

Kolomeets A.V.

«SPARK»

*Software for Photochemical kinetic Analysis using
Runge-Kutta method with global optimization*

Program for fitting kinetic data of laser flash
photolysis



Novosibirsk 2011.

Abstract

The program “SPARK” developed in the Institute of Chemical Kinetics and Combustion SB RAS is designed for fitting of experimental kinetic curves of optical density change within a given system of differential equations. The main goal of this program is to provide a user with a powerful tool for fitting data obtained with laser flash photolysis methods. This software, using the 4th order Runge-Kutta method combined with global optimization, allows one to perform automatic fit for hundreds of input parameters of several types (rate constants, absorbance coefficients and initial concentrations).

Content

Abstract	2
Main purpose	5
General description of the features	5
Scope and limitations	6
About the author	7
License	7
Installation	7
Updates	7
1. Interface	8
1.1. Main interface	8
1.1.1. <i>Description of Toolbar</i>	9
1.1.2. <i>Description of Graph Panel</i>	9
1.1.3. <i>Description of Model panel</i>	10
1.1.4. <i>Description of Kinetic Curve Edit panel</i>	11
1.1.5. <i>Description of Constants panel</i>	11
1.1.6. <i>Description of Equations panel</i>	11
1.1.7. <i>Description of Options panel</i>	12
1.2. Interface for spectral section visualization (Spec section)	14
1.3. Interface for optical spectra manipulation (Set Abs)	16
1.3.1. <i>Overview and description</i>	16
1.3.2. <i>Setting absorbance coefficients</i>	18
1.3.3. <i>Managing the view</i>	18
1.4. Interface for automatic approximation of input parameters (Global fit)	20
1.5. Interface for automatic approximation of the input parameters with advanced settings (Global fit)	22
1.6. Interface for setting ranges of variation for input parameters (Set limits)	24
1.7. Interface for creating correlations between initial concentrations values (Xi(0) Relations)	26
2. Tutorial	27
2.1. Format of files	27
2.2. Input format of differential equations	27
2.3. Fitting of loaded kinetic curves	29
2.4. Automatic approximation of parameters	33
2.5. Managing absorbance spectra	35
2.5.1. <i>Creating a new absorbance spectrum</i>	35
2.5.2. <i>Loading optical spectrum from file</i>	36
2.6. Tips about program usage	38

3. Description of algorithms	39
3.1. The 4th order Runge-Kutta method	39
3.2. Selection of computational algorithms	41
3.3. Calculation of absorbance coefficients	44
3.4. Algorithm «Gradient descent»	45
3.5. Algorithm «Direct»	46
3.6. Combined algorithms	50
3.6.1. <i>Combined algorithm «Gradient»</i>	51
3.6.2. <i>Combined algorithm «Direct»</i>	52
3.6.3. <i>Algorithm for optimization of initial concentrations values</i>	53
3.7. Error calculation	54
Conclusion	55
References	56

Main purpose

This program was developed to fit experimental curves of optical density change, obtained with laser flash photolysis methods, within a given scheme of differential equations. The program is based on the 4th order Runge-Kutta method in conjunction with algorithms for global optimization. The need for such a software was caused by inconvenience of usage of existing mathematical programs (Mathlab, Mathcad, Mathematica etc.), which, though being very powerful math tools, fail to provide sufficient interface for kinetic curve fitting. The number of variables (rate constants, absorbance coefficients, initial concentrations) in many problems of chemical kinetics and photochemistry may reach hundreds, which makes it reasonable to develop a new specialized software for implementing an automatic search of optimal values of input parameters.

General description of the features:

- Solving systems of differential equations with given parameters for an unlimited number of experimental kinetic curves.
- Automatic search for optimal values of input parameters in a scheme of differential equations for a given set of kinetic curves.
- Saving and loading a calculation model* and a working project*.
- Loading absorbance spectra from external text files (allows *.csv, *.asc *.txt etc.) to use file data as input parameters.

*Calculation model includes differential equations, values of initial concentrations, rate constants and cuvette thickness.

**Working project includes all experimental kinetic curve data, fitting ranges for each curve, calculation model parameters and the values of absorbance coefficients of reaction participants at all available wavelengths.

Scope and limitations

The main application of this software is the study of photochemical processes in the liquid phase, occurring mainly in the nanosecond and longer time scales. The experimental data are loaded in the form of kinetic curves of optical density change, obtained by laser flash photolysis with registration at different wavelengths or by the conversion of stationary optical spectra. Analysis of experimental data includes the determination of the rate constants of intermediate reactions and spectral characteristics of all participants.

It should be noted that this program can be used for fitting kinetic curves obtained by non-photochemical methods (e.g. system pressure change, dynamics in EPR spectra, etc.).

It is important to note that there are some limitations in using this software. Firstly, it is not designed to fit optical luminescence data. Secondly, it can not be used to fit femtosecond spectroscopy data since such functions as convolution of instrument response, calculation of group velocity dispersion, and solvent relaxation calculations are not included. Fitting of kinetic processes, whose behavior can not be described by a system of differential equations (e.g. probabilistic relaxation processes in the solid phase) might be challenging.

About the author

This product was developed by Kolomeets Aleksandr Victorovich, researcher at the laboratory of photochemistry of the Institute of Chemical Kinetics and Combustion SB RAS, Russia, Novosibirsk.

License

This program is distributed under the GNU GPL license (<http://www.gnu.org/licenses/gpl.html>). The author is not responsible for any results that may be obtained using the program, as well as possible damage it may cause. If you find any bug in the program operation, feel free to contact the author at kolomeets@live.ru.

Installation

This program was written using Java programming language. To run it on your computer you need to have Java Platform JDK 6 (or any later version) installed. The latter can be downloaded at:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

Updates

Program and manual updates can be downloaded on the official site: <http://code.google.com/p/spark-photochem/>

1. Interface

This program provides a user with a set of powerful tools for convenient fitting of experimental kinetic data. This section gives a brief description about functions of most of the interface elements.

1.1. Main interface

The main interface window is divided into several working panels (Fig. 1.1):

- **Toolbar** – contains buttons for adding new experimental kinetic curves, saving the current working project or loading a new one.
- **Graph Panel** – a panel for kinetic curve visualization.
- **Error Panel** – a window to display the values of the deviations of the experimental data from the calculated curves.
- **Model** – a panel to control the parameters of the calculation model (details below).
- **Constants** – a panel to display values of rate constants.
- **Equations** – a panel containing information about the system of differential equations and initial concentrations of reaction participants.
- **Options** – additional fitting options.

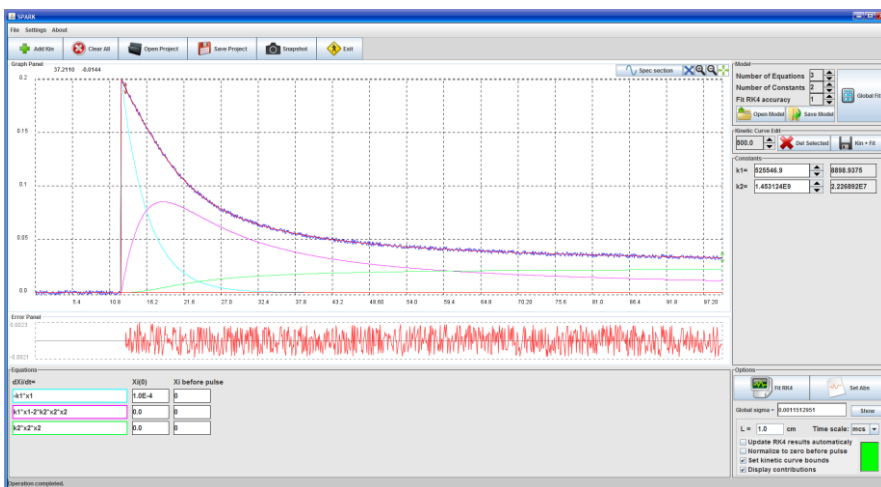













Fig. 1.1. Main interface of program “SPARK”.

1.1.1. Description of Toolbar

The description of the functions of the Toolbar panel items is presented in the table below:

Item	Function
 Add Kin	Adding new experimental kinetic curves.
 Clear All	Removing all experimental data from the current project.
 Open Project	Loading a working project from a file (*.project)
 Save Project	Saving the current working project.
 Snapshot	Getting a snapshot of “Graph panel” to save it to a file.
 Exit	Exiting the program.

1.1.2. Description of Graph Panel

Item	Function
	Calling the interface for setting the boundaries of the visible area (Fig. 1.2).
	Zooming out.
	Zooming in for a selected area of the chart (selection is performed using the mouse while holding the left button, Fig. 1.3).
	Rescaling the image to display all points of the kinetic curves (same as clicking the right mouse button).
 Spec section	Opens the interface for visualizing spectral section (time slice of experimental and calculated kinetic curves) (see § 1.2).

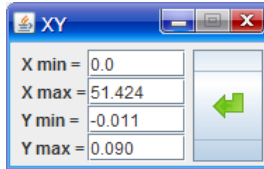


Fig. 1.2. Interface for setting the visible area of the chart.

Additional features:

- Dragging the mouse while holding the scroll button allows one to shift the view along X and Y axes.
- Clicking the right mouse button restores the original image scale.

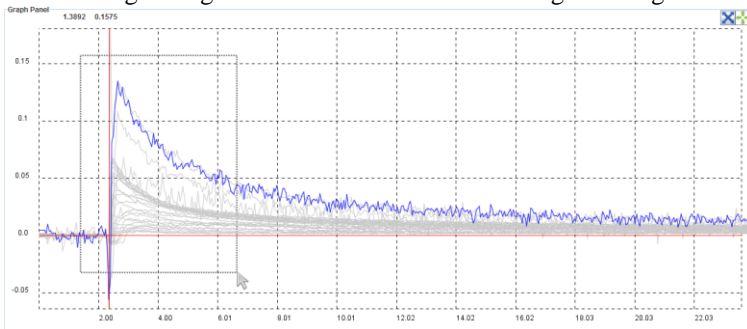











Fig. 1.3. Graph area selection.


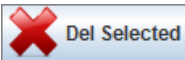

1.1.3. Description of Model panel

Item	Function
Number of Equations <input type="text" value="2"/>   Number of Constants <input type="text" value="1"/>  	Setting the number of differential equations and rate constants.
Fit RK4 accuracy <input type="text" value="1"/>  	Setting accuracy for the 4 th order Runge-Kutta method. Higher value corresponds to a reduction of the integration step (see § 3.1).*
 Save Model	Saving the current calculation model.
 Open Model	Loading a previously saved calculation model.


	Opens the “Global fit” interface for automatic search of the input parameters (see § 1.4).
---	--

*It is reasonable to increase the “Fit RK4 accuracy” value only when dealing with high values of rate constant. For example for $k_1 > 10^7$ (monomolecular reaction rate constant) and $k_2 > 5 \times 10^{10}$ (bimolecular reaction rate constant).


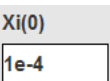

1.1.4. Description of Kinetic Curve Edit panel

Item	Function
	Switching between all available experimental kinetic curves with highlighting the active one.
	Removing the active kinetic curve.
	Saving the active experimental and calculated curves to the clipboard. Allows one to transfer the results of fitting to programs like “ORIGIN” and “Excel”.



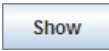

1.1.5. Description of Constants panel

Item	Function
	Displays the current value of a rate constant and its error (on the right). Clicking the buttons with arrows increases/decreases the current rate constant value by 5%.

1.1.6. Description of Equations panel

Item	Function
	Text field to type a differential equation (see § 2.2).
	Text field to type the initial concentration of a compound X with index i right after the laser pulse.
	Text field to type the concentration of a compound X with index i before the laser pulse.

1.1.7. Description of Options panel

Item	Function
	Solving the system of given differential equations with specified input parameters. Calculated kinetic curves are displayed in “Graph Panel”.
	Calls interface for absorbance spectra manipulation (see § 1.3).
Global sigma = <input type="text" value="4.203568E-4"/>	The value of the standard deviation (see § 3.2). Demonstrates goodness of fit.
	Shows dependence of the standard deviation on wavelengths in a new window (Fig. 1.4).
L = <input type="text" value="1"/> cm	The optical thickness of the cuvette used (specified in centimeters).
Time scale: <input type="text" value="mcs"/> ▼	Time scale (to specify the units on the x-axis on the “Graph panel” for kinetic curves). Enables to select from microsecond, millisecond and second time ranges.
<input checked="" type="checkbox"/> Update RK4 results automatically	When enabled, any change in values of rate constants (using arrow buttons) or absorbance coefficients (using the mouse) results in immediate recalculation of the system of differential equations to display new calculated data in “Graph panel”.
<input checked="" type="checkbox"/> Normalize to zero before pulse	Allows to determine zero level for each individual experimental kinetic curve. (Fig. 1.5).*
<input checked="" type="checkbox"/> Set kinetic curve bounds	Allows to specify fitting range for each experimental kinetic curve. Enabling this option reveals two markers at both ends for each curve. Fitting range is specified by positioning the markers appropriately (Fig. 1.1).
<input checked="" type="checkbox"/> Display contributions	Enables to display individual contributions to the total calculated optical density for each component X. Contribution curves are displayed in “Graph panel” (Fig 1.1)
	Indicator of correct markers position. Turns red if markers are set in a wrong order or if one of the markers is located before the zero line (the red vertical line).

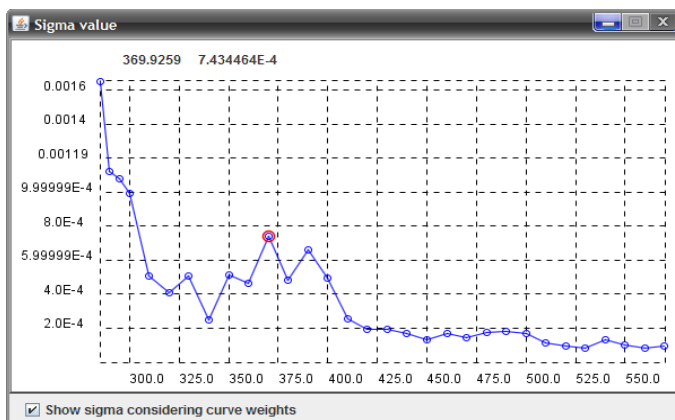


Fig.1.4. Dependence of the standard deviation value on wavelength.

*Enabling ☒ **Normalize to zero before pulse** option brings up a new green line displayed in “Graph panel” (Fig. 1.5). Adjusting its position, tell the program the time interval before the laser pulse. Then the program will automatically determine a zero level for each experimental kinetic curve.

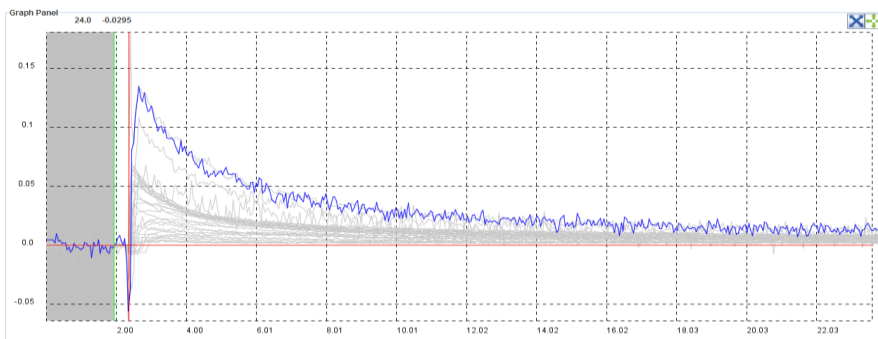


Fig.1.5. Setting time interval before the laser pulse.

1.2. Interface for spectral section visualization (Spec section)

The “Spec section” interface is designed to render the profile of optical density dependence on wavelength for experimental, calculated and contribution kinetic curves at a specified time moment.

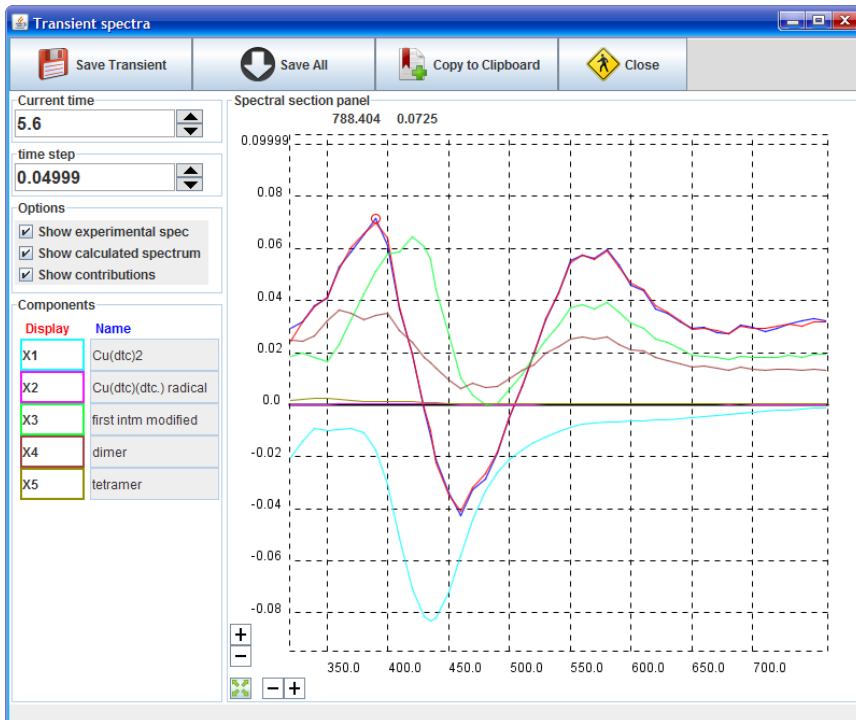


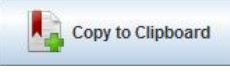



Fig.1.6. “Spec section” interface window.

Item	Function
	Saving an experimental spectral section (transient spectrum) at a specified time moment to a file (*.txt).
	Saving all spectral sections for all enabled types of kinetic curves (experimental, total calculated and contributions) at a specified time moment to a file (*.txt).
	Copying all displayed spectral sections to the clipboard.

	Closing the interface window.												
<p>Current time</p> <p>5.6</p>	Current time.*												
<p>time step</p> <p>0.04999</p>	Time step. Higher values allow one to move along the time axis faster.												
<p>Options</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Show experimental spec <input checked="" type="checkbox"/> Show calculated spectrum <input checked="" type="checkbox"/> Show contributions 	Spectral section type selection. Allows to visualize dependence of optical density on wavelength for experimental, total calculated and contribution kinetic curves.												
<table border="1"> <thead> <tr> <th>Display</th><th>Name</th></tr> </thead> <tbody> <tr> <td>X1</td><td>Cu(dtc)2</td></tr> <tr> <td>X2</td><td>Cu(dtc)(dtc.) radical</td></tr> <tr> <td>X3</td><td>first intm modified</td></tr> <tr> <td>X4</td><td>dimer</td></tr> <tr> <td>X5</td><td>tetramer</td></tr> </tbody> </table>	Display	Name	X1	Cu(dtc)2	X2	Cu(dtc)(dtc.) radical	X3	first intm modified	X4	dimer	X5	tetramer	Displays a list of available compounds X_i and color of the graph line of the corresponding spectral contribution.
Display	Name												
X1	Cu(dtc)2												
X2	Cu(dtc)(dtc.) radical												
X3	first intm modified												
X4	dimer												
X5	tetramer												

*Alternative way to change a current time value is to adjust position of a blue vertical line that is available in “Graph panel” when the “Spec section” interface window is visible.

1.3. Interface for optical spectra manipulation (Set Abs)

1.3.1. Overview and description

This interface is designed to visualize dependence of absorbance coefficients on wavelength ($\text{mol} \times \text{liter}^{-1} \times \text{cm}^{-1}$) for all participants of processes studied.

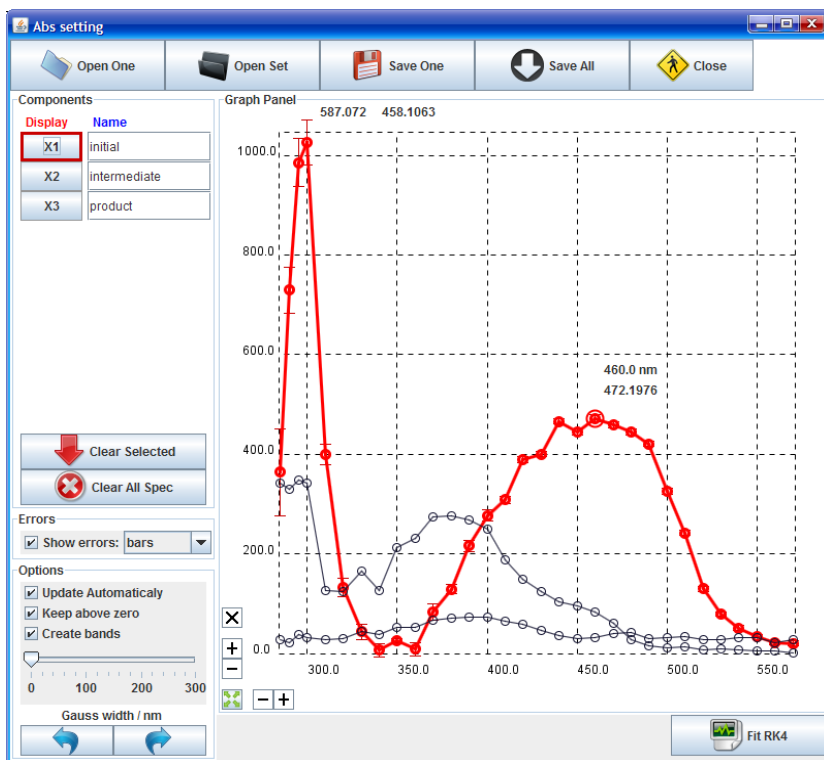





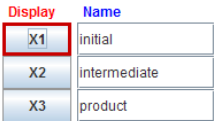
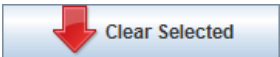
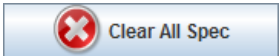
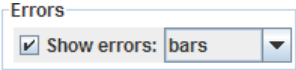
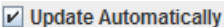

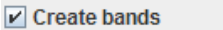






Fig.1.7. Interface for optical spectra manipulation.

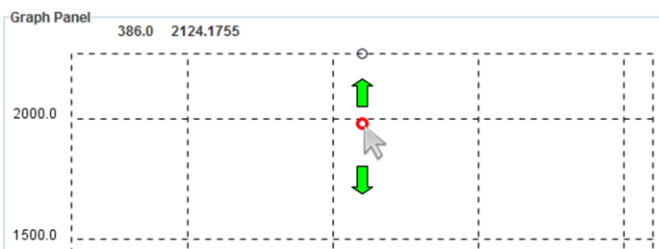
Item	Function
	Loading one optical spectrum from a file *.txt, *.asc or *.csv.
	Loading a set of spectra from a previously created file *.spec (You can save all available absorbance spectra into one file *.spec and then load them all using this button).

	Saving the selected spectrum to a file *.txt.
	Saving all available absorbance spectra to a file *.spec. Names of components X_i are also recorded to a file.
	Closing the window of the “Set Abs” interface.
	Displays all substances X_i and there names.
	Setting all absorbance coefficients to zero for a selected spectrum.
	Setting all absorbance coefficients to zero for all absorbance spectra.
	Enables to visualize errors for absorbance coefficients. Allows to show error values as bars or colored area.
	When enabled, any change in values of rate constants (using arrow buttons) or absorbance coefficients (using the mouse) results in immediate recalculation of the system of differential equations to display new calculated data in “Graph panel”.
	Keeps absorbance coefficients values above zero.
	Enables “Create bands” mode. Allows to create optical bands of Gaussian shape.
	Setting width at half-height (in nanometers) for a spectral band being created.
	Undoing the last operation with an optical spectrum.

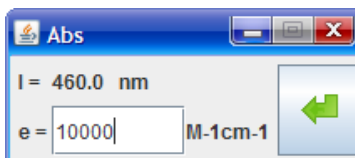
	Redoing the last undone operation.
	Solving the system of given differential equations with specified input parameters. Calculated kinetic curves are displayed in "Graph Panel".

1.3.2. Setting absorbance coefficients

Setting a necessary value for an absorbance coefficient at a certain wavelength for a selected substance X can be performed by dragging a corresponding point on a spectrum using the mouse while holding the mouse left button.







To set an exact value, double click the left mouse button on the point of a spectrum. This action will call a new window where you can input a desirable value and then press "Enter":



Using the latter interface with the "Create bands" option enabled will generate an absorbance band with a given half-width (defined by a slider "Gauss width / nm") and a height equal to the entered value. (for a tutorial see § 2.5.1)

1.3.3. Managing the view

Item	Function
	Zooming in along X or Y axes.
	Zooming out along X or Y axes.

	Rescaling to make all the spectra fit the display panel.
	Enables “Multiply mode”. In this mode, the displacement of one point of a spectrum results in a proportional shift of all other point. Enabling this mode turns on a label Multiply mode is ON! *

*“Multiply mode” allows one to convert data from absorbance–wavelength coordinates to ϵ –wavelength ones for absorbance spectra loaded from a file (see § 2.5.2).

1.4. Interface for automatic approximation of input parameters (Global fit)

The interface “Global fit” allows one to perform automatic fitting of multiple kinetic curves to find optimal input parameters within a given system of differential equations. Basic and advanced settings for each type of the fitting algorithms are available.

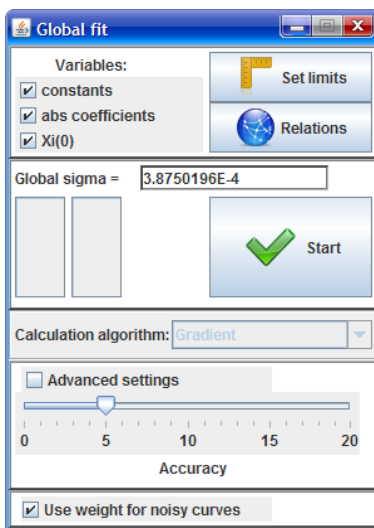




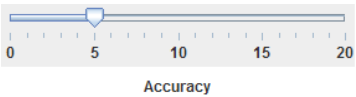


Fig. 1.8. Interface for automatic global fit of experimental data.

Item	Function
<div>Variables:</div> <input checked="" type="checkbox"/> constants <input checked="" type="checkbox"/> abs coefficients <input checked="" type="checkbox"/> Xi(0)	Panel to specify types of variable parameters.
	Opens “Set limits” interface to set ranges of variation for variable parameters (see § 1.6).
	Opens “Xi(0) Relations” interface window to create a set of equations defining correlations between the values of initial concentrations of the compounds X (see § 1.7).

<p>Global sigma = <input type="text" value="4.203568E-4"/></p>	<p>The value of the standard deviation (see § 3.2). Demonstrates goodness of fitting.</p>
	<p>Bars to display progress of fitting.</p>
	<p>Start an automatic approximation.</p>
<p>Calculation algorithm: <input type="text" value="Gradient"/></p>	<p>Computational algorithm selection: “Gradient” (short from “Gradient descent”) or “Direct” (see § 3.4 and § 3.5). Note that optimization with these algorithms is implemented only for rate constants and initial concentrations, but values of absorbance coefficients are found by solving a system of linear equations (see § 3.2, 3.3).</p>
	<p>Slider to set the accuracy of computations. This slider changes some parameters of computational algorithms (for more details see § 3.4 и § 3.5).</p>
<p><input checked="" type="checkbox"/> Use weight for noisy curves</p>	<p>When enabled, each experimental kinetic curve is assigned a certain statistic weight according to its “noisiness” (see § 2.2). The “noisier” a curve is the smaller statistic weight it receives, and the less it contributes to the determination of optimal rate constants of reactions (it’s worth noting that it has no effect on determination of absorbance coefficients values which are calculated separately for each wavelength).</p>
<p><input checked="" type="checkbox"/> Advanced settings</p>	<p>Calling the interface “Global fit” with advanced settings to configure calculation algorithms. (Fig. 1.9). Provides greater control over the process of fitting.</p>

1.5. Interface for automatic approximation of the input parameters with advanced settings (Global fit)

The program “SPARK” provides a user with two types of computational algorithms: “Gradient”(short from “Gradient descent”) and “Direct”. To avoid misunderstanding during the further reading, it is necessary to give a brief explanation about usage of algorithms.

Note that optimal values of absorbance coefficients are found by solving a system of differential equations (see § 3.3). This procedure is fast and simple and does not require any optimizational algorithms.

Computational algorithms “Gradient” and “Direct” are used only to search for optimal values of rate constants and initial concentrations. The two algorithms have some advantages and disadvantages, and their implementation is described in 3.4 and 3.5.

When more than one type of parameters are varied the program implements combined algorithms described in 3.6. Computational methods available depending on types of variable parameters are presented in the table below:

Absorbance coefficients	Rate constants	Initial concentration	Algorithms used
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Method in § 3.3
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Gradient descent see § 3.4
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Combined algorithms: “Gradient” (§ 3.6.1) or “Direct” (§ 3.6.2)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Combined algorithm in § 3.6.3
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

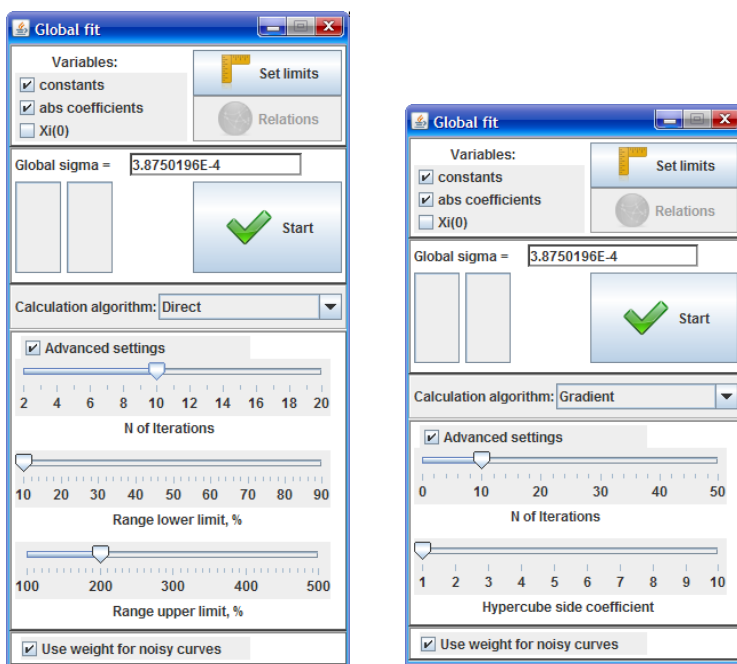

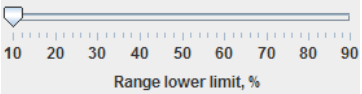
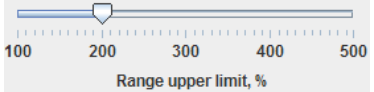

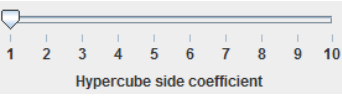


Fig.1.9. Interface for automatic global fit with advanced settings for “Direct”(left) and “Gradient”(right) algorithms.

Item	Function
	Number of iterations for the algorithm “Direct” (see § 3.5).
	Setting lower limit (in %) for all variable parameters for the algorithm “Direct” (see § 3.5).*
	Setting upper limit (in %) for all variable parameters for the algorithm “Direct” (see § 3.5).*
	Number of iterations for the algorithm “Gradient descent” (see § 3.4).



Defines how many times the starting edge of the hypercube in the grid is smaller than the initial value of a variable parameter in the algorithm “Gradient descent” (see § 3.4).

*Let A be the initial value of a parameter a_i . Provided range lower limit is set as N%, and the upper one is M%, then a_i will vary from $(A \times N)$ up to $(A \times M)$ value.

1.6. Interface for setting ranges of variation for input parameters (Set limits)

Interface “Set limits” allows one to specify ranges of variation for input parameters, which in some cases reduces probability of getting stuck on an undesirable local minimum and facilitates search of an optimal solution.

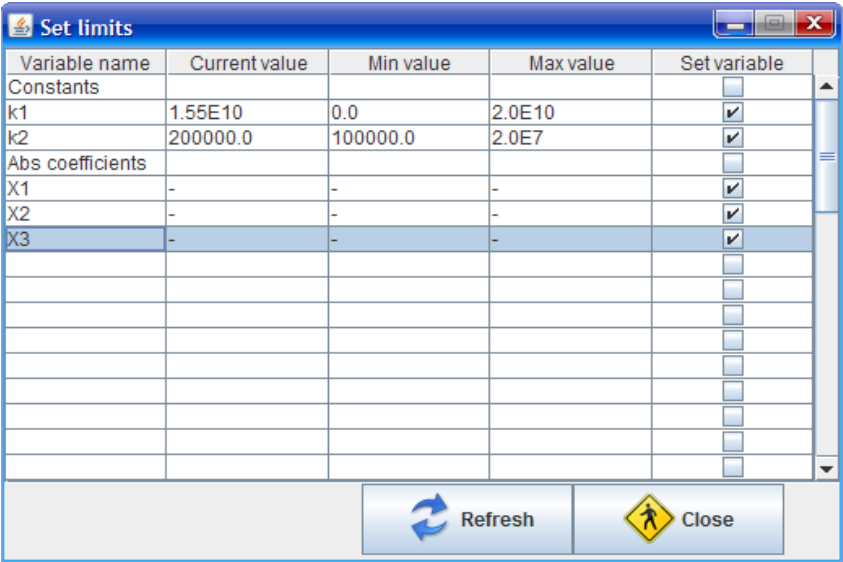




Fig.1.10. Interface for setting ranges of variation for parameters.

Columns description:

Name	Description
Variable name	The name of a variable parameter.
Current value	The current value of a variable parameter.
Min value	Minimal possible value of a parameter. Blank cell means no limits.
Max value	Maximal possible value of a parameter. Blank cell means no limits.
Set variable	Allows one to exclude a parameter for the list of variables (Set as constant).

Note 1: limits of variation can be set only for rate constants and initial concentrations.

Note 2: Sliders “Range lower limit, %” and “Range upper limit, %” for “Direct” algorithm (see § 1.5) set limits of variation for all variables, but the interface “Set limits” allows one to specify ranges for individual parameters.

Item	Function
	Refreshing(updating) data in the table (e.g. after changing values of some parameters).
	Closing the “Set limits” interface.

1.7. Interface for creating correlations between initial concentrations values (Xi(0) Relations)

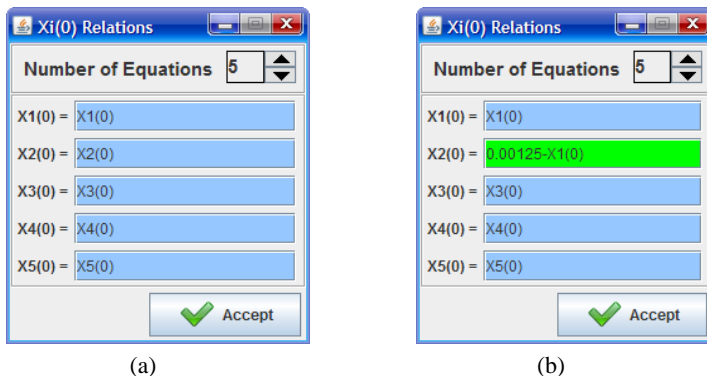
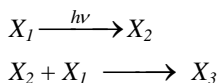


Fig.1.11. Interface for creating correlations between values of initial concentrations: a) default scheme of equations; b) modified scheme.

This interface allows one to create a set of equations to define correlations between values of initial concentrations for compounds X_i . Need for introduction of such equations is due to the fact that there often exist certain proportions between values of initial concentrations of components in a reacting mixture. For example, consider a scheme of chemical reactions:



Obviously, the initial concentration of the primary intermediate X_2 right after a laser pulse ($X_2(0)$) is bound to $X_1(0)$ (right after a laser pulse) concentration according the following equation:

$$X_2(0) = X_1 \text{ before pulse} - X_1(0)$$

Thus, varying $X_1(0)$ and $X_2(0)$ should be performed with preservation of the sum ($X_1(0) + X_2(0)$), which is equal to X_1 concentration before a laser pulse. Since the two parameters are correlated linearly, only one of them (either $X_1(0)$ or $X_2(0)$) should be varied and the other will be defined automatically. The principle described here helps to reduce computation time and speed up search for optimal solution.

Note: after creating a scheme of equations, click “Accept” button to save the changes.

2. Tutorial

2.1. Format of files

Program “SPARK” allows one to load experimental kinetic curves from various types of text files (*.txt, *.asc, *.csv etc.). The only requirement for a file structure is that it must consist of two columns of data: the first one being time values (for X axes) and the second one – optical density (for Y axes).

0.0	-0.0021946146	0.0000E+0000	1.7860E-0004
0.04882	6.7584147E-4	4.8828E-0002	1.7860E-0004
0.09765	8.587457E-4	9.7656E-0002	1.7860E-0004
0.1464	0.002168809	1.4648E-0001	1.7860E-0004
0.1953	3.1067908E-4	1.9531E-0001	1.7860E-0004
0.2441	-6.5937033E-4	2.4414E-0001	1.7860E-0004
0.2929	-0.0017896097	2.9297E-0001	1.7860E-0004
0.3418	0.0023231239	3.4180E-0001	1.7860E-0004
0.3906	-0.0010676915	3.9062E-0001	1.7860E-0004
0.4394	-0.0010913357	4.3945E-0001	1.7860E-0004
0.4882	6.132035E-4	4.8828E-0001	1.7860E-0004
0.5371	-7.1099075E-4	5.3711E-0001	1.7860E-0004
0.5859	-0.0013163395	5.8594E-0001	1.7860E-0004

(a) (b)

Fig. 2.1. Examples of file structures; a) *.txt, b) *.asc..

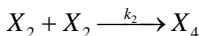
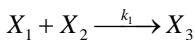
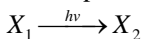
It should be noted that the presence of additional text lines (e.g. names of columns, date, comments etc.) causes no problem in reading the data. Any type of a separator between the two columns is acceptable.

Note: all loaded kinetic curves must have the same number of points.

2.2. Input format of differential equations

A system of differential equations must be typed into text fields located on the “Equation” panel under the “dXi/dt=” label. The number of text fields corresponds to the number of equations in the system. Only the right part (after the “=” sign) of each equation must be typed.

For example, the following scheme of chemical reactions



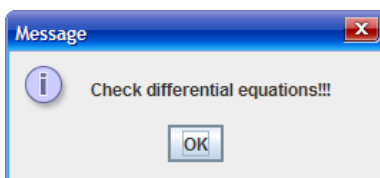
is described by a system of differential equations:

$$\begin{cases} \frac{dX_1}{dt} = -k_1 X_1 X_2 \\ \frac{dX_2}{dt} = -k_1 X_1 X_2 - 2k_2 X_2 X_2 \\ \frac{dX_3}{dt} = k_1 X_1 X_2 \\ \frac{dX_4}{dt} = k_2 X_2 X_2 \end{cases}$$

In order to introduce that system to the program you must type:

Equations
dXi/dt=
-k1*x1*x2
-k1*x1*x2-2*k2*x2*x2
k1*x1*x2
k2*x2*x2

Note that there is no difference between usage of letters in uppercase and lowercase. In case of improper input of equations the program gives a message:

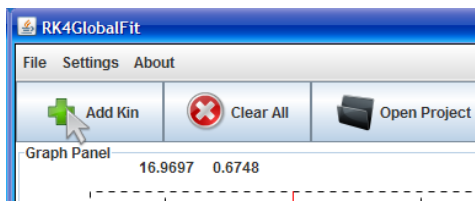


In case the initial concentration of one of the substances X_i doesn't change during reactions, it's acceptable to type " $X_i(0)$ " in an equation line (e.g., " $-k_1 * X_1(0) * X_2$ ").

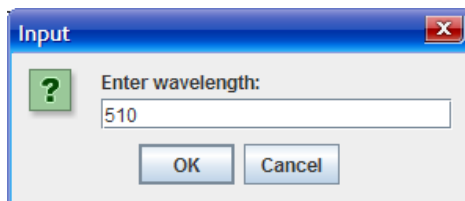
2.3. Fitting of loaded kinetic curves

This section provides a brief step by step tutorial about how to use this program to fit a single kinetic curve within a given system of differential equations. Fitting of multiple kinetic curves can be performed according to the same procedure.

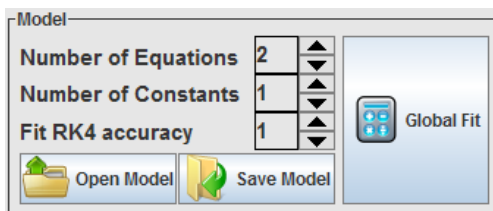
1. Load one or multiple kinetic curves from a file/files (e.g. *.asc or *.txt (see § 2.1)):



For each kinetic curve loaded the program automatically opens a window to input the corresponding wavelength. Note that it is acceptable to use any convenient units for wavelengths (e.g. nanometers, cm^{-1} etc), but the only requirement is to use the same units throughout the entire project. The program will sort the kinetic curves in a right order based on the values entered.



2. In the following panel specify the number of differential equations and rate constants:



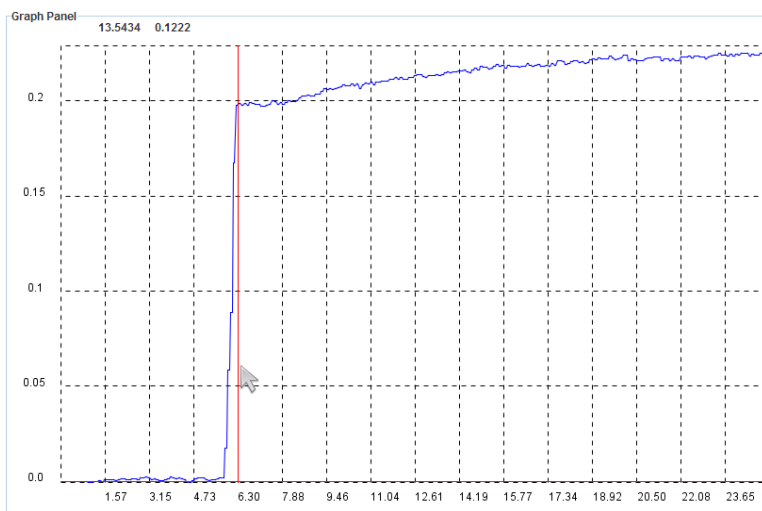
3. Input a system of differential equations (see § 2.2), initial concentrations right after a laser pulse ($X_i(0)$) and the concentrations before the pulse (X_i before pulse):

Equations		
$dX_i/dt=$	$X_i(0)$	X_i before pulse
$-k_1 \cdot x_1$	$1E-4$	0
$k_1 \cdot x_1$	0	0

4. Set an approximate value for a rate constant:

Constants	
$k_1=$	$1e5$

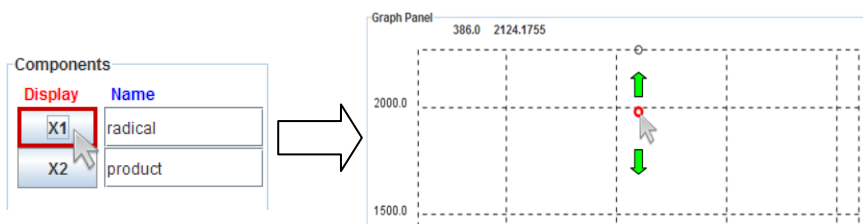
5. Dragging the red vertical line on the “Graph Panel” set the time moment corresponding to the laser flash:



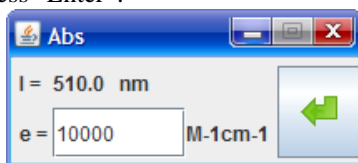
6. Clicking “Set Abs” button open “Set Abs” interface to set the values of absorbance coefficients:

Options	
Fit RK4	Set Abs
Global sigma = <input type="text" value="0"/>	
<input type="button" value="Show"/>	

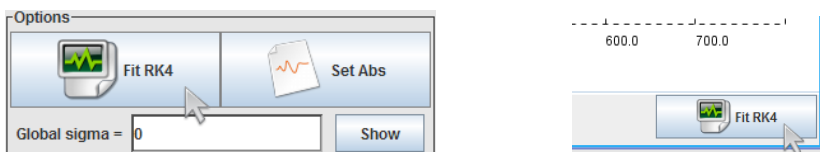
7. Select the first component (“X1”) and then set absorbance coefficient value by dragging a red point on the spectrum using a mouse while holding the left button. Similarly, set the values of absorbance coefficients for the other components.



To set a exact value for an absorbance coefficient, double-click the corresponding point on a spectrum and then in the popped up window type the desirable value. Then press “Enter”:



8. To perform calculation of the system of differential equations press “Fit RK4” button. You can use buttons located either on the “Option” panel or in the lower right corner of “Set Abs” interface. Calculated kinetic curves will immediately appear on the “Graph Panel”:



9. To achieve better fitting results try to adjust the values of input parameters and then press the “Fit RK4” button again. It’s recommended to focus on the “Error Panel” when performing manual adjustment of parameters. The “Error Panel” is designed to demonstrate deviation of a calculated kinetic curve from the corresponding experimental one at every time point available (Fig 2.2). The overall error is displayed as “Global sigma” parameter (see § 3.2):

Global sigma = 0.004653429

It’s worth noting that, though good fit means smaller values of “Global sigma”, it’s practically impossible to reduce the latter to zero. No matter how good experimental data is, it always has statistical errors (“noise”), which increases the global deviation (“Global sigma”) value.

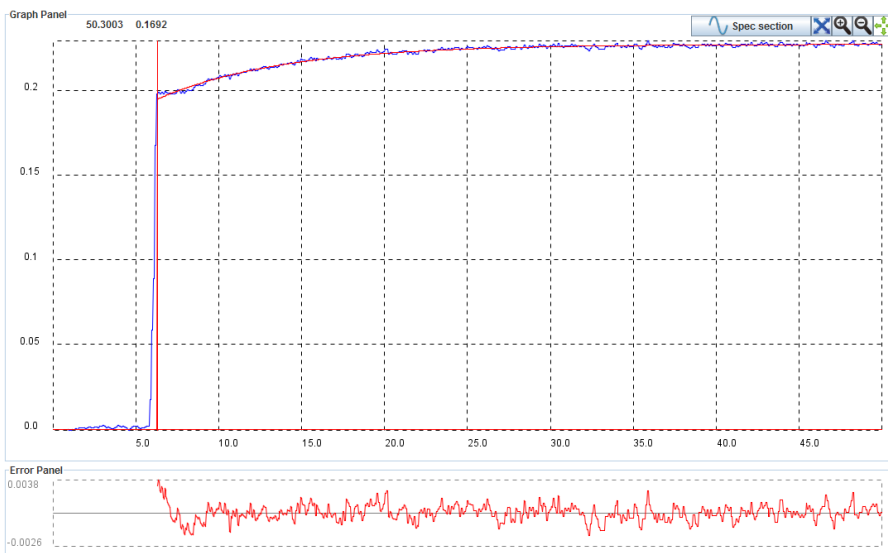
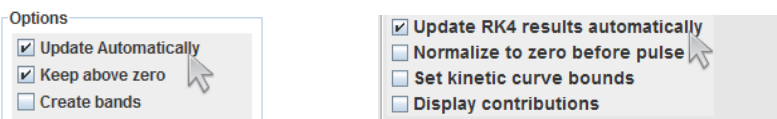


Fig. 2.2. A fitted kinetic curve. Errors for every time point are displayed on the “Error Panel” (bottom).

10. To speed up a fitting process it’s recommended to enable option “Update automatically”:

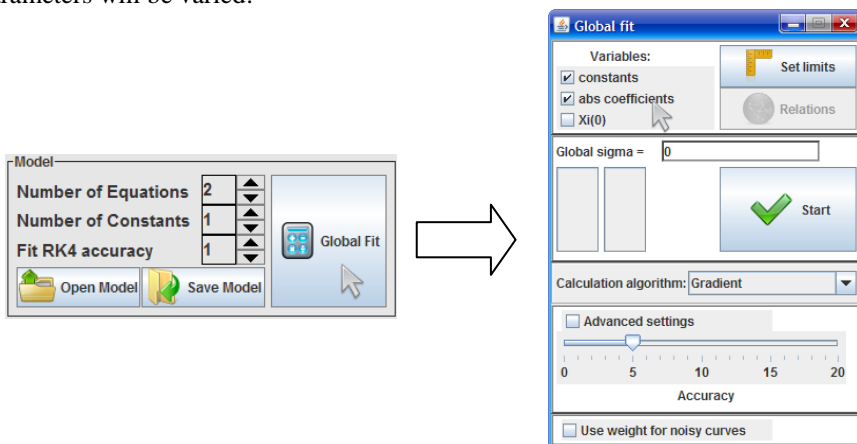


Now any change in values of rate constants (using arrow buttons) or absorbance coefficients (using the mouse) will result in immediate recalculation of the system of differential equations. New calculated kinetic curves will be displayed in the “Graph Panel”.

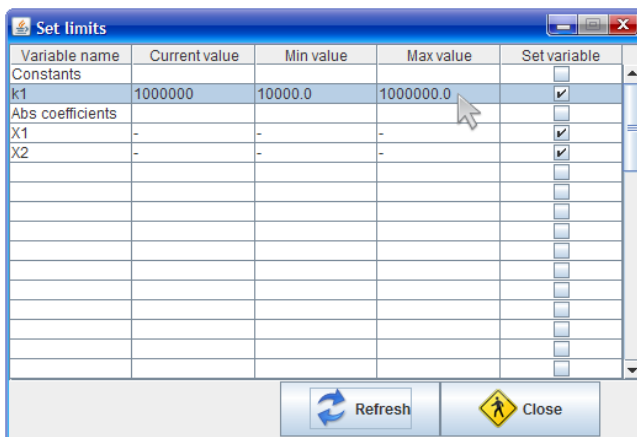
In case of multiple kinetic curves it’s reasonable to use the “Global fit” interface designed for automatic approximation of input parameters.

2.4. Automatic approximation of parameters

1. Open the “Global fit” interface window. Define which types of input parameters will be varied:



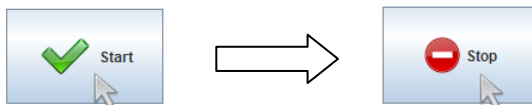
2. Press the “Set limits” button in the upper right corner. In the new widow input max and min limits of variation for parameters. To edit a cell, double click it with the left mouse button. To set some parameters as constants uncheck the corresponding checkboxes in the “Set variable” column. Close the “Set limits” window.



3. In the “Global fit” window set accuracy with the slider “Accuracy” and select a calculation algorithm. The method “Gradient” is set by default (see § 3.4). The alternative method is “Direct” (see § 3.5), which in some cases is more appropriate due to its ability to find the deepest minimum (global minimum) for a

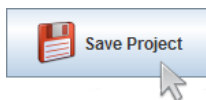
function studied. In fact, both methods have shown good results when tested on many arrays of experimental data. It's important to note that the best fitting strategy is to try both methods with several different sets of values for input parameters.

4. Press the “Start” button to let the program perform automatic fitting. The value of “Global sigma” and the calculated kinetic curves will be updated during the fitting process at regular intervals. You may press the “Stop” button if satisfied with the calculation results, or wait until the program finishes the fitting process.



5. In case the calculation result is poor and the further fitting makes no improvement of the “Global sigma” parameter (the program might get stuck on one of the local minima), it's recommended to alter initial values of parameters and restart the fitting.

6. When finished save the current working project:



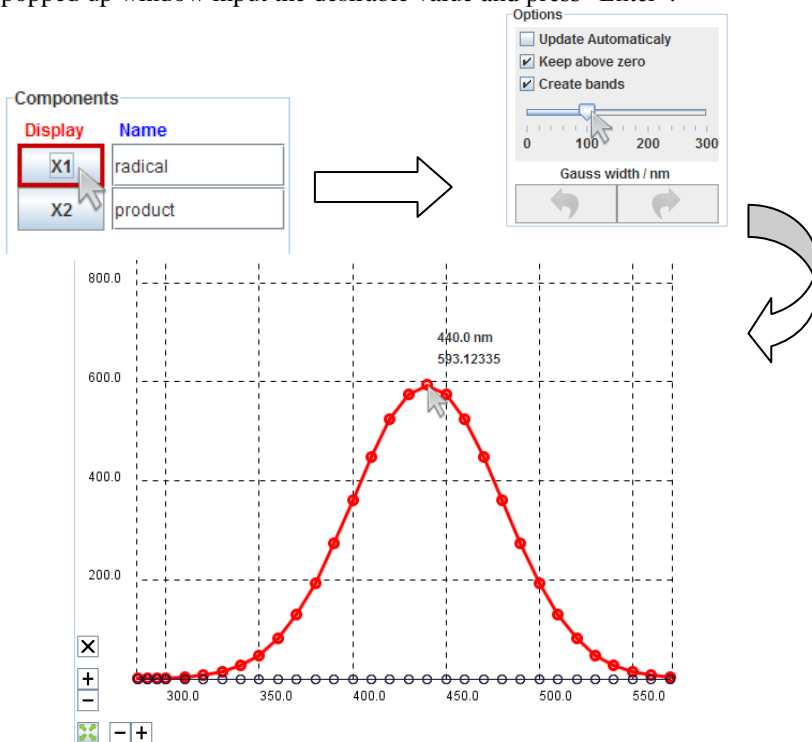
2.5. Managing absorbance spectra

2.5.1. Creating a new absorbance spectrum

This program allows one to create a new absorbance spectrum from scratch, which might be useful in some cases.

Press “Set Abs” button to open the interface window. Select a component whose spectrum needs to be generated. Enable “Create bands” option and using the slider “Gauss width / nm” set the band width at half-height. Place the mouse cursor over the point where the maximum of a new band should be located. Then, holding the mouse left button, drag the point to reach a desirable peak value. Change view scale, if necessary.

In order to create a band with an exact peak value, with the left mouse button double click the point corresponding to the maximum of a new band. Then in the popped up window input the desirable value and press “Enter”.



2.5.2. Loading optical spectrum from file

Format of files

This program allows one to load spectra from various types of text files with extensions *.txt, *.csv, *.asc etc. The only requirement is that a file must contain two data columns, the first one being wavelengths and the second – absorbance coefficient or optical density values.

300.0	15856.13	wavelength (nm);Absorbance (AU);Std.Dev.
310.0	6200.0	190;2.9762110710144;0.160382367054827
320.0	3200.0	191;3.20158576965332;0.289477900424168
330.0	2200.0	192;3.44964075088501;0.34747515472951
340.0	1513.7305	193;3.41220998764038;0.678361673848748
345.0	1433.9343	194;3.79843139648438;0.395747660684811
350.0	1406.166	195;3.26667881011963;0.454092111796818

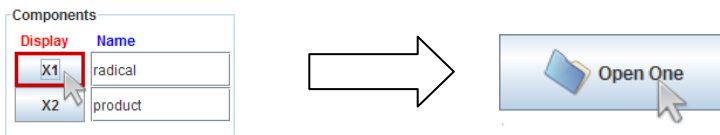
(a) (b)

Fig. 2.3. Examples of files with extension *.txt (a) and *.csv (b).

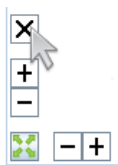
It should be noted that the presence of additional text lines (e.g. names of columns, date, comments etc.) causes no problem in reading the data. Any type of a separator between the two columns is acceptable.

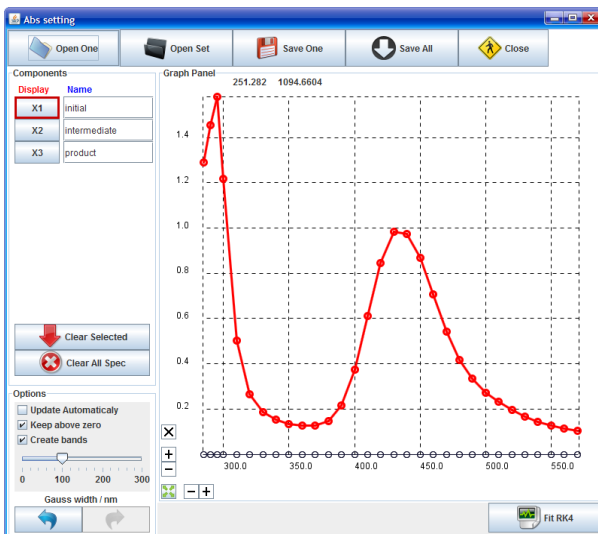
Loading optical spectrum

First of all, open the “Set Abs” interface window and select a component X, whose spectrum should be loaded. Press the “Open One” button and select a file. The absorbance spectrum of the selected component X will be created on the basis of the data in the file loaded. If the file does not contain information about some wavelength, its value will be set to zero automatically.



It often happens that a loaded file contains an optical density values, but not absorbance coefficients (Fig. 2.4). In this case it's recommended to use so called “Multiply mode”, that allows to multiply a selected spectrum to convert it from optical density values to absorbance coefficients. To enable this option press button with a sign “x”. Activation the “Multiply mode” will be indicated by a pink label «Multiply mode is on!» that appears on the bottom of the “Set Abs” interface window.





Puc. 2.4. Absorbance spectrum loaded from a file.

Then double-click any point on a selected spectrum, whose absorbance coefficient is known. The later action will call a new window to input the absorbance coefficient value. Having typed the value, press “Enter”. All other points of the spectrum will be changed accordingly.

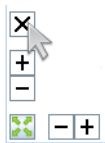
Abs

l = 430.0 nm

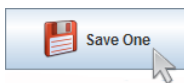
e = 13000 M-1cm-1

Green arrow button

Thus, the spectrum will be converted from (Optical density-wavelength) to (ϵ – wavelength) coordinates. Having finished converting, press the “x” button again to leave this mode



Use the button “Save One” to save the modified spectrum for reuse:



2.6. Tips about program usage

1) The initial values of input parameters are used by the program as the starting point in search for the best solution. In some cases, final fitting results might depend on initial conditions. For this reason, to achieve more adequate results it's recommended to set most expectable initial values and ranges of variation for variable parameters.

2) To avoid a situation when the program gets stuck on one of the local minima (the solution found is not the best one) it's recommended to use the following strategy. You should run an automatic approximation procedure with different initial values of input parameters using different calculation algorithms and compare the results.

Fitting results can be considered adequate, if fitting of experimental data with different sets of initial values of input parameters leads to the same output. In case experimental data can be similarly treated with two or more sets of parameters of different values, it's recommended to determine some of the parameters with experimental methods.

3) The value of the standard deviation ("Global sigma") characterizes the goodness of fitting results. Note that reaching less then **0.001** for this value is considered satisfactory.

4) The program allows one to vary the initial concentration of compounds together with the rate constants and absorbance coefficients. Introduction this third type of parameters gives the system more degree of freedom which in some cases makes the fitting results very ambiguous. Thus, in case of varying initial concentration it's recommended to limit ranges of variation for rate constants and initial concentrations, fix some absorbance coefficients (if possible), and use the "Xi(0) Relations" interface to define correlation between values of initial concentration.

5) There may be situations when a reaction produces two compounds with non-zero absorbance coefficients at one or more wavelengths ($A \rightarrow B + C$). In this case, there is an uncertainty about absorbance coefficients (see § 3.3), since only the sum ($\epsilon_B + \epsilon_C$) is defined. To avoid ambiguity it's recommended to fix some of the input parameters by unchecking corresponding checkboxes in the "Set limits" interface.

3. Description of algorithms

3.1. The 4th order Runge-Kutta method

The core of the program is the 4th order Runge-Kutta method for solving systems of differential equations [1]. The need to use this method is caused by the absence of analytical solution for the vast majority of problems of chemical kinetics. A brief description of this algorithm is presented below.

Consider one problem. Provided a given scheme of chemical reactions, involving compounds X_1, X_2, \dots, X_i , is described within a system of differential equations. The expressions for the first derivatives of functions $X_i(t)$ and the values of $X_i(t=0)$ are known:

$$\left\{ \begin{array}{l} X_i(t=0) = X_i(0), \\ \dots \\ X_i(t=0) = X_i(0), \\ \left\{ \begin{array}{l} \frac{dX_1(t)}{dt} = f_1(X_1(t), \dots, X_i(t)), \\ \dots \\ \frac{dX_i(t)}{dt} = f_i(X_1(t), \dots, X_i(t)), \end{array} \right. \end{array} \right.$$

Then the values of concentrations in every time points are calculated according to the recursive equations:

$$\left\{ \begin{array}{l} X_{1,n+1} = X_{1,n} + \frac{1}{6}(k_{11} + 2k_{21} + 2k_{31} + k_{41}) \\ \dots \\ X_{i,n+1} = X_{i,n} + \frac{1}{6}(k_{1i} + 2k_{2i} + 2k_{3i} + k_{4i}) \end{array} \right.$$

where k_{1i} , k_{2i} , k_{3i} , k_{4i} – are some constants for a compound X_i at n^{th} iteration. The values of these constants are calculated by the following steps:

$$\left\{ \begin{array}{l} k_{11} = \Delta t \cdot f_1(X_{1,n}, \dots, X_{i,n}) \\ \dots \\ k_{1i} = \Delta t \cdot f_i(X_{1,n}, \dots, X_{i,n}) \\ \left\{ \begin{array}{l} k_{21} = \Delta t \cdot f_1(X_{1,n} + \frac{1}{2}k_{11}, \dots, X_{i,n} + \frac{1}{2}k_{1i}) \\ \dots \\ k_{2i} = \Delta t \cdot f_i(X_{1,n} + \frac{1}{2}k_{11}, \dots, X_{i,n} + \frac{1}{2}k_{1i}) \end{array} \right. \end{array} \right.$$

$$\left\{ \begin{array}{l} k_{31} = \Delta t \cdot f_1(X_{1,n} + \frac{1}{2}k_{21}, \dots, X_{i,n} + \frac{1}{2}k_{2i}) \\ \dots \\ k_{3i} = \Delta t \cdot f_i(X_{1,n} + \frac{1}{2}k_{21}, \dots, X_{i,n} + \frac{1}{2}k_{2i}) \\ k_{41} = \Delta t \cdot f_1(X_{1,n} + k_{31}, \dots, X_{i,n} + k_{3i}) \\ \dots \\ k_{4i} = \Delta t \cdot f_i(X_{1,n} + k_{31}, \dots, X_{i,n} + k_{3i}) \end{array} \right.$$

where Δt — is a time step for one iteration. Thus, each successive value of $X_{i,n+j}$ is calculated on the basis of the preceding $X_{i,n}$ and constants k_{1i} , k_{2i} , k_{3i} , k_{4i} , taken with appropriate weights. Each constant represents a small increment of the value of X_i , calculated by multiplying the interval Δt by the local value of the derivative at a particular point: $\Delta X_i = \Delta t \times (dX_i/dt)$. These constants have the following mathematical meaning:

- k_1 — is a small increment, calculated based on the value of the derivative at the beginning of the interval at the point $X_{i,n}$.
- k_2 — is a small increment, calculated based on the value of the derivative inside of the interval at the point $X_{i,n} + \frac{1}{2}k_{1i}$.
- k_3 — is a small increment, calculated based on the value of the derivative also inside of the interval at the point $X_{i,n} + \frac{1}{2}k_{2i}$.
- k_4 — is a small increment, calculated based on the value of the derivative close to the end of the interval at the point $X_{i,n} + k_{3i}$.

During the averaging, increments calculated inside the interval are taken with larger weights:

$$\Delta t = \frac{1}{6}(k_{1i} + 2k_{2i} + 2k_{3i} + k_{4i})$$

In the 4th order Runge-Kutta method error at each step is about Δt^5 by an order of magnitude, and the total error over the entire range of integration — Δt^4 . This method allows one to calculate array of values representing dependence of concentration on time for all compounds X_i . The next problem is adjustment of input parameter to achieve minimal deviation of the array of calculated data from experimental values.

3.2. Selection of computational algorithms

Program “SPARK” allows one to perform automatic approximation of several types of parameters (rate constants, absorbance coefficients, and initial concentrations) within a given scheme of differential equations.

The main criterion during optimization of the above-listed parameters is the value of standard deviation (σ) of experimental data from the calculated values, which is determined according to the expression:

$$\sigma = \sum_{\lambda} \left[W(\lambda) \cdot \sqrt{\frac{1}{n} \sum_t (D(\lambda, t)_{\text{experiment}} - D(\lambda, t)_{\text{calculated}})^2} \right] / N,$$

where \sum_{λ} – the sum for all available wavelengths, \sum_t – the sum for all points in experimental curves at times t , n – number of points in the experimental curve within a fitting range (can be altered, see § 1.1.7), $D(\lambda, t)_{\text{experiment}}$ – the value of optical density of an experimental curve at wavelength λ at time t , $D(\lambda, t)_{\text{calculated}}$ – corresponding value of optical density for a calculated curve, $W(\lambda)$ – the weight of a curve (less “noisy” kinetic curves contribute to the sum more), N – number of experimental kinetic curves.

Thus, the parameter σ – is a function $\sigma = g(k_1, k_2 \dots k_n; \varepsilon_1(\lambda_1), \varepsilon_2(\lambda_1) \dots \varepsilon_n(\lambda_1)) \dots \varepsilon_1(\lambda_m), \varepsilon_2(\lambda_m) \dots \varepsilon_n(\lambda_m))$ depending on rate constants and absorbance coefficients of reaction participants (optimization of initial concentrations will be discussed later). Then the problem can be reduced to finding the minimum of this function. On the way to solve this problem we have to face two kinds of difficulties: the absence of analytical solution for the function and a large number of variable parameters (may reach several hundreds). Note that the absence of analytical solution makes it necessary to recalculate the system of differential equations for every change of input parameters (rate constants or initial concentrations), which involves considerable computation. For this reason, the implementation of global optimization requires an algorithm able to find best solution in a relatively small number of iterations.

To simplify the optimization procedure it has been suggested to implement independent variation of individual groups of input parameters [2, 3]. This principle is clearly illustrated in Fig. 3.1, where a ball rolls down a curved surface, consistently reaching minima along the first and then the second axes.

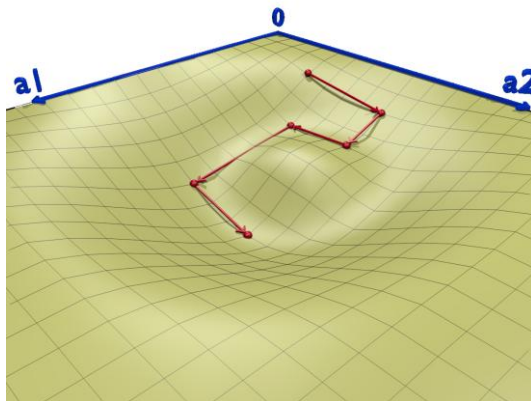


Fig. 3.1. The principle of sequential optimization.

Thus, we hypothesized that the search for the minimum value of $\sigma = g(k_1, k_2 \dots k_n; \varepsilon_1(\lambda_1), \varepsilon_2(\lambda_1) \dots \varepsilon_n(\lambda_1)) \dots \varepsilon_1(\lambda_m), \varepsilon_2(\lambda_m) \dots \varepsilon_n(\lambda_m))$ can be achieved through the consistent optimization of functions that depend only on one type of parameters:

$$\begin{aligned} \sigma &= f_k(k_1, k_2 \dots k_n), \\ \sigma(\lambda_1) &= f_{\lambda}(\varepsilon_1(\lambda_1), \varepsilon_2(\lambda_1) \dots \varepsilon_n(\lambda_1)), \\ &\dots \\ \sigma(\lambda_n) &= f_{\lambda}(\varepsilon_1(\lambda_n), \varepsilon_2(\lambda_n) \dots \varepsilon_n(\lambda_n)) \end{aligned}$$

Note that the search for optimal parameters of the function $\sigma(\lambda_i) = f_{\lambda}(\varepsilon_1(\lambda_i), \varepsilon_2(\lambda_i) \dots \varepsilon_n(\lambda_i))$ can be reduced to solving a system of linear equations (see § 3.3), which greatly speeds up the minimization of the global function.

Thus, it is necessary to minimize only the function $\sigma = f_k(k_1, k_2 \dots k_n)$. Choice of an algorithm of optimization of the function $\sigma = f_k(k_1, k_2 \dots k_n)$ must be carried out on the basis of information about its behavior (continuity, possible number of minima). To determine possible numbers of minima for the function $\sigma = f_k(k_1, k_2 \dots k_n)$, several kinetic schemes have been studied (see below). It was shown that, depending on the kinetic scheme, this function can have one or more minima.

The first example is a scheme consisting of only one monomolecular reaction:

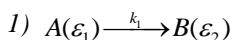


Fig. 3.2 represents dependences of the parameter σ on the rate constant (Fig. 3.2a) and the absorbance coefficients (Fig. 3.3b), reflecting behavior of functions $\sigma = f_k(k_1)$ and $\sigma(\lambda) = f_\lambda(\varepsilon_1(\lambda), \varepsilon_2(\lambda))$, respectively. Note that the absorbance coefficients varied for a single wavelength at a fixed rate constant. The rate constant, in turn, was altered at fixed values of the absorbance coefficients.

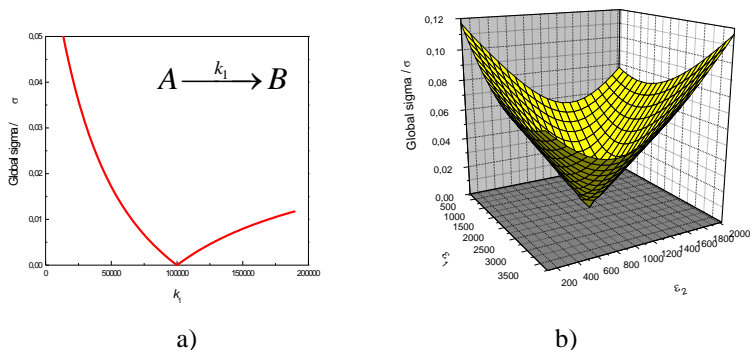
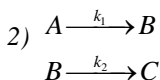


Fig. 3.2. a) dependence of the parameter σ on the rate constant k_1 ;
b) dependence of σ on the absorbance coefficients of the reagent and the product of the scheme (1).

The both functions have one minimum. Analysis of the behavior of the function $\sigma(\lambda) = g(k_1, \varepsilon_1(\lambda), \varepsilon_2(\lambda))$ also showed the existence of one minimum.

Nets, let's consider a scheme consisting of two parallel monomolecular chemical reactions:



Detailed analysis revealed the existence of more than one minimum for a functions $\sigma = f_k(k_1, k_2)$ and $\sigma(\lambda) = g(k_1, k_2, \varepsilon_1(\lambda), \varepsilon_2(\lambda), \varepsilon_3(\lambda), \varepsilon_4(\lambda))$.

Thus, it was shown that the optimized function $\sigma = f_k(k_1, k_2, \dots, k_n)$ may have more than one minimum. For this reason, it is necessary to use an algorithm capable to find the deepest minimum among all possible ones located on a multi-dimensional surface of parameters. For this purpose we have implemented the algorithm "Direct, whose principle is described below (see § 3.5).

It should be noted that for many schemes of chemical reactions, there is only one set of optimal values of input parameters, which corresponds to the presence of only one minimum for the function $\sigma = f_k(k_1, k_2, \dots, k_n)$. For this reason, program "SPARK" provides the user with an alternative, faster method "Gradient descent" (see § 3.4).

3.3. Calculation of absorbance coefficients

It is shown below that the optimization of a function $\sigma(\lambda) = f_{\lambda}(\varepsilon_1(\lambda), \varepsilon_2(\lambda) \dots \varepsilon_n(\lambda))$ can be reduced to solving a system of linear equations. This function can be conveniently presented as:

$$\sigma(\lambda)^2 = \frac{1}{n} \sum_t (D_{exper}(\lambda, t) - D_{calc}(\lambda, t))^2 = \frac{1}{n} \sum_t (\varepsilon_1 \cdot X_1(t) + \dots + \varepsilon_n \cdot X_n(t) - D_{calc}(\lambda, t))^2$$

A necessary condition for the minimum is equality to zero of the first derivatives of all parameters:

$$\left\{ \begin{array}{l} \frac{d\sigma^2}{d\varepsilon_1} = \frac{2}{n} \sum_t (\varepsilon_1 \cdot X_1(t) + \dots + \varepsilon_n \cdot X_n(t) - D_{calc}(\lambda, t)) \cdot X_1(t) = 0 \\ \dots \\ \frac{d\sigma^2}{d\varepsilon_n} = \frac{2}{n} \sum_t (\varepsilon_1 \cdot X_1(t) + \dots + \varepsilon_n \cdot X_n(t) - D_{calc}(\lambda, t)) \cdot X_n(t) = 0 \end{array} \right.$$

Opening the brackets in sums leads to:

$$\left\{ \begin{array}{l} \varepsilon_1 \sum_t X_1(t) \cdot X_1(t) + \dots + \varepsilon_n \sum_t X_n(t) \cdot X_1(t) - \sum_t D_{calc}(\lambda, t) \cdot X_1(t) = 0 \\ \dots \\ \varepsilon_1 \sum_t X_1(t) \cdot X_n(t) + \dots + \varepsilon_n \sum_t X_n(t) \cdot X_n(t) - \sum_t D_{calc}(\lambda, t) \cdot X_n(t) = 0 \end{array} \right.$$

That can also be represented in a matrix form:

$$\begin{pmatrix} \sum_t X_1(t) \cdot X_1(t) & \dots & \sum_t X_1(t) \cdot X_n(t) \\ \dots & \dots & \dots \\ \sum_t X_n(t) \cdot X_1(t) & \dots & \sum_t X_n(t) \cdot X_n(t) \end{pmatrix} \begin{pmatrix} \varepsilon_1 \\ \dots \\ \varepsilon_n \end{pmatrix} = \begin{pmatrix} \sum_t D_{calc}(\lambda, t) \cdot X_1(t) \\ \dots \\ \sum_t D_{calc}(\lambda, t) \cdot X_n(t) \end{pmatrix}$$

Renaming the matrixes we obtain:

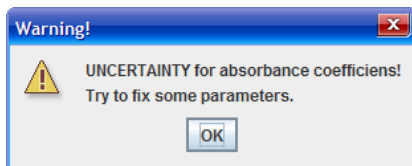
$$A \times E = K$$

Then the vector of solution can be found by inverting the corresponding matrix:

$$E = A^{-1}K$$

Thus, if there exists an inverse matrix A^{-1} (the determinant of A is not equal to 0), the optimal values of absorbance coefficients are defined unambiguously for a given set of rate constants. In case the determinant of A equals to zero, the

solution cannot be found due to uncertainty. Such situation may arise when a reaction produces two compounds with non-zero absorbance coefficients at one or more wavelengths (e.g., $A \rightarrow B + C$). In this case the calculation cannot be performed, and the program displays a message:



Note that the uncertainty can be avoided by fixing some values of absorbance coefficients.

3.4. Algorithm «Gradient descent»

One of the fastest algorithms for finding the minimum value of standard deviation function (Global sigma) in multidimensional space of variable parameters is a method based on the principle of stepwise descent towards the bottom of pit [4]. Consider an example. Suppose there are two variable parameters a_1 and a_2 . At the initial stage it's necessary to determine the motion vector directed to the minimum value. In order to do that we create a grid of 3×3 units, width being a_1 and height - a_2 , and center at (a_1, a_2) (Fig. 3.3a).

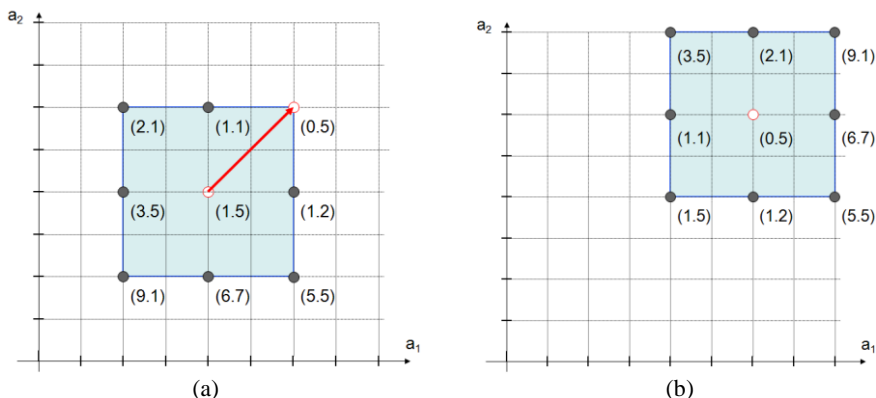


Fig. 3.3. The first (a) and second (b) iterations. In parentheses indicated values of standard deviation.

Based on the values of standard deviation we move the center of the grid to a new position. Again create a grid of the same size and determine the value of the parameter σ in each of the nodes (Fig. 3.3b). The failure to find any better value for the parameter σ indicates the necessity to reduce the size of the grid and continue the search.

Sequential execution of several iterations with a gradual decrease of the grid size is demonstrated in Fig. 3.4 a, b. Note that the accuracy of optimal value determination is proportional to the number of iterations.

The program interface allows one to adjust some parameters of the algorithm described. The initial size of the grid is set using the slider “Hypercube side” (see § 1.5.), which alters the coefficient indicating how many times a starting size of an edge of the grid is smaller then the initial value of the corresponding parameter. The number of iterations is governed by the slider “N of iterations”.

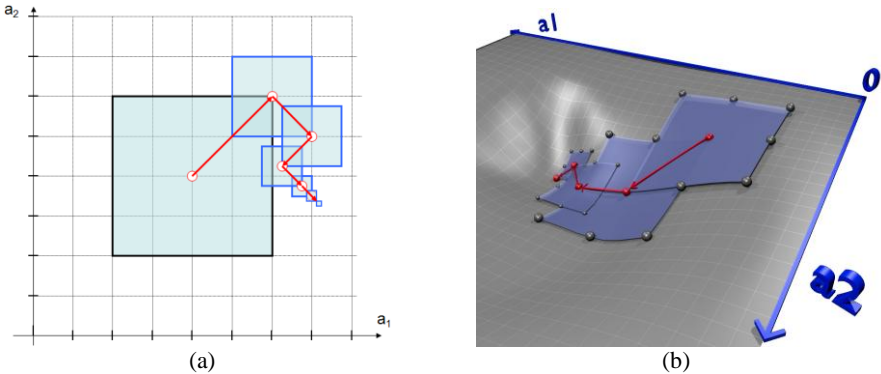


Fig. 3.4. Sequential execution of several iterations with a gradual reduction of the grid size.

The algorithm described above demonstrates good speed characteristics, but has a significant drawback: a quick descent to the nearest minimum value does not guarantee finding the deepest one. This algorithm is most appropriate for solving problems for which it is known that the function investigated has only one minimum.

3.5. Algorithm «Direct»

Algorithm “Direct” (short for "Dividing rectangles"), first proposed in 1993 by D.R. Jones and others [5], is a modification of the standard Lipschitzian approach [6-9] that eliminates the need to specify a Lipschitz constant. Note that the Lipschitz constant is viewed as a weighting parameter that indicates how much emphasis to place on global versus local search. No need to specify a Lipschitz constant makes it easier to find the best solution, since the value of this constant is usually unknown. Note that too big values of the constant lead to slow convergence, while small values increase probability of getting stuck on one of the local minima. The advantage of this algorithm is ability to carry out simultaneous searches using all possible values of the constants and therefore to operate at both the global and local levels.

Description of the algorithm “Direct” in case of two variable parameters is presented below. Suppose the two parameters a_1 and a_2 have their initial values

and ranges of variation defined (Fig. 3.5). The ranges of variation are normalized to vary from 0 to 1 to form a square with a side 1 (in a general case for N parameters – N - dimensional hypercube with a side 1). The search for the best value is performed by dividing the original hypercube into smaller ones with determination of the value of σ (Global sigma) at the center of each new hyperrectangle. Obviously, regardless of the dimension of a hyperrectangle it has only one center point. The absence of need to calculate the value of the function at every vertex of the hyperrectangle significantly reduces the computation time.

One iteration of the algorithm “Direct” is carried out through the following steps: identification of potentially optimal rectangles, sampling the found rectangles, and dividing them into smaller ones.

The first step is to find potentially optimal rectangles/cubes, dividing which into smaller ones will lead to detection of better values of σ . Since there is only one square it is selected for further sampling, which is performed by calculating values of the function $\sigma = f(a_1, a_2)$ at some points around the center (with offset of 1/3 of the length of the cube side) along all the axis available (Fig. 3.5).

The last step is to divide the square into smaller areas. Note that the fragmentation of the square is performed stepwise along all the axis with division by 3. Fragmentation of a rectangle is made only along the long sides to ensure a uniform reduction. As shown in [5], uniform reduction in all directions is necessary to achieve convergence of the search. In “Direct” it is suggested to start partition in the direction containing better values of the function. As a result, better values will be enclosed in rectangle with greater areas.

At the beginning of the second iteration it is necessary to determine potentially optimal rectangles for further sampling and division. It should be noted that a rectangle is considered potentially optimal if the following condition is satisfied:

Let $\varepsilon > 0$ be a positive constant and f_{\min} – the current best value of the function. A hyperrectangle i is said to be potentially optimal if there exists some $K > 0$ such that

$$f(c_i) - K \cdot d_i \leq f(c_j) - K \cdot d_j \text{ для } \forall j, \text{ и}$$

$$f(c_i) - K \cdot d_i \leq f_{\min} - \varepsilon \cdot |f_{\min}|,$$

where c_i is the center of a hyperrectangle i , d_i – distance from the center to its vertices. Implementation of these two conditions are illustrated in Fig. 3.6, which shows dependence of values of the function for a set of rectangles on the distance from their centers to vertices. Rectangles that in the next iteration will be determined as potentially optimal correspond to the points lying on the lower convex hull (solid red line).

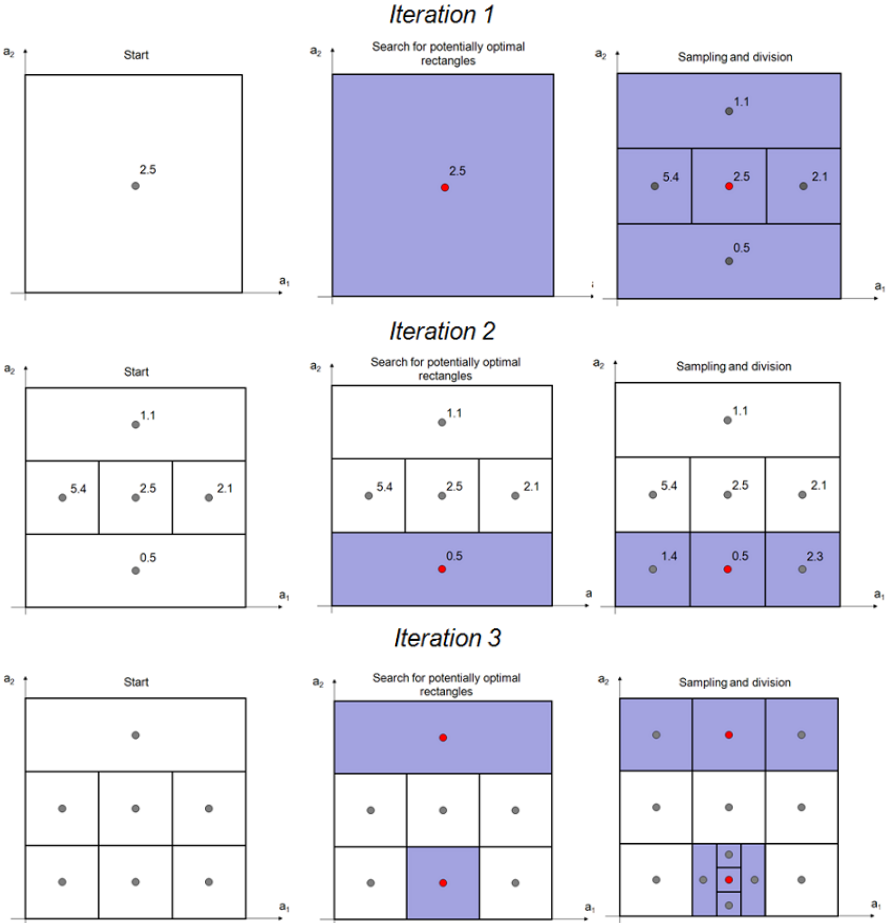


Fig. 3.5. Visualization of 3 successive iterations of the algorithm “Direct”.

Note that the parameter ε establishes balance between global and local search. While providing a uniform search over the entire range of variation for all the parameters, it prevents the program to waste precious time chasing for minor improvement of the function (getting into a local minimum). According to [5] the recommended range for ε is from 10^{-7} to 10^{-2} . In program “SPARK” the value of ε is set to 10^{-5} .

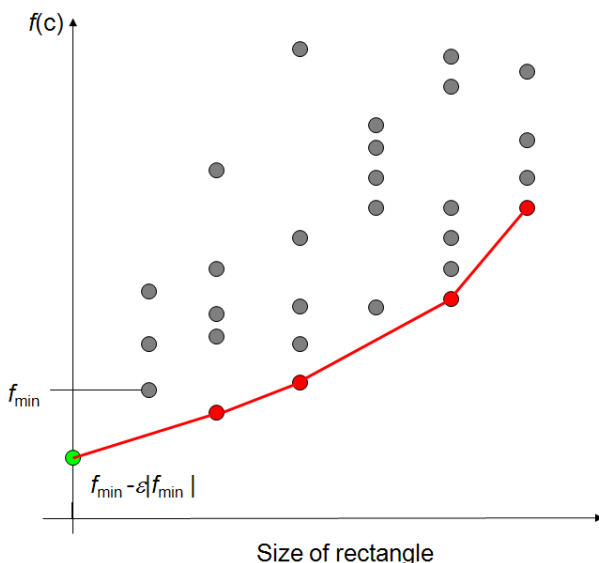


Fig. 3.6. Dependence of value of the function on size of the rectangles, obtained after several iterations. Red points connected by a red line correspond to the potentially optimal rectangles.

The algorithm terminates after implementation of a specified number of iterations. The program allows the user to set upper and lower limits for ranges of variation for parameters using sliders “Range low limit, %” and “Range high limit, %”, respectively (see § 1.5), as well as the number of iterations (slider “N of Iterations”). When using basic settings (see § 1.4), the slider “Accuracy” only changes the number of iterations, leaving the upper and lower limits for ranges of variation set by default: 10 and 190 %, respectively, for each parameter.

The main advantage of this algorithm is the ability to find a global minimum for a multi-extremal functions. Note that the algorithm “Direct” is slightly more time-consuming than the above mentioned “Gradient descent”. The second drawback of the algorithm “Direct” is limitation by the specified range of variation which results in disability to find a global minimum outside of that range.

It is shown below that the most efficient fitting of problems of chemical kinetics can be achieved by combining the method for calculating absorbance coefficients (see § 3.3) with algorithms “Gradient descent” and “Direct”.

3.6. Combined algorithms

It should be noted that, depending on the type of variable parameters one must use different modifications and combinations of the algorithms described. Optimization of different types of parameters has some specifics (see the table in § 1.5):

1) Search for optimal values of absorbance coefficients is carried out separately for each wavelength according to the algorithm described in section 3.3.

2) Search for optimal values of rate constants is implemented by the algorithms “Gradient descent” (3.4) or “Direct” (3.5) based on the value of the standard deviation (Global sigma) for the entire array of kinetic curves. This is due to the fact that a change in a rate constant leads to a change in the parameter σ for all wavelengths.

3) Search for optimal values of initial concentrations is often problematic due to very high degree of freedom, leading to ambiguity of fitting results. In this program it is implemented using the algorithm “Gradient descent” in conjunction with optimization of rate constants or absorbance coefficients.

In § 3.2 it was shown that in case of several types of variable parameters the optimization is performed in a sequential manner, independently for each type. For this reason the program “SPARK” provides several types of combined algorithms described below.

3.6.1. Combined algorithm «Gradient»

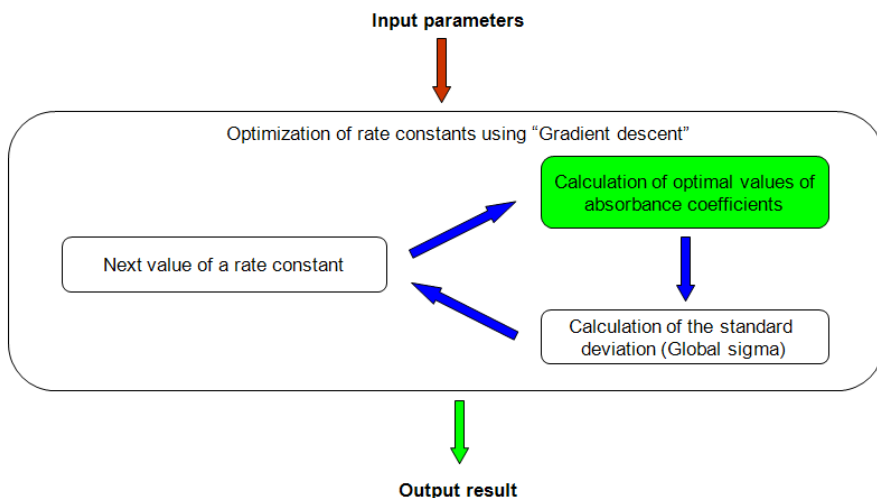


Fig. 3.7. Block diagram of the combined algorithm “Gradient” to optimize values of rate constants and absorbance coefficients.

Block diagram of the implementation of the combined algorithm “Gradient” is shown in Fig. 3.7. This algorithm is a modified method to find optimal rate constants using “Gradient descent” with imbedded calculation of optimal values of absorbance coefficients described in 3.3. For each set of rate constants program rapidly calculates absorbance coefficients, using algorithm in 3.3, and determines the value of the standard deviation. Despite the fact that the existence of multiple local minima may hinder search for the best solution using this algorithm, in practice such problems are rare.

3.6.2. Combined algorithm «Direct»

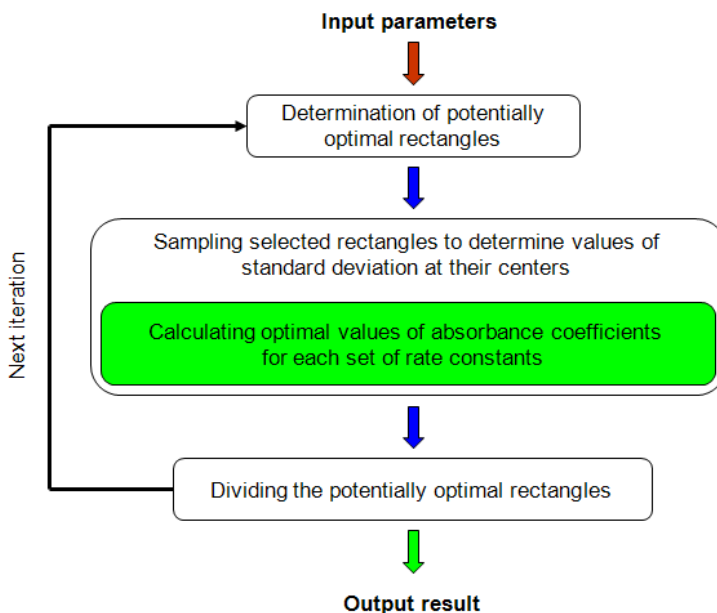


Fig. 3.8. Block diagram of the combined algorithm “Direct” to optimize values of rate constants and absorbance coefficients.

Block diagram of the implementation of the combined algorithm “Direct” is shown in Fig. 3.8. This algorithm is a modified method to find best values of rate constants using “Direct” (see § 3.5) with embedded calculation of absorbance coefficients using method described in 3.3. For each set of rate constants the program rapidly calculates absorbance coefficients and determines the value of the standard deviation. This algorithm is capable to find the deepest (global) minimum on a multi-dimensional surface, which might be essential for some problems. In case of a significant number of parameters this algorithm is much slower than the combined method “Gradient” (3.6.1).

3.6.3. Algorithm for optimization of initial concentrations values

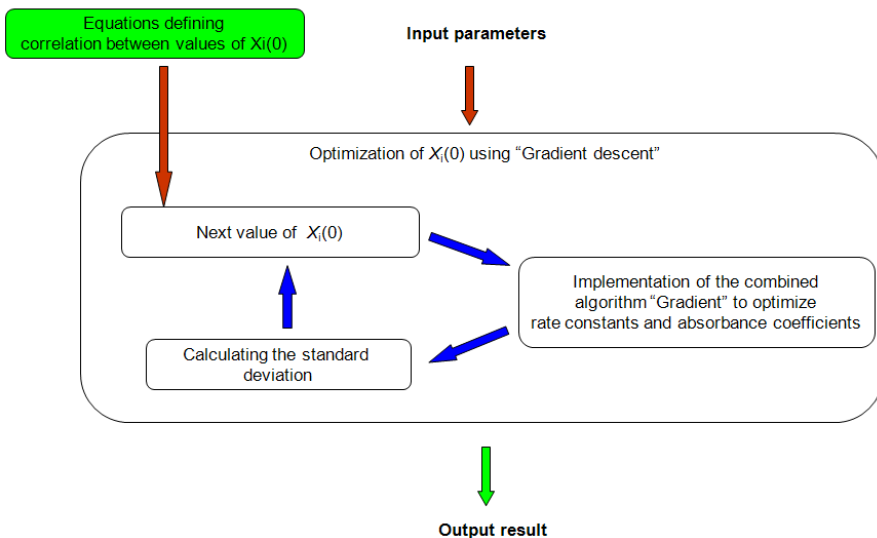


Fig. 3.9. Block diagram of the algorithm to optimize the values of initial concentrations, rate constants and absorbance coefficients.

Fig 3.9 demonstrates a diagram of the implementation of the algorithm to optimize values of $X_i(0)$. This algorithm is a modified method "Gradient descent" for finding optimal values of initial concentrations. Note that for each set of initial concentrations the program implements the combined algorithm "Gradient" (see § 3.6.1) to find optimal values of rate constants and absorbance coefficients, and then calculates the value of the standard deviation (Global sigma). The variation of $X_i(0)$ values is performed taking into account the equations defined in the interface "Xi(0) Relations" (see § 1.7). To avoid fitting problems see § 2.6, tip 4.

3.7. Error calculation

Let a value σ_{\min} be the minimum found for a function $\sigma = f(p, d)$ for some set of variables. It is known that the degree of freedom for the fitting is determined as $N - M$, where N – number of points on the experimental curve, and M – number of variable parameters. Then uncertainty of the parameter Global sigma is determined according to the expression:

$$\Delta\sigma = \frac{\sigma_{\min}}{\sqrt{N - M}}$$

The uncertainty of the parameters σ and error for p are correlated as:

$$\sigma_{\min} + \Delta\sigma = f(p_{\min} + \Delta p)$$

Since the analytical form of the function is unknown, it is necessary to solve the inverse problem by geometric methods. Fig. 3.10 demonstrates the method to calculate the error of the parameter p . After several iterations, points p_1 and p_2 are determined in such a way that the corresponding values of Global sigma (σ_1 and σ_2) are located to the left and to the right from $(\sigma_{\min} + \Delta\sigma)$. Then the unknown point p_x is determined by the following equation:

$$p_x = p_1 + (p_2 - p_1) \frac{\sigma_{\min} + \Delta\sigma - \sigma_1}{\sigma_2 - \sigma_1}; \quad \Delta p = p_x - p_{\min}$$

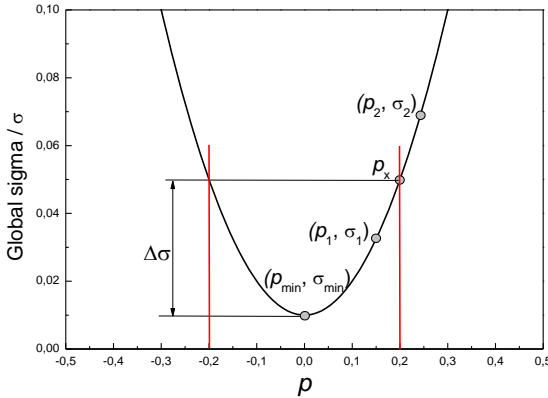


Fig. 3.10. Calculation of the error for the parameter p .

Conclusion

The program “SPARK” implements a number of algorithms to solve non-trivial problems of photochemistry and chemical kinetics. The main goal of these methods is to optimize input parameters of a system of differential equations in order to minimize deviation of experimental data from the calculated values.

The most challenging problems in implementing these methods were: the absence of analytical form of functions studied and a large number of variable parameters. To facilitate the search we have made an assumption about possibility to find best solutions by means of independent optimization of individual groups of input parameters.

This software has been successfully tested on many arrays of experimental data.

References

1. Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., Numerical Recipes. The Art of Scientific Computing, 3rd Edition, *New York: Cambridge University Press*, **2007**, p. 907.
2. Powell M. J. D., An efficient method for finding the minimum of a function of several variables without calculating derivatives, *Computer Journal*, **1964**, V. 7, pp. 155–162.
3. Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., Numerical Recipes. The Art of Scientific Computing, 3rd Edition, *New York: Cambridge University Press*, **2007**, p. 509.
4. Snyman J. A., Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms, *Springer Publishing*, **2005**, [ISBN 0-387-24348-8](https://doi.org/10.1007/978-1-4020-2861-3)
5. Jones D. R., Perttunen C. D., Stuckman B. E., Lipschitzian Optimization without the Lipschitz Constant, *Journal of Optimization theory and application*, **1993**, V. 79, N. 1, pp. 157-181.
6. Shubert B., A Sequential Method Seeking the Global Maximum of a Function, *SIAM Journal on Numerical Analysis*, **1972**, V. 9, pp. 397-388.
7. Pinter J., Globally Convergent Methods for n-Dimensional Multiextremal Optimization, *Optimization*, **1986**, V. 17, pp. 187-202.
8. Horst R., Tuy H., On the Convergence of Global Methods in Multiextremal Optimization, *Journal of Optimization Theory and Applications*, **1987**, V. 54, pp. 253-271.
9. Galperin E., The Cubic Algorithm, *Journal of Mathematical Analysis and Applications*, **1985**, V. 112, pp. 635-640.