Matlab tool “Optimizer”: Construction and Optimization of Multi-Block Mathematical Models

Application to spectroscopy experiments with ultracold gases of alkali metals

Vladimir Sovkov[1,2,*] and Jie Ma[2,3]

[1]St. Petersburg State University, 7/9 Universitetskaya nab., St. Petersburg 199034, Russia
[2]Institute of Laser spectroscopy, Shanxi University, Taiyuan, Shanxi 030006, China
[3]Collaborative Innovation Center of Extreme Optics, Shanxi University, Taiyuan 030006, China
*Corresponding author

Abstract—**The computational tool “Optimizer” written in Matlab is presented. This program is aimed to work with complicated multivariate mathematical models by constructing them from simpler blocks, presented by program routines with predefined interface, and optimizing the resulting multi-block model. The optimization is done with the linear and nonlinear Least Square Fit (LSF) using the Levenberg-Marquardt algorithm based on the Singular Value Decomposition (SVD) of the design matrix; extra tools include robust estimators, Tikhonov regularization, constrained optimization, etc. The program has been widely used for analysis of experimental spectroscopy data; the examples include spectra of ultracold gases of alkali metals.**

Keywords-multivariate optimization; mathematical model construction; spectra of ultracold gases of alkali metals

I. Introduction

The optimization of mathematical models via fitting them to experimental data is, probably, the most frequently encountered problem in natural sciences, technology, industry, economics, social sciences. A lot of efficient algorithms and computational tools have been developed to solve this problem [1–4]. Most of such tools require the user to supply a unique modeling program with a predefined program interface. This implies re-writing such a program for every new model, and makes it somehow difficult to apply such a tool to a complicated multivariate model with many outputs.

For example, a set of experimental spectroscopic data can include [5–9] term values of a molecule in various electronic and ro-vibronic states, intensities of bound–bound, bound–free, and free–bound transitions, complementary *a priory* data from *ab initio* calculations, etc. The parameters of this model can be the adiabatic potential functions, the functions of dipole transitions, the nonadiabatic matrix elements, some technical parameters of the experiment. Modeling of every subset of those data requires quantum–mechanical computations with different algorithms. On the one hand, a separate independent analysis of every subset of the data would be hardly enough—only a simultaneous self–consistent analysis of the entire set, as a rule, is able to convey a reliable information on the physical and chemical processes behind the experimental observations. On the other hand, modeling of a separate subset can be usually done with some standardized algorithm, so that the entire model can be treated as a combination of such simpler models.

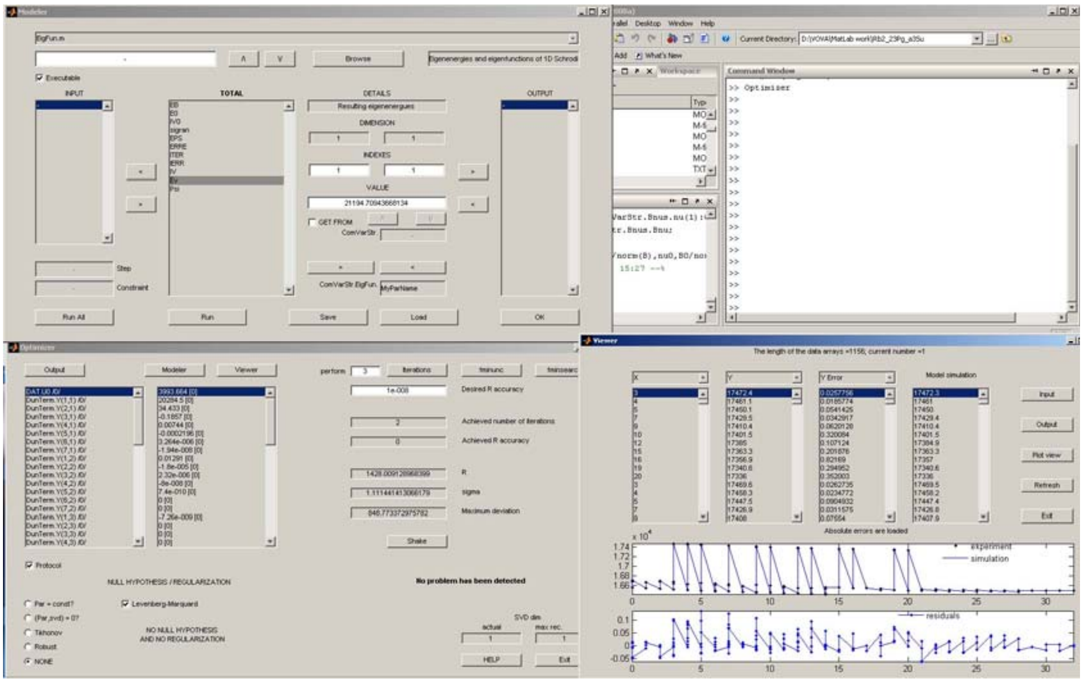
This led us to a concept of the multi–block model. We needed a tool to construct a however complicated model composed from simpler standardized algorithmic blocks, which could be collected into libraries. In earlier years, we tried to do it with FORTRAN, but later we came to a conclusion that Matlab is highly suitable to this task, allowing us to use its libraries of various algorithms and, what is especially important, to transfer the data between the blocks via Matlab workspace.

As a result, within a several years (rather decades) we developed, tuned, and applied to various problems from physics: the computational tool “Optimizer” for working with multi–block models. We would like to emphasize that this tool contains many different features and capabilities. Its full description would require much larger space. Below only a short introduction into the most important features of the package is presented. The detailed description is available now in Russian language.

II. Program Interfaces

The visual interfaces of the program are represented by three windows: “Modeler”, “Optimizer”, and “Viewer”, aimed to construction of a multi–block model, its optimization, and inspecting the results correspondingly. The print-screen copy of them is shown in Fig. 1.

The Modeler window allows a user to work with model blocks, adding or deleting them, and defining how the data flow between the blocks. Blocks can be either executable (modeling programs) or nonexecutable; the latter ones usually play roles of data banks. The entire set of the declared variables of a block is seen in the list “TOTAL”, those variables which are inserted into the list “INPUT” becomes the fitting parameters of the model, the ones inserted into the list “OUTPUT” comprise the set of data to be fitted via their comparison to the experimental data. Short text tags can be added to the entire block and to every variable for their better comprehension. The names of all the blocks are contained in the uppermost list of the window, which can be unfolded, so that the block needed currently can be picked from it. The current state of the entire model with all the parameters and the text tags can be saved into (loaded from) a MOT–file; a separate block can be saved into (loaded from) a MOP–file; the MOT– and MOP– files are formatted as standard Matlab MAT–files. The executable blocks must be supported by executable m–files with the same names; of course, these m–files can call other m–files as well as procedures written in other languages via the mechanisms provided by Matlab.



EigFun.m

Eigenenergies and eigenfunctions of 1D Schrodi

☑ Executable

```
EB
E0
IV0
sigran
EPS
ERRE
ITER
IERR
IV
EV
Psi
```

Resulting eigenenergies

DIMENSION 1 1

INDEXES 1 1

VALUE 21194.70943668134

□ GET FROM ComVarStr.

ComVarStr.EigFun. MyParName

```
>> Optimizer
>>
```

perform 3 Iterations

1e-008 Desired R accuracy

2 Achieved number of iterations

0 Achieved R accuracy

1428.0091 28968399 R

1.111441413066179 sigma

846.77337297 5782 Maximum deviation

```
DAT.U0 /0/            3993.664 [0]
DunTerm.Y(1,1) /0/    20284.5 [0]
DunTerm.Y(2,1) /0/    34.433 [0]
DunTerm.Y(3,1) /0/    -0.1857 [0]
DunTerm.Y(4,1) /0/    0.00744 [0]
DunTerm.Y(5,1) /0/    -0.0002196 [0]
DunTerm.Y(6,1) /0/    3.264e-006 [0]
DunTerm.Y(7,1) /0/    -1.94e-008 [0]
DunTerm.Y(1,2) /0/    0.01291 [0]
DunTerm.Y(2,2) /0/    -1.8e-005 [0]
DunTerm.Y(3,2) /0/    2.32e-006 [0]
DunTerm.Y(4,2) /0/    -8e-008 [0]
DunTerm.Y(5,2) /0/    7.4e-010 [0]
DunTerm.Y(6,2) /0/    0 [0]
DunTerm.Y(7,2) /0/    0 [0]
DunTerm.Y(1,3) /0/    -7.26e-009 [0]
DunTerm.Y(2,3) /0/    0 [0]
DunTerm.Y(3,3) /0/    0 [0]
DunTerm.Y(4,3) /0/    0 [0]
```

☑ Protocol

NULL HYPOTHESIS / REGULARIZATION

No problem has been detected

○ Par = const?

○ (Par,svd) = 0?

○ Tikhonov

○ Robust

● NONE

☑ Levenberg-Marquard

NO NULL HYPOTHESIS AND NO REGULARIZATION

SVD dim actual 1 max rec. 1

The length of the data arrays =1156; current number =1

```
X      Y         Y Error      Model simulation
3      17472.4   0.0257756    17472.3
4      17461.1   0.0185774    17461
5      17450.1   0.0541425    17450
7      17429.5   0.0342917    17429.4
9      17410.4   0.0620128    17410.4
10     17401.5   0.320084     17401.5
12     17385     0.107124     17384.9
15     17363.3   0.201876     17363.3
16     17356.9   0.82169      17357
19     17340.6   0.294952     17340.6
20     17336     0.352003     17336
3      17469.6   0.0262735    17469.5
4      17458.3   0.0234772    17458.2
5      17447.5   0.0904932    17447.4
7      17426.9   0.0311575    17426.8
9      17408     0.07554      17407.9
```

Absolute errors are loaded

FIGURE I. VISUAL INTERFACES OF THE PROGRAM

The “Viewer” window works with the experimental data (Y) and with the current results of their modeling (Model simulation). The experimental data can be weighted via their individual uncertainties (Y Error) or with their full covariance matrix. All the data are shown in the window both numerically (in the corresponding lists) and graphically. The list X is a numeration of the experimental and model data used for their reliable graphical view; it can or cannot coincide with some of the model parameters. The experimental and/or model data can be input from (output into) text files.

The “Optimizer” window allows the user to choose the optimization regime, to tune it, and to perform the optimization. The fitting parameters with their current values and estimated uncertainties are shown in the lists of the window. The stopping criteria are defined by the maximum number of iterations and by the value of the “Desired R accuracy”, where R is the evaluation function. The results of the optimization can be saved into text files.

The workspace of the entire model is concluded into the global structure ComVarStr; every model block is presented by a field with the name of the block, which is also a structure containing all the declared variables and technical parameters. Therefore, when writing a new modeling (block) program, the global structure ComVarStr must be addressed, and all the variables, to be accessed from the “Optimizer”, must be presented by subfields of the block field. The common technical parameters are also contained in the fields of the structure ComVarStr; they can be accessed directly from the Matlab command window for inspecting and fine-tuning of the model, but generally, this is not necessary.

III. Algorithms

By default, the evaluation function is constructed as a residual sum of squares of differences between all the experimental data and results of their modeling. Consequently, the optimization is done with the standard Least Square Fit (LSF) [1–4] based on the Singular Value Decomposition (SVD) [4] of the design matrix. The local design matrix (Jacobi matrix) is composed from the 1st derivatives of the output parameters with respect to the input parameters. The derivatives can be either provided by the modeling (block) programs or estimated by the “Optimizer” with the finite–differences algorithm. The initial steps for computing the derivatives are defined in the corresponding text box of the “Modeler” and automatically adapted in a course of the computations; zero steps indicate that the parameter (e.g., some element of an array) should not be varied. The partial design matrices of every block are automatically combined into the global design matrix of the entire model.

Nonlinear LSF uses the Levenberg–Marquardt algorithm [2]; when the finite–differences scheme is implemented, it combines with the coordinate–wise descent method.

The least–square estimator can be replaced by one of the robust estimators from the list provided, with tunable parameters of robustness. Other ways to regularize the solution include the truncated SVD and the Tikhonov regularization. Constraints in a form of equalities can be applied, which are reliable within the local linear approximation. The *p*–value of the hypothesis of the validity of the constraints or of the SVD truncation is estimated with the linear Fisher criterion.

The full covariance matrix of the solution is computed and can be output into the text file along with other results.

The program supports parallel computing. In one of the regimes, the model can be parallelized by indicating the groups of blocks, which can be computed independently. In the other regime, the optimization algorithm itself is parallelized; the latter approach is usually more efficient.

The button "Shake" in the "Optimizer" window is implied to be a remedy against local minima. The strength of the shaking can be regulated via fields of the global structure ComVarStr.

The program supports a rather flexible mechanism of errors processing. In a case of an error in one of the model (block) programs, the optimization process is not terminated, but the steps of the fitting parameters are adjusted to avoid an error at the next stage of the computation.

There is also a set of auxiliary routines. In part, they help to apply any optimization tool from Matlab libraries to the model, as soon as it is constructed.

IV. Model Program Blocks

Although, in our belief, the present tool can find applications in various fields, our primary interest has concentrated on the analysis of the molecular spectra. Our library of model blocks contain relevant programs. It includes a group of common approximants, such as generalized splines [9], Pade approximants, (bi)Gaussians and (bi)Lorentzians, generalized Pearson curves [10]. The most principal part of the library includes routines for simulating the quantum–mechanical dynamics in a molecule: the Numerov algorithm, the Renormalized Numerov algorithm, and the Mapped Fourier Grid algorithm for solution the time–independent 1D (generally, multichannel) Shrödinger equation; the split–operator algorithm for solution the time–dependent 1D multichannel Shrödinger equation; and many other routines specific for the molecular spectroscopy. See [5–9] and references therein.

V. Applications

The program has been successfully applied to many problems in experimental molecular spectroscopy. Most of the results could be hardly obtained without it, or would at least require much extensive efforts.

Some of the examples are presented below.

Figure 2 shows results of our analysis of the rotational constant of near–dissociation vibrational levels of the 0^+_u state of Cs_2 at the atomic limit $6s_{1/2}+6p_{1/2}$, measured in the photoassociation experiments of ultracold atoms. The origin of the undulations observed in the dependence of the rotational constant on the vibrational quantum number is the nonadiabatic mixing with other states, mostly with the 0^+_u state below the next fine–structure limit $6s_{1/2}+6p_{3/2}$. Simultaneously with the rotational constant, the experimental vibrational term values, as well as the *ab initio* and the earlier empirical potential energy curves were included into the set of the initial data of the analysis. We modeled the data with the full–scale quantum–mechanical computation using the Mapped Fourier Grid algorithm. As a result, we determined (updated) parameters of the potential energy curves and of the function of the nonadiabatic mixing (the interaction matrix element). The results of this analysis are published in [8].

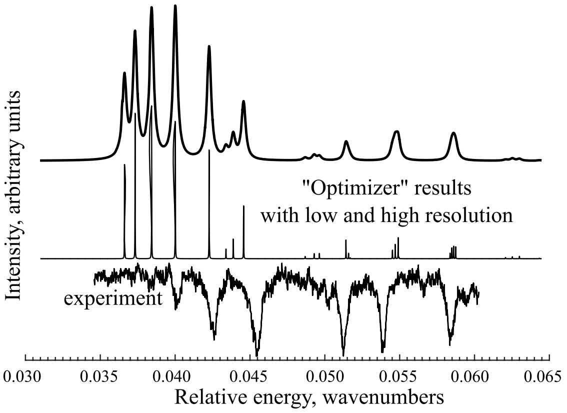


FIGURE II. THE EXPERIMENTAL AND MODELLED SPECTRUM OF NaCs IN THE NEAR–DISSOCIATION LEVEL AT THE LIMIT $6S_{1/2}+6P_{3/2}$ WITH THE PARTLY RESOLVED HYPERFINE STRUCTURE

Figure 3 presents for the first time our most resent result—the analysis of the hyperfine structure in a near–dissociation level of NaCs below the limit $6s_{1/2}+6p_{3/2}$, which was also observed in the photoassociation experiments of the ultracold atoms. To simulate the spectrum we computed intensities of the free–bound transitions from the states of the lowest atomic limit $6s_{1/2}+6s_{1/2}$ to the states of the limit $6s_{1/2}+6p_{3/2}$ within a model of 4 initially degenerated levels, which split into the multitude of the fine and hyperfine sublevels. The only fitting parameters were the relative shifts of these unperturbed levels, while the interaction matrix elements were kept equal to their asymptotic atomic values. In spite of the relative simplicity of this model, it allowed us to reproduce the most important features of the experimental spectrum.

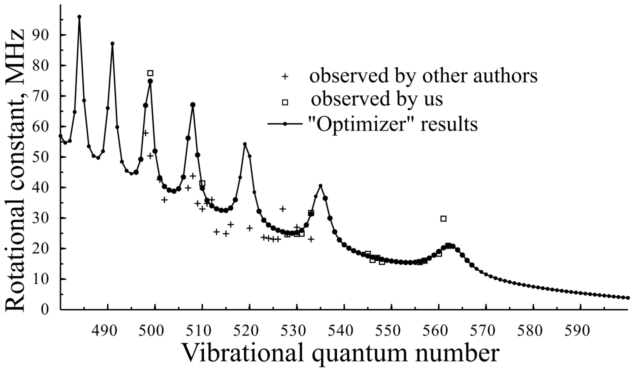


FIGURE III. THE EXPERIMENTAL AND MODELLED ROTATIONAL CONSTANTS OF THE NEAR–DISSOCIATION LEVELS IN THE STATE 0^+_u OF Cs_2 AT THE LIMIT $6S_{1/2}+6P_{1/2}$

VI. Conclusions

The computational tool “Optimizer” for constructing and optimizing multi–block models is presented. The main characteristics of this program are described. The applications to the spectroscopy experiments with ultracold gases of alkali metals exemplify the capabilities of this tool.

The first author did most of the programming work. The group in Shanxi University headed by the second author did most of the experiments. The analysis of the experimental data was done together.

Acknowledgment

V. B. Sovkov acknowledges Shanxi University for the support of his visits there in 2015 and in 2016.

The work was supported by the 973 Programs (No. 2012CB921603), the PCSIRT (No. IRT13076), the National Natural Science Foundation of China (No. 91436108, No. 61378014, No. 61308023, No. 61378015, and No. 11434007), the SRFDPHE (No. 20131401120012).

References

[1] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical Recipes: The Art of Scientific Computing, 3rd ed., Cambridge: Cambridge University Press, 2007.

[2] J. Nocedal and S. J. Wright. Numerical Optimization, New York: Springer, 1999.

[3] K. Varmuza and P. Filzmoser. Introduction to Multivariate Statistical Analysis in Chemometrics. Boca Raton: Taylor & Francis Group, 2009.

[4] G. H. Golub, Ch. F. van Loan. Matrix Computations, 3rd ed., Baltimore and London: The John Hopkins University Press, 1996.

[5] B. Beser, V. B. Sovkov, J. Bai, E. H. Ahmed, C. C. Tsai, F. Xie, Li Li, V. S. Ivanov, and A. M. Lyyra. “Experimental investigation of the ^{85}Rb$_2$ $a^3\Sigma_u^+$ triplet ground state: Multiparameter Morse long range potential analysis,” J. Chem. Phys., vol. 131, 094505, 2009.

[6] F. Xie, Li Li, D. Li, V. B. Sovkov, K. V. Minaev, V. S. Ivanov, A. M. Lyyra, and S. Magnier. “Joint analysis of the Cs$_2$ $a^3\Sigma_u^+$ and 1_g ($3^3\Pi_{1g}$) states,” J. Chem. Phys., vol. 135, 024303, 2011.

[7] J. Ma, W. Liu, J. Yang, J. Wu, W. Sun, V. S. Ivanov, A. S. Skublov, V. B. Sovkov, X. Dai, and S. Jia. “New observation and combined analysis of the Cs$_2$ 0_g^-, 0_u^+, and 1_g states at the asymptotes $6S_{1/2}$+$6P_{1/2}$ and $6S_{1/2}$+$6P_{3/2}$,” J. Chem. Phys., vol. 141, 244310, 2014.

[8] W. Liu, R. Xu, J. Wu, J. Yang, S. S. Lukashov, V. B. Sovkov, X. Dai, J. Ma, L. Xiao, and S. Jia. “Observation and deperturbation of near-dissociation ro-vibrational structure of the Cs$_2$ state 0_u^+ ($A^1\Sigma_u^+ \sim b^3\Pi_{0+u}$) at the asymptote $6S_{1/2}$+$6P_{1/2}$,” J. Chem. Phys., vol. 143, 124307, 2015.

[9] J. Yang, Y. Guan, W. Zhao, Zh. Zhou, X. Han, J. Ma, V. B. Sovkov, V. S. Ivanov, E. H. Ahmed, A. M. Lyyra, and X. Dai. “Observation and analysis with the spline-based Rydberg-Klein-Rees approach for the $3^1\Sigma_g^+$ state of Rb$_2$,” J. Chem. Phys., vol. 144, 024308, 2016.

[10] V. S. Ivanov and V. B. Sovkov. “Generalization of the formalism of Pearson curves in the problems of approximation of profiales of continous bands in electronic spectra of molecules,” Opt. Spectrosc., vol. 81, pp. 680–684, 1996.