and a `simplde_DRT.m` – implementation of the transformation of immittance into a relaxation time distribution [66].

6. NELM application for real-life systems research

One of the important steps for any software in the final stage of development is to check its performance on the several test input examples, for which the supposed outcome is known. Thus, this section is devoted to demonstration and verification of NELM performance and efficiency to solve CNLS problems in various scientific areas. We have started such a demonstration with simple RLC-circuits analysis, further we will show the applicability of NELM for semiconductor device diagnostics, after that we will present a bright example of the NELM application for biosensing and electrochemistry, and finally we will demonstrate the built-in NELM AI-based noise reduction technique, namely the adaptive filtering impedance spectroscopy approach.

The samples, whose impedance spectra was measured by the setup, discussed in Refs. [1, 12], were used as input for NELM. The data were collected with usage of the sweep-shaped 15 mV excitation voltage with frequency band 10 Hz – 40 kHz. The sampling rate used was 500 kHz, and the spectra were measured with 2 Hz

resolution. For calculating high-resolution spectra, the FFT and AF approaches were used [1, 12, 59–61]. For data analysis, we have used the admittance representation for immittance, and no weights function was used, *i.e.* the Eq. (1) with $\mathcal{W} = 1$ was solved for CNLS processing. After collecting impedance measurement, we analyzed the obtained results with NELM using the Nelder-Mead minimalisation method and the Monte-Carlo multistart stabilization mode [56]. For all experiments, except AF-noise immunity test, as a model the RL-CPE equivalent circuit with delay exponent was used, which impedance is

$$Z = \left(R + i\omega L + \frac{1}{W(i\omega)^\alpha} \right) \times e^{i\omega\tau}, \quad (4)$$

where R is resistance, L is inductance (modeling inductor or Op-Amp), W is pseudo-capacitance, α is non-ideality parameter (CPE exponent), τ is delay between voltage and current collection in ADC. For AF-noise immunity test we have used Lapique model (Tab. 5) with additional inductor and delay exponent.

In order to test abilities of NELM on lumped schemes, we gathered impedance spectra from RLC circuits composed of inductor (Bouruns, UK) with factory ratings $L = 15$ mH and $R = 100 \, \Omega$ and capacitors with factory ratings 10, 15, and 20 nF. For demonstrating NELM applicability for semiconductor studies we have selected the silicon photodiode BPW 20 RF (Vishay, USA) and measure its capacitance-voltage response in the absence of photocurrent. Next, we have shown the power of the NELM for biosensing and electrochemistry; to be specific, we have demonstrated single HeLa cell detection using CNLS and multielectrode array 60MEA200/30iR-ITO from MultiChannels Systems (Germany). Finally, for AF noise immunity test, we have made Lapique circuit with series resistance 212 Ω , leakage resistance 992 Ω , and capacitance 48 nF. The ratios of the values of this chain were chosen similar to the ratios of the elements rating for living cell EC.

6.1. RLC circuit and semiconductor study

The obtained results for RLC-circuit and photodiode are presented in Fig. 5. From panels (a)-(b) in Fig. 5, it is easy to see that NELM correctly fits the RLC circuits spectra, which is also justified by the resonance peak height and position agreement with the quantities $1/R$ and $1/(2\pi\sqrt{LC})$ calculated from the NELM fit output [for RLC experiment CPE exponent in Eq. (4) was fixed at unity value].

The results for a more complex system, a semiconductor photodiode, are presented in panels (c-d) of Fig. 5. These graphs show a set of photodiode admittance spectra measured with a reverse bias voltage applied to it in the range from 0 to -1.8 V. Analyzing obtained data with NELM we have found that the CPE exponent

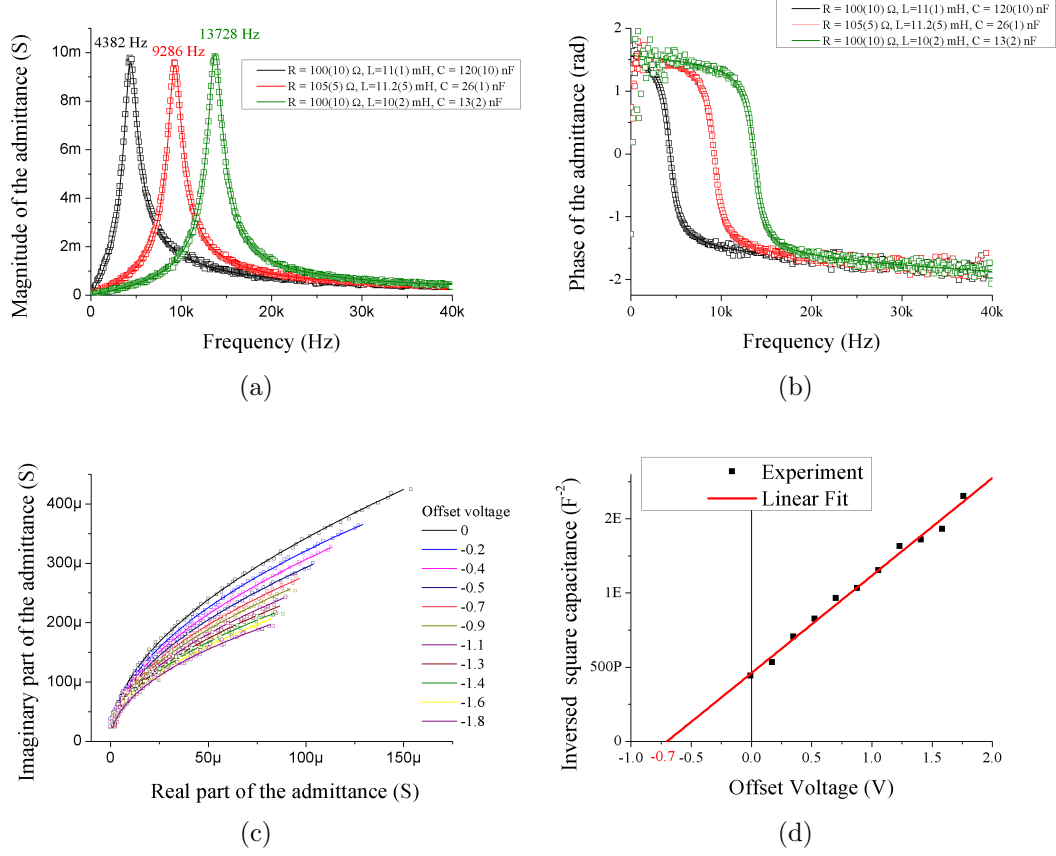


Figure 5: Kaleidoscope of the NELM applications for various scientific tasks. (a) and (b) Identification of simple passive RLC-circuits by NELM. (c) and (d) Measurement of junction barrier voltage using NELM for the BPW 20 RF photodiode. The scatter points correspond to experimental data, and the solid lines denote the CNLS approximation. For making graphs easy-to-read the experimental points were plotted using decimation with step 100 Hz. One can see that NELM can be successfully applied for element rating determination in passive circuits as well as for non-linear electronic systems study.

value for this devices is close to unity and does not significantly depend on the offset voltage. That means that the investigated pn-junctions work as ideal capacitors, and thus the considered device functioning agrees with the classical semiconductor theory [67]. Secondly, we have found that the spectra of the photodiode can be described by series RC circuit, which means that its pn-junction is near-perfect because it has no significant leakage resistance across junction, which can indicate that junction defects concentration is low. Finally, we have observed that photodiode CV-curve, calculated using CNLS, is linear, which means that its pn-junction are abrupt. Using data presented on Fig. 5(c-d) we have calculate the barrier diffusion potential value 0.7 V. The same value we have found from the digitized CV-plot for BPW 20 RF manual [68, 69], which confirms the NELM applicability for studying non-linear impedance phenomena.

In summary, we can conclude that NELM not only can be used to fit the impedance spectra of electronic devices, but its application also allows extracting their key electrical and structure characteristics.

6.2. Cells research

Contrary to the widespread CNLS software products, NELM was developed with the ability to analyze the non-stationary impedance data, which were stored as chronological set of files. This feature was introduced to NELM, especially for processing bio-impedance data, in particular for studying living cells using the electrical cell-substrate impedance sensing (ECIS) concept. The original ECIS technique consists of the impedance measuring of the microelectrode covered by cells, and conventionally for cells state determination in ECIS used raw impedance data in the format “frequency *v.s.* impedance value” [70, 71] or in the Cell Index format [52, 53], which is the normalized impedance at specific frequency. In this section, we will show the advantages of the combination of CNLS with ECIS by demonstration of the NELM application for ECIS data. To do this, we have used HeLa cells [72], which were obtained from the Institute of Cytology RAS, and for experiments they were incubated in multielectrode arrays in 5% CO₂ atmosphere at 37°C in DMEM medium (Biolot, Russia) overnight. Before experiments, the cells’ medium was replaced with a phosphate saline buffer (PBS, Biolot, Russia). The chronological impedance spectra were collected once at 500 ms in the range 20 Hz - 40 kHz (2 Hz resolution) with setup, discussed in Refs. [1, 12].

To demonstrate the advantage of NELM for cell research, we have show the experimental proving for Giæver-Keese statement above resistive action of the cell influence of the electrode/electrolyte impedance [1, 73, 74]. To do this, we captured the evolution of the microelectrode’s impedance near which the single cell detached

from the MEA dish is floating. As can be seen on the Fig. 6, NELM-approximation had successfully detected it, reacting to this process by raise and drop of the resistance, which is caused by electrical current blocking by cells membrane, as predicted by Giæver-Keese model. It is even possible to pinpoint the moment when the cell was fully above an electrode with the help of resistance *v.s* time plot and data from photographs, which correspond with each other. At the same time, the pseudo-capacitance of the electrode is stable during experiment, which proves that living cell placed in proximity to electrode acts as passive resistor. Moreover, in contrast to resistive part, which reflect the biological phenomena, obtained by NELM data include also the pseudo-capacitive part, which reflect the electrochemical structure of the microelectrode (TiN)/electrolyte(0.9% NaCl) interface. In the above experiment, we found that such a system acts as a CPE element [75, 76] with a non-ideality parameter value of 0.8, which indicates that diffusion phenomena and surface inhomogeneity effects significantly determine the mechanism of current flow through it [46, 77–81].

The examples presented in this section argue that NELM is a comprehensive tool for solving a wide variety of scientific problems, from electrical schemes testing to living matter investigation.

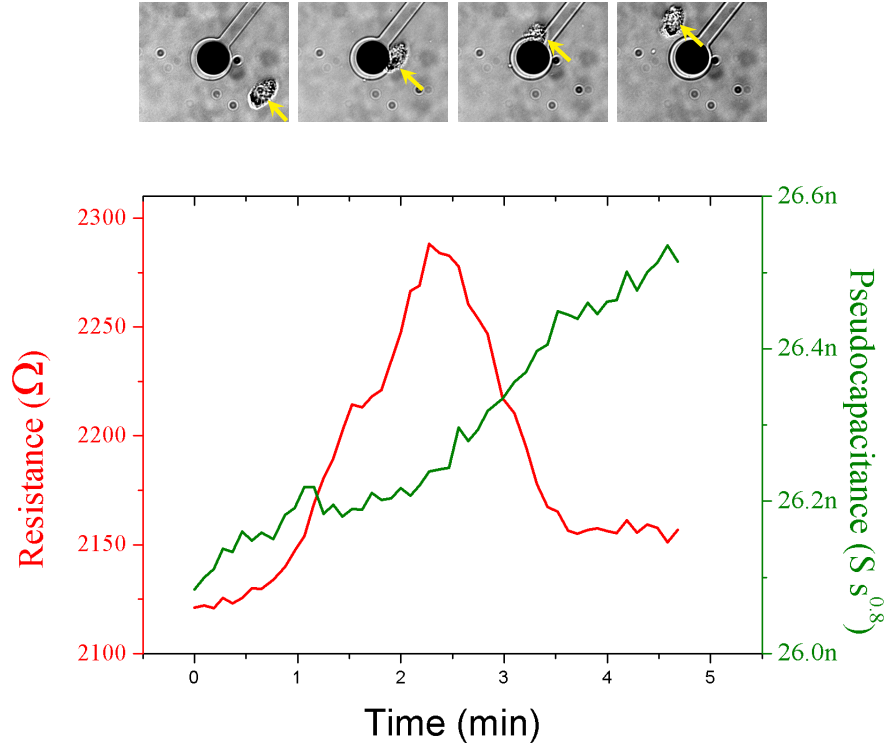


Figure 6: The demonstration of the NELM applicability for the cell biophysics research. In this experiment we have measured the impedance of the bioelectrode when the living cell is floating above it (denoted by arrows). It can be seen that obtained by CNLS electrodes resistance is increasing synchronously with cell movement, which is in accordance with Giæver-Keese model (R-CPE equivalent circuit usage for cell/electrode interface). At the same time, the electrodes pseudo-capacitance is stable to the cell's movement, which is also agree with R-CPE model for bioelectrode, because the pseudo-capacitance effects should be significant when cell interact with electrode on the distance about 8 Å (Debye length for 0.9% NaCl solution), which is lower than “electrode – floating cell” distance.

6.3. AI-based adaptive filtering approach for noise combat

Bioimpedance measurements are often performed in noisy environments because low voltage and current values are required for the safety of biological objects. One of the software methods for noise suppression is impedance spectroscopy based on adaptive filtering (AF) (see Tab. 5), which is implemented in NELM in the procedures `AF_get_w.m` and `M_AF`. The first one is used for AF training (obtaining the Wiener solution using the Gaussian normal equation method [82]), and the second one is used to calculate the AF admittance spectrum. The main pre-tuning parameter for the AF model is the number of weighting coefficients (or the order of the adaptive filter, WC), which can be set in the `Channels.Model_Options.ell_n` (voltage WC) and `Channels.Model_Options.ell_d` (current WC) fields. As input to the training process, the `AF_get_w.m` function takes as arguments the measured time-domain excitation voltage and current response sequences. The third parameter to the `AF_get_w.m` procedure is the `Model_Options` structure, which controls the training process. *E.g.*, if the `Model_Options.exact_Gram_Calc` field is `true`, then AF will exactly calculate the Gram matrix for the system of normal Gaussian equations, which for high orders of AF can be a time-consuming process. In contrast, if this field is `false`, then the Gram matrix will be calculated approximately, but this process will take no longer to evaluate than the FFT. To save the AF data in ASCII format, the user can set the `options.export_to_ascii=true` field, after which NELM will export the spectra during training to `data*_AF.dat` files in the folder where the voltage and current time domain response sequences are stored. For further CNLS processing of the AF spectra, the user can set the `I_select_file_type_number=2` variable (in the original source-code version 1.0 beta) at the beginning of the main `NELM.m` script before running NELM, which tells NELM that it should now provide CNLS processing with the AF data in ASCII format as input. In other words, to process the AF, the user must call NELM twice: once to obtain the denoised AF spectra and once to analyze these CNLS spectra. Note that the theory for AF impedance spectroscopy in Ref. [12] is based on the assumption of the infinite length of current response and excitation voltage, so to prevent the overlearning (interpolation effect) it is necessary to have sufficient excitation voltage and current response length for selected AF order.

The example of the AF noise-canceling effect is presented in Fig. 7, on which are depicted immittance spectra of the frequently used in biology Lopicque circuit (see Tab. 5), with and without external white noise numerical addition. The signal-to-noise ratio (SNR) -6 dB is used for noisy data simulation. One can see that in the noise absence the CNLS results for both AF and FFT approach are identical, however, after noise addition only AF is able to estimate the elements rating because

the FFT-estimation has error much higher than the mean rating values. Thus, the build-in the NELM AF-based approach increases the chance of calculating the correct CNLS fit even in the case where SNR is negative, which can be observed in the real-life bioisensing application. It should be stressed that calculated model spectra are similar both for AF and FFT for both high and low SNR, while CNLS rating are significantly different for FFT-approach. This “mirage” effect is very dangerous for fit-quality visual estimation. The ability of the AF-technique to detect correct elements ratings confirms the necessity of using noise-suppression methods in impedance spectroscopy.

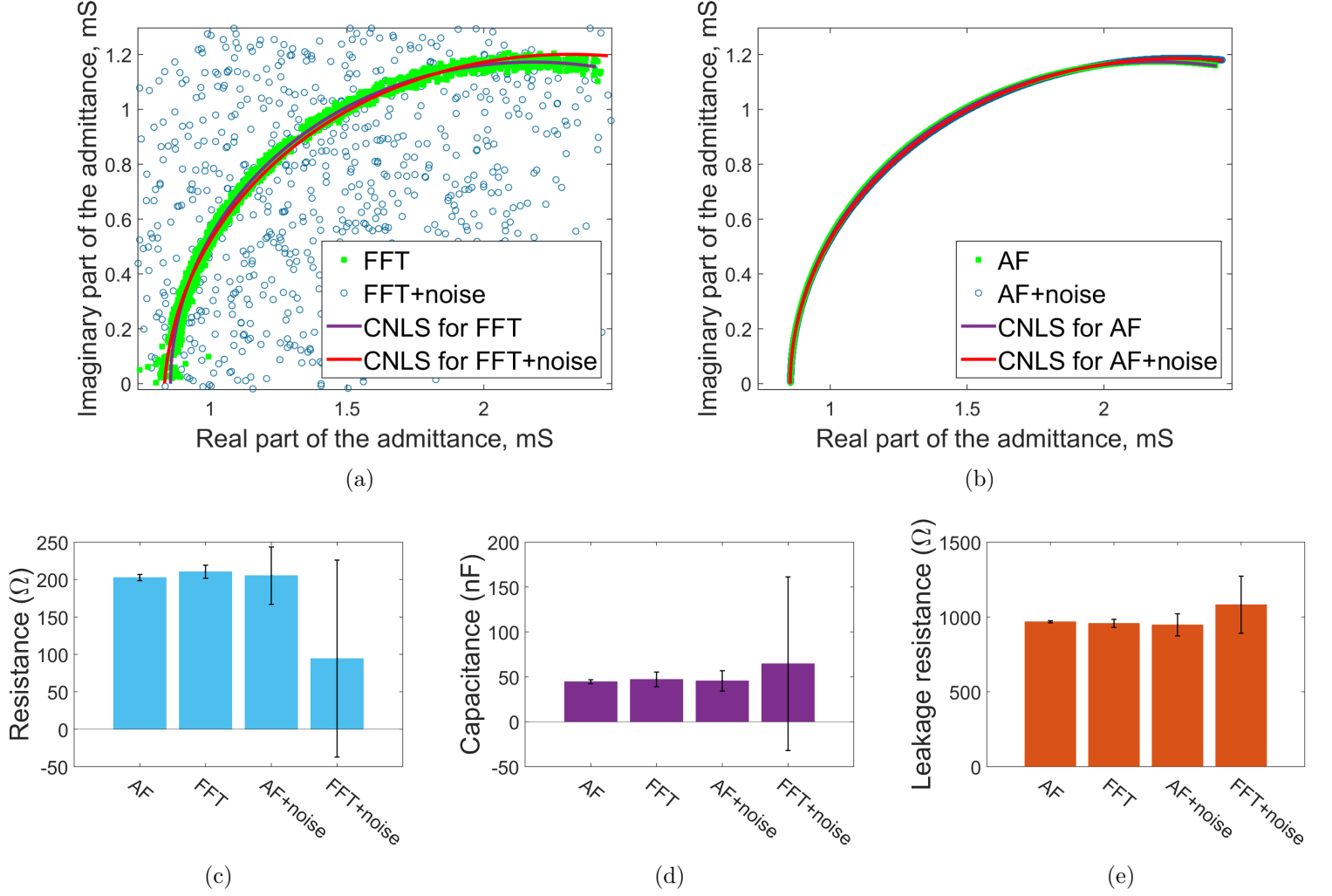


Figure 7: Demonstration of the AF impedance spectroscopy noise immunity. Here green dots are corresponds to the experimental input data with high SNR, blue dots corresponds to the spectra obtained with low SNR, red and magenta lines are corresponds to the spectra CNLS fit with Lapicque model (Tab. 5). Panel (a) show example of the FFT-approach usage, panel (b) corresponds to AF-technique. Lower panels presents the obtained Lapicque scheme elements ratings. For obtaining 99.9%-confidence intervals the statistics were collected over 10 experiments. The presented data clearly demonstrate, that AF is much stable to external random noises, which make this approach is promising for wearable biosensors or implants.

7. Conclusion and outlooks

In this study, we have developed NELM software, which allows us to perform powerful impedance spectra analysis, including the CNLS-approach, statistical treatments, noise canceling adaptive filtering, as well as raw spectra visualization.

We tried to make our software as flexible as possible, allowing the operator to be in full control of the approximation process. To be specific, the user can not only choose build-in approximation protocols, but also they can construct their own model schemes, input/output data files, and approximation types by included in NELM package set of the constructing tools. Also, we tried to sculpt NELM as user-friendly software, and for this reason in addition we tried to equip its messages with human-like quirky language with some sense of humor. Possibly, in the future we will equip it with more progressive speaking features based on AI.

Implemented in the modern MATLAB language with the usage of the open source ideology, the NELM is the next-step-generation software for impedance data analysis, which combines the modern parallel computing opportunities with well-tuned MATLAB algorithms and abilities of symbolic calculus. In addition, even though NELM was originally created in order to analyze bioimpedance, we have proved that it is capable to solve same tasks for the other fields of science, from semiconductor studies to electrochemistry. Thus, the results of our study can find an applications practical biology, experimental medicine, semiconductor technology, electronic industry, electrochemical and biosensing devices. We believe that NELM will be helpful for fundamental biology and bioelectronic research, as well as for solutions of relevant problems of healthcare and environmental studies. We will try to make the NELM updates according to the last scientific discoveries and challenges in the art of approximation.

Acknowledgement

The authors are grateful to Kraevskya A.A. for valuable programming advise, Yakovleva L.E., Dubina M.V. for comprehensive assistance and support. This study was carried out with the support of the Ministry of Science and Higher Education (Project № FSRM-2024-0001).

References

- [1] D. D. Stupin, E. A. Kuzina, A. A. Abelit, A. K. Emelyanov, D. M. Nikolaev, M. N. Ryazantsev, S. V. Koniakhin, M. V. Dubina, Bioimpedance spectroscopy: Basics and applications, ACS Biomaterials Science & Engineering 7 (6) (2021) 1962–1986.

- [2] A. Oppenheim, R. Schafer, L. Rabiner, B. Gold, B. Hunt, Digital signal processing and theory and application of digital signal processing (1976).
- [3] U. Tietze, C. Schenk, E. Gamm, Electronic Circuits: Handbook for Design and Application, 2nd Edition, Springer-Verlag Berlin Heidelberg, 2008.
- [4] V. K. I. Dimitris G. Manolakis, Applied Digital Signal Processing: Theory and Practice, 1st Edition, Cambridge University Press, 2011.
- [5] John O. Rawlings, Sastry G. Pantula, David A. Dickey, Applied Regression Analysis: A Research Tool, 2nd Edition, Springer Texts in Statistics, Springer, 1998.
- [6] Xin Yan, Xiao Gang Su, Linear Regression Analysis: Theory and Computing, 1st Edition, World Scientific Publishing Company, 2009.
- [7] Robert V. Hogg, Allen T. Craig, Introduction to mathematical statistics, 4th Edition, Macmillan, 1978.
- [8] Per Christian Hansen, Víctor Pereyra, Godela Scherer, Least Squares Data Fitting with Applications, Johns Hopkins University Press, 2013.
- [9] E. Barsoukov, J. R. Macdonald, Impedance spectroscopy: theory, experiment, and applications, John Wiley & Sons, 2018.
- [10] J. R. Macdonald, J. Schoonman, A. Lehen, Applicability and power of complex nonlinear least squares for the analysis of impedance and admittance data, Journal of Electroanalytical Chemistry and Interfacial Electrochemistry 131 (1982) 77–95.
- [11] J. R. Macdonald, J. Garber, Analysis of impedance and admittance data for solids and liquids, Journal of the Electrochemical Society 124 (7) (1977) 1022–1030.
- [12] D. D. Stupin, S. V. Koniakhin, N. A. Verlov, M. V. Dubina, Adaptive filtering to enhance noise immunity of impedance and admittance spectroscopy: Comparison with Fourier transformation, Phys. Rev. Applied 7 (2017) 054024. doi:10.1103/PhysRevApplied.7.054024.
- [13] Rahman, Abdur Rub Abdur and Lo, Chun-Min and Bhansali, Shekhar *et al.*, A detailed model for high-frequency impedance characterization of ovarian cancer epithelial cell layer using ECIS electrodes, IEEE transactions on biomedical engineering 56 (2) (2009) 485–492.

- [14] Y. Qiu, R. Liao, X. Zhang, Real-time monitoring primary cardiomyocyte adhesion based on electrochemical impedance spectroscopy and electrical cell-substrate impedance sensing, *Analytical chemistry* 80 (4) (2008) 990–996.
- [15] Y. Qiu, R. Liao, X. Zhang, Intervention of cardiomyocyte death based on real-time monitoring of cell adhesion through impedance sensing, *Biosensors and Bioelectronics* 25 (1) (2009) 147–153.
- [16] A. Ganguly, Passively Addressable Electrochemical Sensor for Detection of Chloride Ions in Ultra-Low Volumes of Human Sweat, The University of Texas at Dallas, 2019.
- [17] K. M. Ismail, S. S. Hassan, S. S. Medany, M. A. Hefnawy, A facile sonochemical synthesis of the Zn-based metal-organic framework for electrochemical sensing of paracetamol, *Materials Advances* 5 (14) (2024) 5870–5884.
- [18] A. Zarei, A. Hatefi-Mehrjardi, M. A. Karimi, A. Mohadesi, Impedimetric glucose biosensing based on drop-cast of porous graphene, nafion, ferrocene, and glucose oxidase biocomposite optimized by central composite design, *Journal of Electroanalytical Chemistry* 919 (2022) 116544.
- [19] R. K. Shervedani, A. H. Mehrjardi, N. Zamiri, A novel method for glucose determination based on electrochemical impedance spectroscopy using glucose oxidase self-assembled biosensor, *Bioelectrochemistry* 69 (2) (2006) 201–208.
- [20] S. Lee, K. J. Price, D.-H. Kim, Two local built-in potentials of h2s processed cztsse by complex impedance spectroscopy, *Solar Energy* 225 (2021) 11–18.
- [21] D. Crain, S. Rock, J. Garland, D. Roy, Comparison of DC and AC electroanalytical methods for measuring diode ideality factors and series resistances of silicon solar cells, *Current Applied Physics* 13 (9) (2013) 2087–2097.
- [22] J. Macdonald, CNLS (complex nonlinear least squares): Immittance, inversion, and simulation fitting programs–LEVM manual. version 8.08, issue data: February, 2007d. W. Marquardt, *SIAM Journal on Applied Mathematics* 11 (1963) 431.
- [23] D. D. Macdonald, A method for estimating impedance parameters for electrochemical systems that exhibit pseudoinductance, *Journal of the Electrochemical Society* 125 (12) (1978) 2062–2064.

- [24] Y.-T. Tsai, D. Whitmore, Nonlinear least-squares analyses of complex impedance and admittance data for solid electrolytes, *Solid State Ionics* 7 (2) (1982) 129–139.
- [25] J. Nocedal, S. J. Wright, *Numerical optimization*, Springer, 1999.
- [26] Ross Macdonald, *LEVM / LEVMW Manual* (Version 8.13 (May 2015)).
URL <https://jrossmacdonald.com/jrm/wp-content/uploads/LEVMMANUAL.pdf>
- [27] BioLogic Science Instruments, *EC-Lab Software User’s Manual* (Version 10.38 – August 2014).
URL [https://mmrc.caltech.edu/BioLogic%20Echem/ECLab%20Manuals/EC-Lab%20software%](https://mmrc.caltech.edu/BioLogic%20Echem/ECLab%20Manuals/EC-Lab%20software%20User's%20Manual.pdf)
- [28] D. Naranjo-Hernández, J. Reina-Tosina, L. M. Roa, G. Barbarov-Rostán, F. Ortega-Ruiz, P. C. Ramos, Smart bioimpedance device for the assessment of peripheral muscles in patients with COPD, *Sensors (Basel, Switzerland)* 24 (14) (2024).
- [29] Scribner Associates, Inc., *ZView 4.0 Operating Manual Impedance/Gain-Phase Graphing & Analysis Software* (Rev. C, 2/2023).
- [30] Boitsova, N.A., Abelit, A.A., Verlov, N.A., Stupin, D.D, Using Monte-Carlo based randomisation for stabilisation of data fitting in bioimpedance spectroscopy: Proof-of-concept, *St. Petersburg State Polytechnical University Journal. Physics and Mathematics*. 17 (2024) 315. doi:10.18721/JPM.173.164.
- [31] F. Cavallini, M. Tarantola, Ecis based wounding and reorganization of cardiomyocytes and fibroblasts in co-cultures, *Progress in biophysics and molecular biology* 144 (2019) 116–127.
- [32] V. Yrjänä, DearEIS – a GUI program for analyzing impedance spectra, *Journal of Open Source Software* 7 (80) (2022) 4808. doi:10.21105/joss.04808.
URL <https://doi.org/10.21105/joss.04808>
- [33] Matthew D. Murbach *et al.*, Getting started with impedance.py.
URL <https://impedancepy.readthedocs.io/en/latest/getting-started.html>
- [34] Matthew D. Murbach *et al.*, Impedance.py frequently asked questions.
URL <https://impedancepy.readthedocs.io/en/latest/faq.html>
- [35] V. Yrjänä, Pyimpspec source code.
URL <https://github.com/vyrjana/pyimpspec>

- [36] K. B. Knudsen, PyEIS manual.
URL <https://github.com/kbknudsen/PyEIS>
- [37] ImpedanceAnalyzer source code.
URL <https://github.com/mdmurbach/ImpedanceAnalyzer>
- [38] EIS spectrum analyser help.
URL <http://www.abc.chemistry.bsu.by/vi/analyser/fitting.html>
- [39] R. Zhang, R. Black, D. Sur, P. Karimi, K. Li, B. DeCost, J. R. Scully, J. Hatrick-Simpers, AutoEIS: automated bayesian model selection and analysis for electrochemical impedance spectroscopy, *Journal of The Electrochemical Society* 170 (8) (2023) 086502.
- [40] AutoEIS source code.
URL <https://github.com/AUTODIAL/AutoEIS>
- [41] OSIF source code.
URL <https://github.com/NREL/OSIF>
- [42] Kobayashi, Kiyoshi and Suzuki, Tohru S, Free analysis and visualization programs for electrochemical impedance spectroscopy coded in python, *Electrochemistry* 89 (2) (2021) 218–222.
- [43] pyZwx manual.
URL https://www.nims.go.jp/pyZwx/en/pyZwx_UserGuide/userGuide-e.html
- [44] O. H. Maghsoudi, A. Vahedipour, J. Gerstenhaber, S. P. George, T. Hallowell, B. Robertson, M. Short, A. Spence, Matlab software for impedance spectroscopy designed for neuroscience applications, *Journal of neuroscience methods* 307 (2018) 70–83.
- [45] B. A. Boukamp, A nonlinear least squares fit procedure for analysis of immitance data of electrochemical systems, *Solid state ionics* 20 (1) (1986) 31–44.
- [46] D. Stupin, A. Lihachev, A. Nashchekin, How to take fractional-order derivative experimentally?, in: *Journal of Physics: Conference Series*, Vol. 1124, IOP Publishing, 2018, p. 071011.
- [47] BioElectronics Lab: D. Stupin, A. Abelit, N. Boitsova, NELM software 1.0 beta (2025).
URL <https://github.com/BioElectronicsLab/NELM-1.0-beta>

- [48] BioElectronics Lab: D. Stupin, A. Abelit, N. Boitsova, NELM software playlist (2025).
URL <https://www.youtube.com/playlist?list=PLGs8WJXPVSmXVLjqGhC6q4nZcFmngpXkf>
- [49] J. C. Lagarias, J. A. Reeds, M. H. Wright, P. E. Wright, Convergence properties of the Nelder–Mead simplex method in low dimensions, *SIAM Journal on optimization* 9 (1) (1998) 112–147.
- [50] C. T. Kelley, *Iterative methods for optimization*, SIAM, 1999.
- [51] basinhopping help.
URL <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.basinhopping.html>
- [52] N. Ke, X. Wang, X. Xu, Y. A. Abassi, The xCELLigence system for real-time and label-free monitoring of cell viability, in: *Mammalian Cell Viability*, Springer, 2011, pp. 33–43.
- [53] A. Golke, J. Cymerys, A. Słóńska, T. Dzieciatkowski, A. Chmielewska, A. Tucholska, M. Bańbura, The xCELLigence system for real-time and label-free analysis of neuronal and dermal cell response to Equine Herpesvirus type 1 infection, *Polish journal of veterinary sciences* (2012).
- [54] C. L. Lawson, R. J. Hanson, *Solving least squares problems*, *Classics in applied mathematics* 15, Society for Industrial and Applied Mathematics, 1987.
- [55] J. R. Rice, *Matrix Computations and Mathematical Software* (McGraw-Hill Computer Science Series), McGraw-Hill Companies, 1981.
- [56] J.-B. Hiriart-Urruty (auth.), Reiner Horst, Panos M. Pardalos (eds.), *Handbook of Global Optimization*, 1st Edition, *Nonconvex Optimization and Its Applications* 2, Springer US, 1995.
- [57] S. J. Hanson, Confidence intervals for nonlinear regression: A basic program, *Behavior Research Methods & Instrumentation* 10 (3) (1978) 437–441.
- [58] Ivchenko G.I and Medvedev Yu. I., *Introduction to statistics (In Russian)* (2010).
- [59] G. Popkirov, R. Schindler, A new impedance spectrometer for the investigation of electrochemical systems, *Review of scientific instruments* 63 (11) (1992) 5366–5372.

- [60] M. Leisner, J. Carstensen, H. Föll, FFT impedance spectroscopy analysis of the growth of anodic oxides on (100) p-Si for various solvents, *Journal of Electroanalytical Chemistry* 615 (2) (2008) 124–134.
- [61] B.-Y. Chang, S.-M. Park, Electrochemical impedance spectroscopy, *Annual Review of Analytical Chemistry* 3 (2010) 207–229.
- [62] D. Stupin, A. Kornev, N. Filatov, S. Koniakhin, Cell-substrate lift-off lithography: Proof of concept, in: *Journal of Physics: Conference Series*, Vol. 1135, IOP Publishing, 2018, p. 012015.
- [63] E. O. Brigham, *The fast Fourier transform: An introduction to its theory and application*, prentice-hall Edition, Prentice Hall, 1973.
- [64] B. Y.-K. Hu, Kramers–kronig in two lines, *Am. J. Phys* 57 (9) (1989) 821.
- [65] M. Urquidi-Macdonald, S. Real, D. D. Macdonald, Applications of Kramers–Kronig transforms in the analysis of electrochemical impedance data III. stability and linearity, *Electrochimica Acta* 35 (10) (1990) 1559–1566.
- [66] A. Gavriluk, D. Osinkin, D. Bronin, The use of Tikhonov regularization method for calculating the distribution function of relaxation times in impedance spectroscopy, *Russian Journal of Electrochemistry* 53 (2017) 575–588.
- [67] S. M. Sze, *Physics of Semiconductor Devices*, 2nd Edition, Wiley-Interscience, 1981.
- [68] Vishay Semiconductors, BPW20RF manual, Fig. 6, Vishay Semiconductor GmbH, P.O.B. 3535, D-74025 Heilbronn, Germany (Rev. 1, 19–Mar–02).
- [69] A. Rohatgi, Webplotdigitizer.
URL <https://automeris.io>
- [70] Y. Cui, Y. An, T. Jin, F. Zhang, P. He, Real-time monitoring of skin wound healing on nano-grooves topography using electric cell-substrate impedance sensing (ECIS), *Sensors and Actuators B: Chemical* 250 (2017) 461–468.
- [71] D. Das, F. A. Kamil, K. Biswas, S. Das, Electrical characterization of suspended hela cells using ECIS based biosensor, in: *2012 Sixth International Conference on Sensing Technology (ICST)*, IEEE, 2012, pp. 734–737.
- [72] J. R. Masters, HeLa cells 50 years on: the good, the bad and the ugly, *Nature Reviews Cancer* 2 (4) (2002) 315.

- [73] I. Giaever, C. R. Keese (auth.), W. G. Jiang (eds.), Electric Cell-Substrate Impedance Sensing and Cancer Metastasis, 1st Edition, Cancer Metastasis - Biology and Treatment 17, Springer Netherlands, 2012.
- [74] I. Giaever, C. R. Keese, A morphological biosensor for mammalian cells., *Nature* 366 (6455) (1993) 591.
- [75] Z. Kerner, T. Pajkossy, On the origin of capacitance dispersion of rough electrodes, *Electrochimica Acta* 46 (2-3) (2000) 207–211.
- [76] T. Pajkossy, Impedance of rough capacitive electrodes, *Journal of Electroanalytical Chemistry* 364 (1-2) (1994) 111–125.
- [77] D. D. Stupin, A. A. Abelit, A. S. Mereshchenko, M. S. Panov, M. N. Ryazantsev, Copper–ruthenium composite as perspective material for bioelectrodes: Laser-assisted synthesis, biocompatibility study, and an impedance-based cellular biosensor as proof of concept, *Biosensors* 12 (7) (2022) 527.
- [78] M. S. Panov, A. E. Grishankina, D. D. Stupin, A. I. Lihachev, V. N. Mironov, D. M. Strashkov, E. M. Khairullina, I. I. Tumkin, M. N. Ryazantsev, *In Situ* laser-induced fabrication of a ruthenium-based microelectrode for non-enzymatic dopamine sensing, *Materials* 13 (23) (2020) 5385.
- [79] R. De Levie, The influence of surface roughness of solid electrodes on electrochemical measurements, *Electrochimica Acta* 10 (2) (1965) 113–130.
- [80] S. Liu, Fractal model for the AC response of a rough interface, *Physical Review Letters* 55 (5) (1985) 529.
- [81] M. Itagaki, Y. Hatada, I. Shitanda, K. Watanabe, Complex impedance spectra of porous electrode with fractal structure, *Electrochimica Acta* 55 (21) (2010) 6255–6262.
- [82] B. Widrow, P. N. Stearns, Adaptive Signal Processing, Prentice-Hall signal processing series, Prentice-Hall, 1985.