

JEDS 2.0 - Users Manual

JEDS 2.0 solves the Nernst-Planck-Poisson (NPP) problem with/without the reaction term, using Neumann-like boundary conditions for system consisted of arbitrary number of phases. The Method of Lines with new unconditionally consistent space discretization (which is one order of approximation higher than Brumleve and Buck) is used. The resulting system of non-linear ordinary differential equations with initial conditions (Cauchy problem) can be solved using a method with adaptive time step. Software uses the class `StiffIntegratorT`, an implicit integrator based on the procedure RADAU5, which has been implemented in C++ by B. Ashby. Source of this class is an object oriented interpretation of the procedure, written originally in Fortran by E. Hairer and G. Wanner [i].

Computer program has been written in object oriented language - C++^[ii]. Software is freely available for anyone interested in using it. The presented software serves as a flexible and powerful tool that enables inspection of numerous cases, i.e. liquid-junction potential in reference electrodes, potentiometric response of ion-selective electrodes with ion-exchanger and neutral-carrier membranes, electrochemical impedance spectroscopy and other applications. Results obtained using previous versions of this software (JEDS 1.x) has been used in the several scientific papers, e.g. [iii,iv,v].

In this manual, the way of execution of the program and structure of input-files is explained, program working scheme is presented and the structure of output-files with results is shown.

1 USING JEDS 2.0

In order to run the program you need to have MS DOS or MS Windows operating system. Under Linux based systems, program can be used under wine emulator. No special configuration of the hardware is required in order to run JEDS 2.0, although time of calculations strongly depends on the speed of the processor used.

The installation procedure is very simple. Simply, copy "JEDS.exe" executable file to the folder with your input files and run it. The user has to provide the data, which is stored in the text format input-file. The results calculations made by computer program are saved in output files.

2 INPUT DATA FILES

Input file with ".cci" extension is an ASCII text file, which contains all data necessary to perform simulation. This file was divided into following sections:

- First Section consists of:
 - Header #JEDS2 informs the software that input-file provided by user is a correct input

file for JEDS 2.0.

Then the pairs of lines contain one after another: first line - short description of physical parameters and second line - suitable values. All values in the input files have to be expressed using SI units. Parameters describing the system are:

- Used Model (in Beta version value has to be set to 0)
- First value is the electrochemical method simulated - choose 0 for Potentiometry, or 1 for EIS (in Beta1 version, 2-Chronopotentiometry is not supported)
Second value (important only if method different than potentiometry is used) is the the value of external current
- n - number of layers (phases),
- d (n values) - thickness of the respective layer [m],
- r (number of components),
- T_{const} - absolute temperature,
- $\text{epsilon}[\text{layers}]$ - dielectric constant in each layer,
- t_{End} - time of the process duration [s] (for each point of calibration curve!);

Then two pairs of lines describing calibration curve:

- Calibration curve parameters (First value is the phase for which the change occur, second represents the number of the ion, which concentration is changed. Third value is the number of the opposite-charged ion, which concentration is changed to provide electroneutrality, and the fourth one is the number of points in the calibration curve),
- calibration curve iterations - list of multipliers for each point of the calibration curve),

- Next section starts after one line of bars. It contains following information about each component in the system (line by line):
 - description of the component,
 - diffusivity of the component in each layer (n values),
 - charge of the component (1 value),
 - concentrations ($n+2$ values): first value represents c_L – concentration of the component in the left bulk. Following n values are $c_M[\text{layers}]$ – the initial concentration of the component in each layer. Last value represents c_R – concentration of the component in the right bulk,
 - K_{if} ($n+1$ values) – the forward heterogeneous rate constants at each boundary,
 - K_{ib} ($n+1$ values) – the backward heterogeneous rate constants at each boundary.
 - number of the component representing the ligand, the component is reactive with. For all non-reacting components, ligands and complexes, you should choose 0. If 0 is chosen, three of the lines below do not appear in the file.
 - number of the component representing complex, which is formed in reaction
 - $k_{i,as}$ – association constants in each of the layers (n values)
 - $k_{i,dis}$ - dissociation constants in each of the layers (n values)
- Last section (after one line of bars) contains information about rescaling parameters, discretization options and additional input-output options:
 - absolute and relative tolerance (for RADAU solver)
 - values of the scaling constants L , t_c , c_{max} , E_{max} and the first step of space discretization (is value 0 is provided values will be set automatically)
 - grid type for each layer (0 – grid denser near both boundaries, 1 - grid denser near the left boundary, 2 - grid denser near the right boundary),
 - N_j - number of discretization points in each layer j (n values),
 - number of points near the boundaries, having constant distance of the first step of space discretization, in each layer j (n values),
 - load initial concentration from $y.\text{vec}$ – if set to 1, the initial conditions will be imported from external file (file $y.\text{vec}$ has to be in the same folder where JEDS is executed)
 - Generate the PostScript plots (wyn.ps , et.ps and curve.ps)

3 ALGORITHM OF THE PROGRAM

JEDS 2.0 can be run from the console in two ways: 1) with one parameter, which represents the name of input file (without extension “.cci”), that will be calculated or 2) with no parameters. If the parameter will not be provided, program will use the file “b.cci” (equivalent to running the program as follows: “JEDS b”).

The Algorithm of the program is shown in Figure 1. First the program rescales the variables and generates a spatial grid. Values of space grid points $x_{j,k}$, $y_{j,k}$ and steps $h_{j,k}$ are stored in xh.dat file. Then folders and subfolders required for the output files are created (see *Results and output data* files). The dane.html file is also saved – it contains detailed information on calculations.

Program runs the number of iterations (loops) provided by user in input file (number of the points in the calibration curve). In each iteration:

- Initial concentrations and values of electrical field are provided. For the first loop the flat-band constant concentration from *.cci file and zero electrical field condition are read in. Alternatively, the initial values can be loaded from the y.vec file. For each next loop values of concentration and electrical field obtained at the end of previous loop are used.
- Now the time-dependent vector of values is solved in zero-current conditions. After each step of the integrating method the actual electrical potential is calculated and displayed on the screen.
- If user in *.cci file decided to perform any simulations with non-zero current, the constant current is applied to the system. Then impedance spectrums are calculated.

The values of end-time potential are calculated in each loop. After executing all the loops the calibration curve as well as the information about the detection limit are calculated and files ccurve.dat, tcurve.dat, curve.ps, tcurve.ps and limit.txt are created.

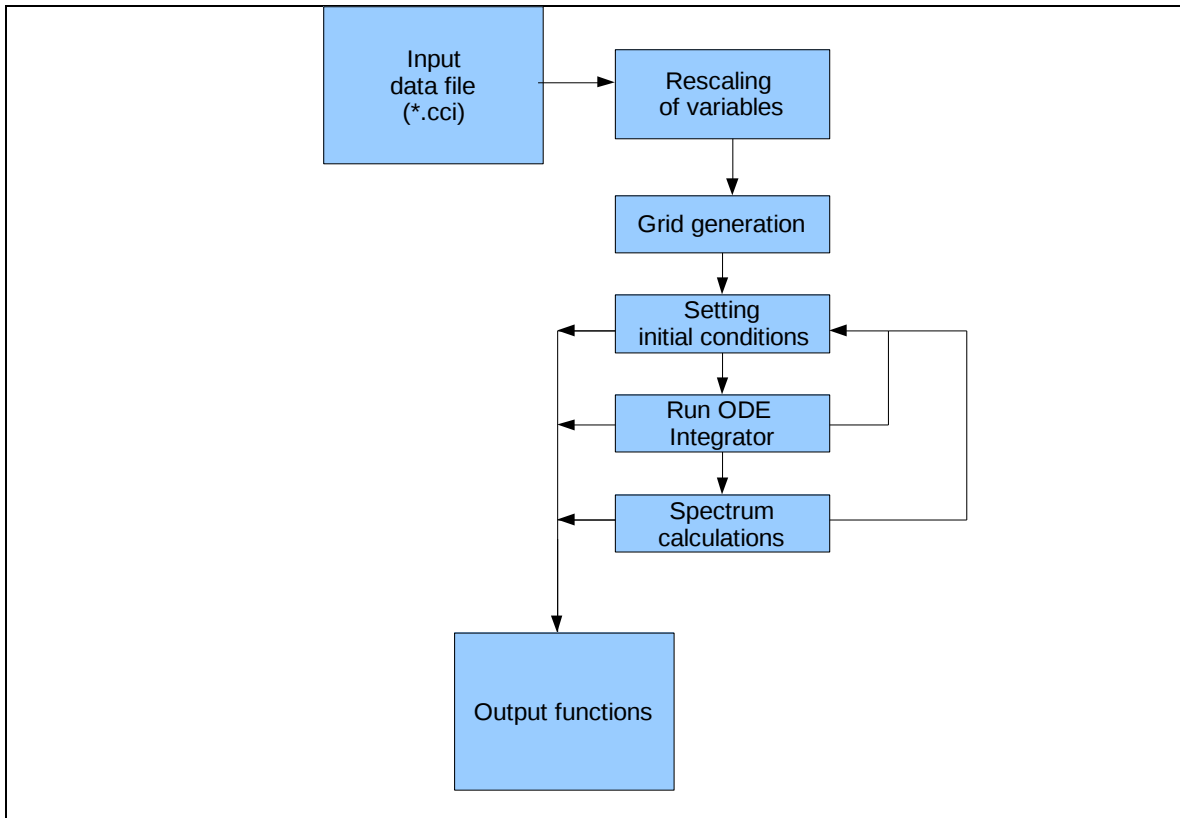


Figure 1 The simplified algorithm of the JEDS program.

4 RESULTS AND OUTPUT DATA FILES

Computer program generates a folder of the same name as the provided *.cci input file and all subfolders required for output data files. Subfolders names correspond to the values of provided multipliers (if there's only one all results will be saved in main folder). If spectrum is calculated, subfolders called "spectrum" are created in the folders above.

Main folder contains the following output files:

- ccurve.dat – contains a calibration curve for an electrode. It is the ASCII file, which consist of the three columns:
 - value of multiplier (which corresponds concentration of changed ion),
 - value of overall electrical potential calculated by the software,
 - value of time required to perform the calculations.
- curve.ps is a graphical representation of the above data in PostScript format.
- tcurve.dat is the ASCII file with the potential for whole calibration process as a function of time. Two columns represents:
 - time for each time-step,
 - potential obtained for each time-step.
- tcurve.ps is a graphical representation of the above data in PostScript format.
- dane.html is a documentation of input data provided to calculations and information on values of the rescaling constants and brief description on the calculation grid.
- limit.txt is the ASCII file which contains calculated values of the Nernstian slope and the value of $\text{slope} \cdot \log(2)$ subsequently used to calculate the detection limit of the ISE calibration curve using method described by Sokalski^[vi].

Subfolders contain:

- xh.dat describes the discretization. It is the ASCII file, which consists of the four columns:
 - the point's number in spatial discretization,
 - values of space position of $x_{j,k}$ points,
 - values of space position of $y_{j,k}$ points,
 - values of steps $h_{j,k}$ in the space discretization.
- y.vec is the vector of values, which contains:
 - size of the vector for numerical procedure,
 - time of the process (t_{proc}) after which file was made,
 - values of one-dimensional vector solved by integrator.
- et.dat shows potential-time dependency. It is the ASCII file, which consist of the three columns:
 - time of process t_{proc} [s], after each time step of integrator,
 - overall electrical potential after each time step of integrator,
 - time t_{calc} [s] required to calculate all previous steps.
- mempot.ps is a graphical representation in PostScript format of following dependencies:
 - overall electrical potential $\Delta\varphi$ as a function of time of the process - $\Delta\varphi(t)$, in linear scale,
 - $\Delta\varphi(t)$, in logarithmic scale - $\Delta\varphi(\log(t))$,
 - time required to perform calculations for the given time of process in double logarithmic scale - $\log(t_{\text{calc}})(\log(t_{\text{proc}}))$.
- ywyn.dat shows values of concentration in each $y_{j,k}$ point at time t_{END} . It is an ASCII file, which consists of three columns:
 - values of $y_{j,k}$,
 - r columns with $c_i^{j,k}$ - values of concentration in points $y_{j,k}$,

- deviation from electroneutrality condition $\sum_{i=1}^r z_i c_i^{j,k}$ in points $y_{j,k}$.
- xwyn.dat contains values of fluxes and electrical field at each discretization points $x_{j,k}$ at time t_{END} . The following columns contain:
 - values of $x_{j,k}$,
 - $E^{j,k}$ - values of the electrical field potential at points $x_{j,k}$,
 - $\varphi^{j,k}$ - values of the potential at points $x_{j,k}$,
 - r columns of fluxes $J_i^{j,k}$ at points $x_{j,k}$,
 - the electric current $\sum_{i=1}^R z_i J_i^k$ at points $x_{j,k}$.
- wyn.ps is a graphical representation in PostScript format of following dependencies obtained for time t_{END} :
 - $c_i^j(x)$ – concentration of i-th component as a function of space,
 - $\sum_{i=1}^r z_i c_i^j(x)$ – deviation from electroneutrality condition as a function of space,
 - $J_i^j(x)$ – flux of i-th component as a function of space,
 - $\sum_{i=1}^r z_i J_i^j(x)$ – the electric current as a function of space,
 - $E^j(x)$ – electrical field as a function of space,
 - $\varphi^j(x)$ - electrical potential as a function of space.
- spectrum.dat and its graphical representation, spectrum.ps, are created only if Electrochemical method was chosen as EIS. It's a Nyqvist plot of impedance spectrum ($Z'(\omega)$, $Z''(\omega)$).
- folder spectrum contains all files which describe calculations with non-zero current (y.vec, et.dat, mempot.ps, ywyn.dat, xwyn.dat and wyn.ps have the structure identical with their equivalents for zero-current simulations).

5. References:

-
- [i] Hairer, E.; Wanner, G.; *J. Comput. Appl. Math.* **1999**, 111, 93.
 - [ii] B. Stroustrup: The C++ Programming Language (3rd edition). Addison Wesley Longman, Reading, MA. 1997
 - [iii] Jasielec, J.J.; Szyszkiewicz, K.; Filipek, R.; Fausek, J.; Danielewski, M.; Lewenstam, A.; *Comp. Mat. Sci.*, **2012**, 63, 75–90.
 - [iv] Jasielec, J.J.; Sokalski, T.; Filipek, R.; Lewenstam, A.; *Electrochim. Acta* **2010**, 55, 6836.
 - [v] Jasielec, J.J.; Wierzba, B.; Grysakowski, B.; Sokalski, T.; Danielewski, M.; Lewenstam, A.; *ECS Trans.* **2011**, 33, 19.
 - [vi] T. Sokalski, A. Ceresa, M. Fibbioli, T. Zwickl, E. Bakker, E. Pretsch *Anal. Chem.* 1999, 71, 1210-1214