KIPET – AN OPEN-SOURCE KINETIC PARAMETER ESTIMATION TOOLKIT

Michael Short[a], Christina Schenk[a], David Thierry[a], Jose Santiago Rodriguez[b], Lorenz T. Biegler[a*] and Salvador Garcia-Muñoz[c]
[a] Carnegie Mellon University
Pittsburgh, PA 15213
[b] Purdue University
West Lafayette, IN 47907
[c] Eli Lilly and Company
Indianapolis, IN 46285

Abstract

This paper presents a new software package, *KIPET*, which is designed to estimate kinetic parameters from dynamic chemical reaction systems. The software toolkit is based on a unified framework that makes use of maximum likelihood principles, collocation-based discretization methods, and large-scale nonlinear optimization. *KIPET* contains a wide array of tools for kinetic parameter estimation and model evaluation in an easy-to-use open-source Python-based framework. The package can currently be used for data pre-processing, simulation of reactive systems described with differential algebraic equations, estimability analysis, estimation of system variances and measurement errors separately, estimation of kinetic parameters from spectroscopic data or concentration data, and the estimation of parameter confidence intervals using the NLP sensitivities obtained from sIPOPT. Since large-scale NLP problems require robust initialization strategies, a variety of tools for initialization are also included. *KIPET* utilizes Pyomo, a Python-based open-source optimization modeling language, in the background to formulate and solve all optimization problems and leverages other open-source Python packages to provide visualization of results. *KIPET* is well-documented and available for free download from the code-hosting site Github.

Keywords

Parameter Estimation, Nonlinear Programming, Chemical and Pharmaceutical Processes

Introduction

Great improvements have been made to the speed, reliability, accuracy, and cost associated with instrumental analytical measurement techniques and the use of these to gain insights into lab-scale processes is vital in the chemical and pharmaceutical industries to design scalable, controllable, and optimized process designs. Spectroscopy is commonly used for the monitoring of chemical reactions and produces large amounts of data which can be used to infer the concentrations of species in the reactor, based on the absorption of individual species in the ultraviolet visible, Raman, near-infrared, and infrared wavelength regions.

Many techniques are available to utilize this valuable information obtained from experimental data in order to obtain estimates for the kinetic parameters based on a candidate reaction mechanism model. A commonly applied approach is that of multivariate curve resolution-alternating least squares (MCR-ALS) (de Juan and Tauler, 2000), which combines the soft-modeling MCR approach (Lawton and Sylvestre, 1971) with a hard-modeling approach. MCR

* To whom all correspondence should be addressed

estimates pure component absorption spectra and their concentration profiles from the spectroscopic data based on Beer's light absorption law when no information is known regarding the reaction mechanism. The problem suffers from rotational ambiguities and thus cannot be decomposed to unique solutions. MCR-ALS attempts to impose hard constraints on this problem by introducing a chemical reaction model and non-negativity constraints. This significantly reduces rotational ambiguity and has been applied to a wide variety of problems (Jaumot et al., 2015; de Juan et al., 2014), however the approach lacks the guidance of explicit search directions when minimizing the fitting error and thus converges very slowly.

Another common technique, referred to as a hard-modeling approach, is the Gauss-Newton-Levenberg-Marquardt (GNLM) method (Marquardt, 1963**)** where the kinetic parameters are initialized and a system of ordinary differential equations (ODEs) is integrated to provide initial concentration profiles for the reaction species. Least squares regression is then used to obtain the pure component spectra. Following this the parameters are updated based on the error and the procedure repeats until convergence. Many variants of this approach have been proposed, particularly looking to reduce the large numbers of data points handled by the curve resolution algorithm through factor analysis (Bijlsma et al, 2001; Maeder and Zuberbuhler, 1990; Sawall et al., 2012). A comprehensive review of these techniques and other MCR techniques can be found in (de Juan et al, 2014).

These methods all converge slowly and perform poorly when unstable and ill-conditioned dynamic systems are solved. Additionally both methodology classes rely heavily on initial values and provide no guarantee on convergence (Chen et al., 2018). These techniques also fail when the concentration matrix columns are dependent, as they require (C^TC) to be nonsingular. Furthermore, system noise in the reaction model is also not considered in these methods.

These methods are commonly utilized in industry to obtain kinetic parameter estimates and are implemented in an array of different modeling tools and proprietary and commercial software packages. Statistical languages such as R and S are commonly employed for a variety of parameter estimation problems, however few packages deal with spectra directly. The commercial process modeling platform gPROMS is also widely used for experimental design and parameter estimation, however these tools do not provide simultaneous parameter estimation combined with spectra, rather allowing the user to test the candidate models with responses sequentially and then allowing the user to validate and judge the quality based on statistical tools.

Toolkits like ACADO, CasADi, PARFIT provide many useful features regarding the solution of parameter estimation, sensitivity analysis, etc., but do not provide capabilities of solving simultaneous parameter estimation combined with spectra.

General regression (GREG and GREGPLUS) (Stewart et al., 1992), included in the commercial package Athena Visual Studio (Stewart and Caracotsios, 2008) tests models against data on a limited number of response types and helps users to plan further experiments using linear regression and statistical techniques.

MCR-ALS GUI2.0 from Tauler et al. (2015) is implemented in MATLAB and freely available, and a number of other open-source tools based on these soft- and hard-modeling approaches described above are available such as HyperSpy (de la Pena et al., 2018).

A new approach was proposed by Chen et al. (2016) which is a more comprehensive and holistic unified framework for reaction kinetic parameter estimation based on maximum likelihood principles and collocation methods. This approach forms the backbone of the *KIPET* (Kinetic Parameter Estimation Toolkit) software package, which will be introduced in the next section. The approach uses an extended maximum likelihood approach and collocation methods to first deconvolve system variable noise from the measurement errors and then solve the parameter estimation problem simultaneously. This method provides reliable, robust solutions quickly. In addition to the application of these new methods to an open-source, Python-based framework, *KIPET* also aims to provide users with a set of additional tools to aid in measuring the estimability of parameters, simulating, and testing their models, without tedious programming in a user-oriented workflow design with detailed examples and documentation.

The rest of this paper will focus on the mathematical details of the unified framework of Chen et al. (2016) followed by a description of the software implementation in *KIPET*. A tutorial example is then presented before the conclusion and future work.

Mathematical Background and Framework

This section briefly describes the mathematical details of the unified framework upon which *KIPET* is based. For more details and derivations, the reader is referred to the original paper. Beer-Lambert's absorbance law, when considering measurement error, can be expressed as Eq (1):

$$D = CS^T + E \tag{1}$$

where D is the spectral data matrix (*ntp* x *nwv*) with *ntp* being the number of sampling time points and *nwv* the measured wavelengths. C is the species concentration matrix (*ntp* x *nc*), containing system variable noise, where *nc* is the number of absorbing species and S is the pure component absorbance matrix (*nwv* x *nc*). E is the *ntp* x *nwv* measurement error matrix, which we assume to have the same normal distribution for all measurements. Assuming that the concentration profile without system noise, $z(t)$, can be represented by the following ordinary differential equations (ODEs) Eq (2).

$$\frac{dz(t)}{dt} = f(z(t), y(t), \theta)$$
$$g\big(z(t), y(t)\big) = 0 \tag{2}$$

where $y(t)$ are the algebraic variables (e.g. reaction rates) and θ is the kinetic parameter vector. If the measurement variance (δ^2) and system variable (model) variance (σ_k^2) are known, one can solve a nonlinear optimization problem combining maximum likelihood principles, Beer-Lambert's law (Eq. 1) and the DAE system in Eq. (2) to estimate the kinetic parameters from spectral data. One can also efficiently discretize the problem using orthogonal collocation on finite elements to obtain Eq. (3):

$$
\begin{aligned}
min \sum_{i=1}^{ntp}\sum_{l=1}^{npw}(d(t_i,\lambda_l)-\sum_{k=1}^{nc}c_k(t_i)s_k(\lambda_l))^2/\delta^2 \\
+\sum_{i=1}^{ntp}\sum_{k=1}^{nc}(c_k(t_i)-z_k(t_i))^2\ /\sigma_k^2 \\
s.t.\sum_{m=0}^{K}\dot{l}_m(\tau)\,z_{jm}=h_j.f\left(z_{jm},\theta\right),\qquad j=1..ne, \\
m=1..K \\
z^K(t_i)=\sum_{m=0}^{K}\dot{l}_m(\tau)\,z_{jm},\qquad \tau=\frac{t_i-tp_{j-1}}{tp_j-tp_{j-1}} \\
z^K(t_i)=z_{i-1}+h_i\sum_{m=1}^{K}\dot{l}_m(\tau).\dot{z}_{im},\qquad \dot{z}_{im}=f(z_{im},t_{j,m}) \\
C\geq 0, S\geq 0 \qquad (3)
\end{aligned}
$$

where $z^K(t)$ is a Lagrange interpolating polynomial and l_m is the polynomial basis function. Note that the second term in Eq. (3) describes the system variable noise, with variance σ_k^2, minimizing the difference between noised concentration profiles, C, and modeled ones, Z.

In addition to introducing a framework to directly estimate kinetic parameters from spectral data, Chen et al. (2016) was the first to propose a numerical procedure to estimate measurement and model variances δ^2, σ_k^2.

Variance Estimation

The iterative variance estimation algorithm splits problem Eq. (3) into a series of optimization subproblems. The initial subproblem, used to obtain the initialization for the concentration profiles $c_k(t_i)$, assumes that there is no system noise, solving Eq. (3) for only the first term in the objective function, and ignoring δ^2. These are then applied to an equivalent probability distribution function Eq. (4), derivation in Chen et al. (2016):

$$min \sum_{k=1}^{c} ln(\frac{1}{ntp}\sum_{i=1}^{ntp}((c_k(t_i)-z_k(t_i))^2) \qquad (4)$$

subject to the discretized DAE system. After obtaining estimates for $c_k(t_i)$ and $z_k(t_i)$ from Eq. (4) one can solve a second subproblem, Eq. (5), to obtain guesses for the kinetic parameters and the $z(t)$ profiles.

$$min \sum_{i=1}^{ntp}(d_{i,l}-\sum_{k=1}^{nc}(z_k(t_i)s_k(\lambda_l))^2 \qquad (5)$$

This provides an initialization for $s_k(\lambda_l)$. Following this another manipulated maximum likelihood formulation, Eq. (6) can be solved with $s_k(\lambda_l)$ and $c_k(t_i)$ fixed:

$$min \sum_{i=1}^{ntp}(d_{i,l}-\sum_{k=1}^{nc}(c_k(t_i)s_k(\lambda_l))^2 \qquad (6)$$

The sequence of optimization problems repeats until convergence for C, S, δ, and σ_k^2 is achieved. Once convergence is achieved ,Eq. (7) provides an estimate of measurement variance:

$$v^2=\frac{1}{ntp}\sum_{i=1}^{ntp}(d_{i,l}-\sum_{k=1}^{nc}(z_k(t_i)s_k(\lambda_l))^2 \qquad (7)$$

where v^2 is an estimate of the measurement variance. Once there is convergence it is possible to estimate σ_k^2 and δ^2 from the overdetermined system. This algorithm is represented graphically in Figure 1.

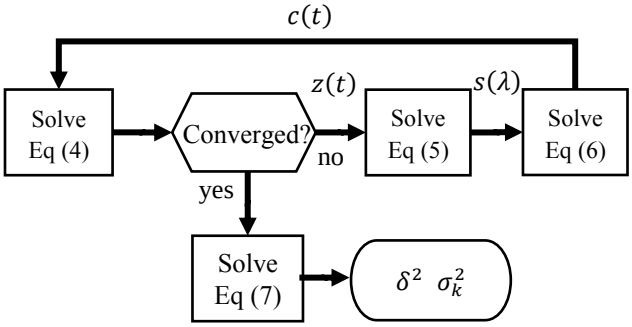

Figure 1. The variance estimation algorithm, adapted from Chen et al. (2016)

In addition to providing estimates for the system and measurement variances, the final solution provides very good initializations for the remaining system variables. Thus, the solution to the variance estimation problem is then used to initialize the full simultaneous parameter estimation problem, Eq. (3).

Solution and confidence intervals

Upon solution of the large-scale NLP problem, *KIPET* uses implicit function theorem and the optimality conditions together with the calculation of the reduced Hessian to approximate the covariance matrix. *KIPET* uses either sIPOPT (Pirnay et al., 2012) or the newly developed sensitivity package, k_aug (Thierry and Biegler, 2018) to calculate the reduced Hessian. k_aug uses IPOPT's output to generate and factorize the KKT matrix and is able to be used for calculation of both the sensitivities and the reduced Hessian, which provides an advantage over sIPOPT for its usefulness to estimability analysis. sIPOPT and k_aug can be used interchangeably for covariance calculations.

Further mathematical details about the models and convergence proofs are found in in Chen et al. (2016). In their study, the rigorous approach described here was shown to outperform other toolboxes such as the MCR-ALS 2.0 for a large number of test problems. It is thus a prime candidate for expansion and implementation into an all-in-one software suite, with enhanced capabilities and improvements to usability. The newly developed software toolbox, its software implementation, functionality and capabilities are described in the following section.

Software Framework

The software implementation of the techniques described above is summarized in Figure 2. The software is roughly split into simulation (left side of Fig. 2), parameter estimation (right side of Fig 2), and estimability analysis. Each one of the blocks represented in the figure shows a *KIPET* module, all programmed in Python, with its name in brackets. This section will briefly described the contents of each module and how these are used to solve kinetic parameter estimation problems.

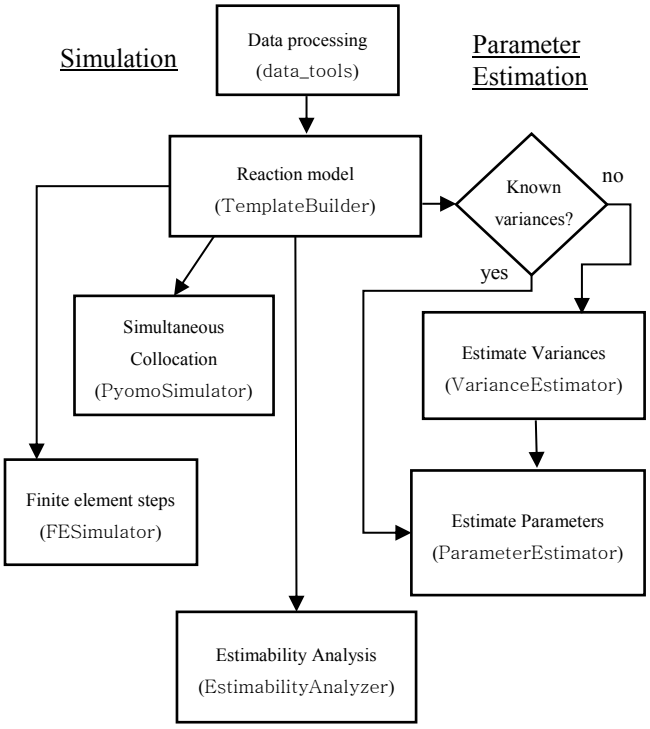

Figure 2. The module structure / steps involved in solving a problem in Kipet

Data Tools

The data tools module reads in data from commonly used file formats and sorts and inputs the data into pandas dataframes, a data structure and analysis Python package, which *KIPET* uses to store and manipulate matrices. This module also contains tools used to generate and add noise to simulated data, as well as commonly-used spectral pre-processing techniques including the Savitzky-Golay filter (Savitzky and Golay, 1964), Standard Normal Variate (Barnes et al., 1989), and Multiplicative Scatter Correction (Maleki et al., 2007).

Model Building

KIPET uses the open-source optimization modeling language Pyomo (Hart et al., 2017) to formulate all optimization problems as it provides a flexible optimization modeling environment with access to many state-of-the-art solvers. The flexibility and object-oriented aspects of Pyomo along with the dynamic optimization tools that are included in pyomo.dae (Nicholson et al., 2018) form the fundamental basis for how *KIPET* functions.

On the other hand, *KIPET* does not require the user to interact with Pyomo to utilize its functionality. The TemplateBuilder is essentially the Pyomo model constructor, taking in the differential and algebraic equations and states, parameters and their initializations, and the feeding times for dosed systems from the user and formulating the optimization problem (Eq. 3) automatically in the background. After building the model the user can then select and apply the discretization scheme, which utilizes pyomo.dae to automatically discretize the dynamic system using orthogonal collocation on finite elements. The TemplateBuilder thus allows for an easy-to-use intuitive model building framework that should be accessible for users of all programming skill levels. An example of what a model built in this class looks like is presented in the tutorial problem section of this paper.

Simulation

KIPET provides a number of simulation options. Simulation is an essential part of *KIPET* as it not only allows a user to create simulated data and test different potential reaction mechanisms, but also gives the user robust and reliable ways to initialize the parameter estimation problem, which is a crucial and difficult task for all large-scale NLP problems.

The PyomoSimulator solves the fully discretized differential-algebraic equations (DAE) problem simultaneously, with the parameters fixed. After solving this problem, the PyomoSimulator automatically patches the locally optimal values for C and S back into the full model. Another option for simulation is to use the FESimulator, which utilizes fe_factory as its basis, allowing for a finite element-by-element march-forward solution strategy. In this approach, *KIPET* uses pyomo.dae to solve each individual finite element's collocation equations as a separate problem with IPOPT (Wächter and Biegler, 2006), using the fact that when Radau collocation points are utilized we also have the solution for the initial point of the next element. FESimulator thus provides *KIPET* with a form of implicit Runge-Kutta integration. This technique is extremely useful for solving large problems that are difficult to solve simultaneously and also has the advantage that it can be adapted for systems in which we have inputs (dosing) as the sizes of the finite elements can be easily manipulated.

Typically, for large problems, a simulation in *KIPET* is run first with fixed parameters and the results are used to initialize the full variance estimation and parameter estimation problems. This procedure is common practice when solving dynamic optimization problems as simulation provides initial feasible solution to the optimization problem. This initialization is done automatically within *KIPET*.

Variance Estimation

The VarianceEstimator class performs the iterative variance estimation algorithm detailed in the previous section. This algorithm is the slowest part of the *KIPET* package as it requires a sequence of optimization problems to be solved until the convergence criteria is reached. In addition to obtaining the δ^2 and σ_k^2, the VarianceEstimator also obtains values for kinetic parameters and concentration profiles that can be used as improved initializations to the parameter estimation problem.

Parameter Estimation

Finally, in order to obtain the kinetic parameter estimates, the ParameterEstimator is used. This module solves the full simultaneous optimization problem, based on the initializations given and with the variances obtained from the previous step. The *S*, *C*, *Z*, θ, and any algebraic state variables are all left as variables in this case. Through using IPOPT to solve the NLP we can also calculate the covariance matrix using k_aug or sIPOPT. *KIPET* also provides the option to solve kinetic parameter estimation problems where concentration data is known by removing the first term of the objective function in Eq (3) and solving a minimization of the least squares problem.

Estimability Analysis

Another of *KIPET*'s capabilities is to provide the user with a measure of parameter estimability through the EstimabilityAnalyzer module. This module contains tools to rank parameters in terms of estimability and then solves a sequence of simplified models using the ParameterEstimator. The estimability assessments are actively being expanded to include estimability based on spectra, however, the latest version of *KIPET* contains the ability to perform detailed estimability analysis for concentration data problems only. The parameter-ranking scheme of Yao et al. (2003) is used to perform the parameter ranking based on the sensitivities obtained through k_aug. After the parameter rankings are obtained, a modified version of the techniques proposed by Wu et al. (2011) is used to obtain the number of parameters that should be estimated. This is done through solving a sequence of parameter estimation problems, using ParameterEstimator, where the most estimable parameters are left as variables and the least estimable are fixed. These simplified models are then compared to the full problem through mean squared errors. The EstimabilityAnalyzer returns which parameters should be fixed and which should be left as variables in the model.

With these core modules, *KIPET* provides a large number of useful capabilities to a research chemist. *KIPET* has many other useful features, such as providing the option of solving problems where there are known non-absorbing components (reacting or not-reacting), and the option to include complementary measurement data, such as temperature and volume measurements. The ability to easily include dosing into the parameter and variance estimation by exploiting the discretization structure and new tools developed in the package is another major advantage of *KIPET*.

Using the detailed and adaptable discretization strategies discussed, in combination with the advantages of an object-oriented programming language like Python, we are able to exploit these benefits to provide an adaptable and reliable tool which, when a fast and powerful nonlinear optimization solver such as IPOPT is utilized, provides fast solutions to an array of kinetic parameter estimation problems.

A full list of *KIPET*'s capabilities, how they work, the installation instructions, and a large number of instructional tutorial problems and additional examples are provided in *KIPET*'s documentation. The documentation aims to help beginner users with little programming and optimization experience to utilize all of *KIPET*'s powerful capabilities.

Example Problem

A simple illustrative example problem of a dosed system is presented here. The reaction system is shown in Eq. (8).

$$\begin{aligned}
&\frac{dc_A}{dt} = -k_1 c_A + d(t), \quad c_A(0) = 0.01 \\
&\frac{dc_B}{dt} = k_1 c_A - k_2 c_B, \quad c_B(0) = 0 \\
&\frac{dc_C}{dt} = k_2 c_B, \quad c_C(0) = 0 \\
&0 \le k_1 \le 5, \quad 0 \le k_2 \le 5 \\
&d(t) = \begin{cases} 0.001, & if\ t = 3.6341 \\ 0 & else \end{cases}
\end{aligned} \tag{8}$$

The spectral data matrix for this problem is a 300 x 100 matrix. After running the VarianceEstimator, we obtain variance estimates for species *A, B,* and *C* of 4.928 x 10^{-8}, 8.817 x 10^{-8}, and 9,366 x 10^{-8} respectively and measurement error of 3.346 x 10^{-6}. The ParameterEstimator is then run and the results obtained are shown in Table 1 and Figures 3 and 4. The entire process of simulation for initialization using FESimulator, followed by variance estimation and parameter estimation took 135.62 CPUs.

Table 1. Results of kinetic parameter estimation for example problem

Parameter	Initial	Estimated	Confidence Interval
k_1	0.9	0.2951	(0.2764, 0.3137)
k_2	0.2	1.4667	(1.3490, 1.5844)

This example illustrates the speed and efficiency of the proposed approach, as well as how easily dosed systems can be handled. After the model setup, only a few lines of code are needed to run the problem. The confidence intervals are computed using k_aug in this example. This example is one of many that are available on the software download page along with documentation.

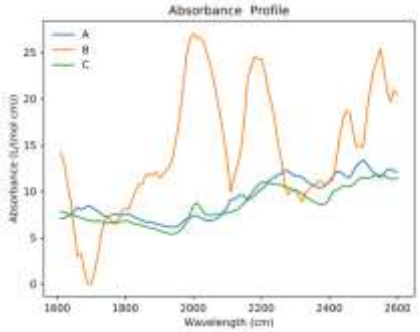

Figure 3. Individual species' absorbance profiles

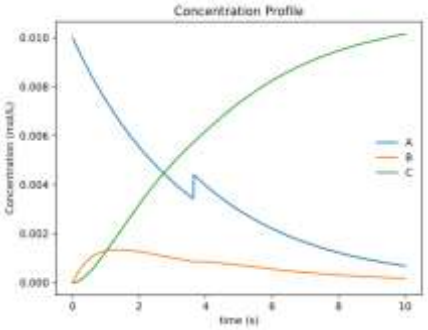

Figure 4. Concentration profiles

Conclusions and Future Work

A new Python-based open-source software package, *KIPET*, has been presented in this paper. The software aims to provide an array of tools to chemists that allow for the efficient estimation of kinetic parameters from experimental data. The software is capable of reading spectral data or concentration data directly from commonly-used instruments, provides data pre-processing tools, estimability analysis tools, built-in variance estimation, and kinetic parameter estimation with confidence intervals obtained from the problem sensitivities. The software is capable of solving a wide variety of problems, including problems with different dosing schemes, all based on the unified simultaneous optimization-based framework presented by Chen et al. (2016), utilizing maximum likelihood principles and collocation methods.

The software uses the Pyomo optimization package to formulate the optimization problem and perform the model discretization, along with fe_factory, a newly developed class that performs a march-forward finite element-by-element approach. IPOPT is used to solve the resulting large-scale NLPs and sIPOPT and k_aug can be utilized to obtain the covariance matrices through optimal solution sensitivity calculations.

The paper briefly describes the ways in which *KIPET* can be used to solve large parameter estimation problems through a variety of strategies and briefly demonstrates example code in order to demonstrate *KIPET*'s ease-of-use.

KIPET can provide fast solutions to a variety of problems, however the drawbacks include the amount of time involved in the iterative variance estimation algorithm, reliance on a good kinetic model provided by the user, as well as good initializations. *KIPET* provides the user with a large array of initialization options. However, the user can potentially spend a lot of time selecting the most suitable initialization approach.

KIPET, and all third-party software it uses (apart from HSL libraries which are free for academic use), is released under the GNU general public license 3.0 and is freely available on Github[1]. Kipet also has an extensive user manual with installation instructions and an array of tutorial problems to make it accessible to new users.

KIPET currently includes all of the features discussed above and is currently under active development with a slew of new features to be added. Methods to obtain parameter estimation with unknown absorbing species (based on Chen, et al, 2018), enhanced estimability methods for problems with spectra, and the application of the package to enhance experimental design are planned.

Acknowledgments

The authors would like to gratefully acknowledge the funding of Eli Lilly and Company.

References

Barnes R.J., Dhanoa M.S., Lister S.J. (1989). Standard normal variate transformation and de-trending of near-infrared diffuse reflectance spectra. *Appl Spectrosc. 43(5),* 772-777.

Bijlsma, S.. Boelens, H.F.M., Smilde, A.K. (2001). Determination of rate constants in second-order kinetics using UV-visible spectroscopy. *Appl. Spec., 55(1),* 77-83.

Chen, W., Biegler, L.T., Garcia-Munoz, S. (2016). An Approach for Simultaneous Estimation of Reaction Kinetics and Curve Resolution from Process and Spectral Data, *Journal of Chemometrics, 30,* 506-522.

Chen, W., Biegler, L.T., Garcia-Munoz, S. (2018). Kinetic Parameter Estimation based on Spectroscopic Data with Unknown Absorbing Species. *AIChE Journal, 64*, 3595-3613.

De Juan, A., Tauler, R. (2006). Multivariate curve resolution (MCR) from 2000: progress in concepts and applications. *Crit. Rev. Anal. Chem. 36 (3-4),* 163-176

De Juan, A., Jaumot, J., Tauler, R. (2014). Multivariate curve resolution (MCR). Solving the mixture analysis problem. *Anal. Methods, 6,* 4964-4976.

de la Peña, F., Fauske, V.T., Burdet, P., Prestat, E., Jokubauskas, P., Nord, M.,… Skorikov, A. (2018). hyperspy/hyperspy v1.4.1, Zenodo. http://doi.org/10.5281/zenodo.1469364

Hart, W.E., Laird, C.D., Watson, J.-P., Woodruff, D.L., Hackebeil, G.A., Nicholson, B.L., Siirola, J.D. (2017). Pyomo –

[1] https://github.com/salvadorgarciamunoz/kipet

Optimization Modeling in Python, Second Edition. 67. Springer.

Jaumot, J., de Juan, A., Tauler, R. (2015). MCR-ALS GUI 2.0: new features and applications. *Chemom. Intelli. Lab. Sys., 140(15),* 1-12

Lawton, W.H., Sylvestre E.A. (1971). Self modelling curve resolution. Dent. Tech. 13(3). 617-633.

Maeder, M., Zuberbuhler, A.D. (1990). Nonlinear least-square fitting of multivariate absorption data. Anal. Chem., 62(20), 2220-2224.

Maleki M.R., Mouazen A.M., Ramon H., De Baerdemaeker J. (2007) Multiplicative scatter correction during on-line measurement with near infrared spectroscopy. *Biosyst Eng. 96(3),* 427-433.

Marquardt D.W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *J SIAM., 11(2)*, 431-441.

Nicholson, B., Siirola, J.D., Watson, J.P., Zavala, V.M., Biegler, L.T. (2018). pyomo.dae: a modeling and automatic discretization framework for optimization with differential and algebraic equations. *Math. Prog. Comp., 10(2)*. 187-223.

Pirnay, H., Lopez-Negrete, R., Biegler, L.T. (2012). Optimal sensitivity based on IPOPT. *Math Prog Comp., 4(4),* 307-331.

Sawall, M., Borner, A., Kubis, C., Selent, D., Ludwig, R., Neymeyr, K. (2012). Model-free multivariate curve resolution combined with model-based kinetics: algorithm and applications. *J Chemom., 26(10)*, 538-548.

Savitzky, A, Golay, M.J.E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Anal Chem.*, 36(8), 1627-1639.

Stewart, W.E., Caracotsios, M., Sorenson, J.P. (1992). Parameter estimation from multiresponse data. *AIChE J., 38(5),* 641-650.

Stewart, W.E., Caracotsios, M. (2008). Computer-Aided Modeling of Reactive Systems. John Wiley and Sons, Inc.

Tauler R. (1995). Multivariate curve resolution applied to 2nd order data. *Chem. Intell. Lab. Syst.* 30(1), 133-146.

Thierry, D. M., Biegler, L.T. (2018). Dynamic real-time optimization for CO_2 capture process. *AIChE J, to appear*

Wächter, A., Biegler, L.T. (2006). On the implementation of a primal-dual interioir point filter line search algorithm for large-scale nonlinear programming. *Math Prog, 106(1),* 25-57.

Wu, S., McLean, K.A.P., Harris, T.J., McAuley, K.B. (2011). Selection of optimal parameter set using estimability analysis and MSE-based model-selection criterion. *Int J of Advanced Mechatronic Systems, 3(3),* 188 – 197.

Yao, K.Z., Shaw, B.M., McAuley, K.B., Bacon, D.W. (2003). Modeling ethylene/butene copolymerization with multi-site catalysts: parameter estimability analysis and experimental design, *Polym. React. Eng., 11 (3),* 563-588.