A Parameter Estimation Method for Stiff Ordinary Differential Equations Using Particle Swarm Optimization

William Arloff, Dr. Karl Schmitt, Dr. Luke Venstrom

November 2015

Contents

Abstract

Finding coefficients for stiff ordinary differential equation (ODE) models with unbounded solution spaces has traditionally been a difficult task. This is because the standard derivative based methods of solving stiff ODE's cannot handle undefined regions. We propose a two step method for finding coefficients to match a stiff ODE model to experimental data. The first step creates a polynomial approximation of the experimental data and then uses Particle Swarm Optimization to obtain initial solutions for the coefficients. We choose PSO because of its ability to overlook regions of undefined, or poorly defined coefficients in its search. After reasonably close coefficients have been determined, we then use traditional ODE solvers to solve the ODE. We demonstrate this method on the shrinking core model for finding activation energies, and pre-exponential factors in the reduction in state of Cobalt Oxide.

1 Introduction and Background

1.1 Introduction

In this paper we will develop a technique which will attempt to be used to find parameters for very stiff ordinary differential equations without requiring initial estimates. Stiff ordinary differential equations (ODE's) arise in a variety of contexts such as fluid dynamics [2], thermochemical reactions [4], and chemical kinetics [13]. Stiffness arises in these fields due to the existence of both fast and slow dynamics within the phenomena investigated. Traditionally finding parameters for stiff ODE's is very difficult. Here we propose a method which allows researchers with experimental data best modeled with a stiff ODE to determine model parameters for these equations without any prior knowledge or estimates.

We start by providing a motivating application problem and background information on stiff ODE's, typical parameter estimation techniques, and particle swarm optimization. The method section details the solution procedure including the polynomial approximation of experimental data, use of particle swarm optimization, and solving ODE's once parameters are found. We then apply this procedure to the motivating problem to obtain parameters for a shrinking core model of a thermo-chemical reaction with only observed reaction data. We conclude with a discussion on the more general application of the procedure and what additional improvements can be made.

1.2 A Motivating Application Problem

A current interest to researchers in the field of solar chemistry is the solar-driven reduction of Cobalt Oxide. This solar-driven reduction of Co3O4 is a key step in a thermochemical process for producing hydrogen from water in a more efficient, cost effective, and environmentally friendly manner [10, 11]. The addition of reduced Co3O4 reduces the electricity required during electrolysis by up to 80 percent [5].

The shrinking core model is commonly applied to gas-solid reactions in which the solid does not change its size. The reduction reaction has three layers in the shrinking core model. First, the metal oxides is reduced to a lower oxidation state between the core and the shell via an "interface reaction." Second, oxygen diffuses through the shell to the surface [7]. Lastly the oxygen is transferred away from the surface in the via diffusion. The interaction between these three processes is modeled by Equation 1.

$$\frac{d\alpha}{dT} = \frac{1}{\beta}\left[\frac{1}{10^{A_g}\exp\left(\frac{-E_{a,g}}{RT}\right)} + \frac{2\left((1-\alpha)^{-\frac{1}{3}}-1\right)}{10^{A_s}\exp\left(\frac{-E_{a,s}}{RT}\right)} + \frac{(1-\alpha)^{-\frac{2}{3}}}{3\times 10^{A_c}\exp\left(\frac{-E_{a,c}}{RT}\right)}\right]^{-1} \tag{1}$$

Within each of the three components of the model there are two unknown coefficients, the activation energy and pre-exponential factor. In total there are six unknowns in our model: $E_{a,g}$; $E_{a,s}$; $E_{a,c}$; A_g; A_s; A_c. Previous work has found values for the pre-exponential factors and activation energies by testing the reaction in extremely controlled isothermal settings [5]. However, it is not always possible to perform iso-thermal experiments to find these values. Being able to find these values from non-isothermal data will improve the quality of the estimate of the parameters because it includes a larger range of T.

1.3 Stiff Ordinary Differential Equations and Standard Methods of Solving Them

Ordinary differential equations, for which numerical methods must take small step sizes because nearby solutions significantly vary, are considered *stiff* ordinary differential equations [3]. Smaller step sizes in numerical methods require more computational time and resources [14]. Traditional methods of searching for coefficient values in stiff ODE's still require the solvers to find feasible approximations because many of them require the evaluation of the Jacobian. This is not always possible without some initial, reasonable, guesses. Furthermore, depending upon the landscape of the feasible region for coefficient values, even reasonable guesses may lead to nearby infeasible values.

Optimization methods unable to deal with this non-linearity in the solution space may fail entirely. Even when using numerical methods capable of solving stiff ODE's, like implicit Runge-Kutta methods [12], a poor choice of coeffi-

cients may leave the problem infeasible or unsolvable. The methods for finding parameters given experimental data for non-linear stiff ODE's involve a similar approach to the one we propose. Most standard methods involve using an optimization technique, typically one which finds the Jacobian, combined with a step which solves the function where the differential equations are implemented. The method then calculates the cost function of the difference in fit. Other optimization functions commonly used for the parameter search procedure include the Levenberg-Marquardt algorithm [9, 8].

1.4 Particle Swarm Optimization

We chose particle swarm optimization to search for coefficient values because of its ability to disregard poor or infeasible solutions and traverse the search space efficiently without the evaluation of the Jacobian. Particle Swarm Optimization (PSO), is a form of optimization that was developed by James Kennedy and Russell C. Eberhart as a way to mathematically model the behavior of schooling fish and flocking birds [6]. In PSO, particles are possible solutions in a search space which are evaluated according to a fitness function that is either minimized or maximized. For more detailed information on PSO see [1]. PSO is a very good heuristic for searching difficult search spaces with many minimums and maximums because its swarming behavior allows ample diversification. [1] Aside from its ability to search spaces thoroughly, PSO can be easily implemented with fewer parameter adjustments compared to other techniques of optimization like genetic algorithms.

2 Methods

To determine unknown coefficients for a stiff ODE we propose using a heuristic search algorithm with a variable evaluation method. For the overall search al-

gorithm, we used particle swarm optimization 1.4. The novelty in our approach lies in using PSO to find initial coefficients. We use these coefficients to evaluate the stiff ODE against the derivative of an approximate polynomial fit of the experimental data. Once a set of coefficients is found for which the ODE model directly produces a similar curve to the derivative of the polynomial regression, we implement standard ODE solvers to numerically solve the ODE. At this stage of the process standard ODE solvers become effective since the close estimates for the coefficients provide the necessary starting point. Recall from 1.3 and 1.4 that some stiff ODE's can have un-differentiable regions within the feasible region where standard methods of solving stiff ODE's break.

The remainder of this section details our general method of solving for stiff ODE's with regions of infeasibility and recorded experimental data.

2.0.1 Notation

Mathematically, we choose $\vec{x}$ to denote the independent variable's experimental data points and $\vec{y}$ to denote the dependent variable's experimental data points. The relationship between $\vec{x}$ and $\vec{y}$ represents the experimental reaction to be modeled by an ODE with unknown coefficients. $f(x)$ will describe the nonlinear relationship between our independent variable $\vec{x}$, and our dependent variable $\vec{y}$, to be defined in more detail later.

2.1 Polynomial Approximation

Finding the derivative of the polynomial approximation of the experimental data will allow us to get an approximation of the ODE function. We chose polynomial approximation because it gives us a simple, differentiable, polynomial that approximately matches the experimental data. This approximation is crucial to our method because its derivative gives a curve which can be compared directly

to the output from the ODE. Standard regression on the experimental data was used to obtain the polynomial function $f(x)$.

$$f(x) = \beta + \beta_1 + \beta_2 x^2 + ... + \beta_k x^k \tag{2}$$

Where β_k are the coefficients of the polynomial regression of degree k. This polynomial function is easily differentiated giving us an approximation for $\frac{dy}{dx}$.

2.2 Fitness

Now that we have an approximation of $\frac{d\vec{y}}{d\vec{x}}$ to compare against. We create a generic function which represents the ODE. Our generic function accepts the vectors $\vec{y}$, $\vec{x}$ and $\vec{c}$ as inputs, with $\vec{c}$ representing the unknown coefficients for the ODE. The unknown coefficients in $\vec{c}$ is what the PSO software will find values of.

$$\frac{d\vec{x}}{d\vec{y}} = S(\vec{x}, \vec{y}, \vec{c}) \tag{3}$$

A fitness (or objective) function is required for the optimization process. We use the standard RMSE to compare the to compare the output from $S(\vec{x}, \vec{y}, \vec{c})$ and $f'(x)$.

$$r = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f'(\vec{x}) - S(\vec{y}_i, \vec{x}_i, \vec{c}))^2} \tag{4}$$

Where n is the number of data points in the vectors $\vec{x}$ and $\vec{y}$. This function 4 is evaluated for each particle p in the swarm. In practice, the fitness function is implemented to accept a vector of particles we denote $\vec{p}$. This vector contains different potential solutions for the coefficients in $\vec{c}$.

The fitness function also dictates when to switch to using a standard ODE solver. Given the assumed stiff nature of the ODE we strive to find $\vec{x}$ which

is close enough to allow traditional ODE solvers to work and find a solution. It is up to the practitioner to define the RMSE threshold for switching fitness functions.

2.3 Implementation of Particle Swarm Optimization

Particle swarm optimization was used in conjunction with the polynomial approximation of the experimental data to generate approximate coefficient values for the ODE. A single set of unknown parameters for the ODE is considered a particle. Our search space consists of all possible coefficients $\vec{c}$. The PSO algorithm attempts to find the set of coefficients, $\vec{c}$ with the lowest r. We optimize to the minimum RMSE because we want the smallest difference between $S(\vec{x}, \vec{y}, \vec{c})$ and $f'(\vec{x})$.

We implement PSO because traditional methods for solving ODE's will break due to the stiffness. PSO gives us the ability to overlook very poor parameters which lead to undefined ODE solutions. PSO does this by ignoring bad or infeasible parameters and moving particles towards regions of better fitness values. Traditional single search methods for solving an ODE may not be able to find any solution in spaces of poor parameters. Overlooking these poor parameters allows us to explore the search space more thoroughly looking for more feasible parameters.

2.4 Generalized Method for Parameter Estimation

We now give a broad summary of the method and how its parts combine. Algorithm 1 provides a pseudo-code version of the method.

The algorithm starts with the polynomial approximation of the experimental data. We then take the derivative of this polynomial regression obtaining $f'(x)$. After the derivative is taken, we define a function holding the ODE which accepts

experimental data points along with the vector of unknowns, $\vec{c}$. This function encompassing the ODE is the function we denote as $S(\vec{x}, \vec{y}, \vec{c})$. PSO is now implemented to search for the values of $\vec{c}$. PSO evaluates potential solutions for $\vec{c}$ using the fitness function which uses the root mean-squared error to compare the outputs of $S(\vec{x}, \vec{y}, \vec{c})$ and $f'(x)$. PSO continues its search until there are values for $\vec{c}$ that generate corresponding r values which below a tolerance threshold. This threshold is determined by the user and marks when the $\vec{c}$ values are considered acceptable enough to pass onto a standard ODE solver. Once this threshold is met, PSO continues to search, comparing the output of the ODE solver to the experimental data.

Algorithm 1: General Method

Data: Experimental Data
Polynomial Approximation
$\vec{y} \approx f(\vec{x}) = \beta + \beta_1 + \beta_2 x^2 + ... + \beta_k x^k$
$f(x) \Rightarrow f'(x)$ *find derivative of polynomial regression*
$\frac{dx}{dy} =$
$S(\vec{x}, \vec{y}, \vec{c})$ *Define ODE as a function with input for unknown parameters* $\vec{c}$
while *r for any 250 particles* ≥ 0.003 **do**
  Generate 250 new particles $\vec{p}$ *containing vector* $\vec{c}$ during PSO Stage
  Input vector $\vec{c}$ *in* $S(\vec{x}, \vec{y}, \vec{c})$
  Calculate for all particles $r = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f'(\vec{x}) - S(\vec{y}_i, \vec{x}_i, \vec{c}))^2}$
Choose 25 particles randomly from the 250 particles generated by PSO
Execute PSO again with fitness function optimizing to ODE solver results

The method developed, as generalized here, accommodates finding parameters of any stiff ODE with experimental data. Given a stiff ODE that is too stiff for standard stiff ODE solvers, we suggest this method, in conjunction with accurate experimental data, to provide a fast and inexpensive way to find the parameters of the ODE.

3 Application to Solid-State Kinetics Problem

We now apply our method to our original motivating problem, parameter estimation for the reduction of cobalt oxide under the shrinking core model. The experimental data we use for our method is the reduction of cobalt oxide as a percentage over temperature. As previously stated the shrinking core model has been proposed to model this reaction and is a stiff ODE with regions of infeasibility. Using this application problem to validate our method, we searched for the pre-exponential factors and activation energies of this reaction. What makes the shrinking core model significant to our research is its high degree of stiffness in terms of the feasible solution regions. The pre-exponential factors seen in Equation 1 ($E_{a,g}$, $E_{a,s}$, or $E_{a,c}$) are all multiplied by -1 and are in the exponents of e. Very high values for any one of $E_{a,g}$, $E_{a,s}$, or $E_{a,c}$ can force the denominator of a term to approach zero if the activation energies A_g, A_s, and A_c are not high enough to counteract them. We note that standard methods of finding an initial starting point will have great difficulty navigating this stiffness in our example.

3.1 Implementation

To show the implementation to the application problem we first equate the method variables to the application variables. The temperature, denoted T, is our experimental independent variable denoted as $\vec{x}$ in the methods section. The variable α, the percentage of the reactive core that has been reduced in state corresponds to the dependent variable $\vec{y}$ from the methods section. The shrinking core model is used to predict the change in α over the change in T. The full equation can be seen below from 1.2 with the generalized variables repeated for convenience.

$$S(\alpha, T, \vec{c}) = \frac{d\alpha}{dT} = \frac{1}{\beta}\left[\frac{1}{10^{A_g}\exp\left(\frac{-E_{a,g}}{RT}\right)} + \frac{2\left((1-\alpha)^{-\frac{1}{3}}-1\right)}{10^{A_s}\exp\left(\frac{-E_{a,s}}{RT}\right)} + \frac{(1-\alpha)^{-\frac{2}{3}}}{3\times 10^{A_c}\exp\left(\frac{-E_{a,c}}{RT}\right)}\right]^{-1}$$

Where R is the universal gas constant. The variables $E_{a,g}$, $E_{a,s}$, $E_{a,c}$ are the pre-exponential factors and A_g, A_s, A_c are the activation energies we are searching Together these make up the vector $\vec{c}$ defined in the methods section.

Before beginning our method, we clean up the experimental data measuring α over T by removing post and pre-experimental readings. False readings can warp the polynomial regression causing a loss in accuracy while finding parameters. Next we take a standard regression to get a differentiable polynomial.

$$\vec{\alpha} \approx f(\vec{T}) = \beta + \beta_1 + \beta_2 T^2 + ... + \beta_k T^k \tag{5}$$

We then take the derivative of the polynomial function $f(\vec{T})$ to get the function:

$$f(\vec{T}) \Rightarrow f'(\vec{T}) \tag{6}$$

Figure 1 shows the experimental data points in relation to $f(\vec{T})$. We see that the function maps almost exactly to the experimental data.

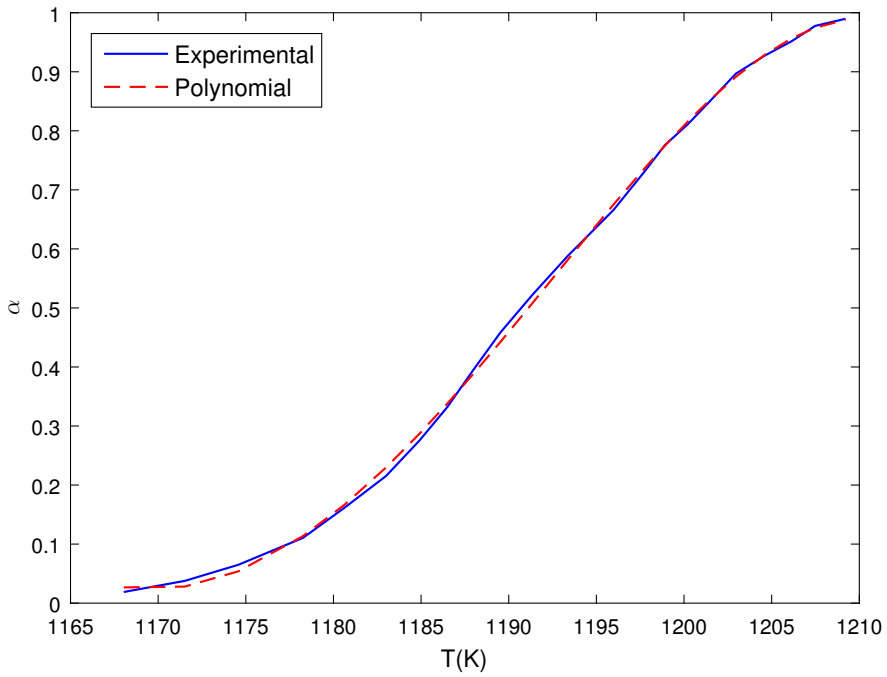

Figure 1: Experimental Data Vs. Polynomial Regression

This function, $f'(T)$, will be compared to the function $S(\vec{\alpha}, \vec{T}, \vec{c})$ using the root mean-squared error function as given in 7.

$$r = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f'(T) - S(\vec{\alpha}_i, \vec{T}_i, \vec{c}))^2} \tag{7}$$

The search algorithm (PSO) will then generate values for $\vec{c}$ which attempt to minimize r. When the fitness values for the coefficient vectors pass below the threshold 0.003 the fitness function switches over to comparing results from $f'(\vec{T})$ to the standard ODE solver and the function $S(\vec{\alpha}, \vec{T}, \vec{c})$.

4 Results

Implementing our method to the shrinking core model yielded linear patterns of coefficient possibilities instead of approximate numbers. We initially predicted each of the coefficients to be approximately similar. After running our model 47 times to achieve results we found that the coefficients were in a wide range spanning the bounds we had placed upon them. Figure 2 below the box-plots for each of the coefficients. The bounds were between 0 and 100 for the activation energies, and between 100 and $1 \times e^7$ for the pre-exponential factors. The box plots in figure 2 show the range of the 47 outputs.

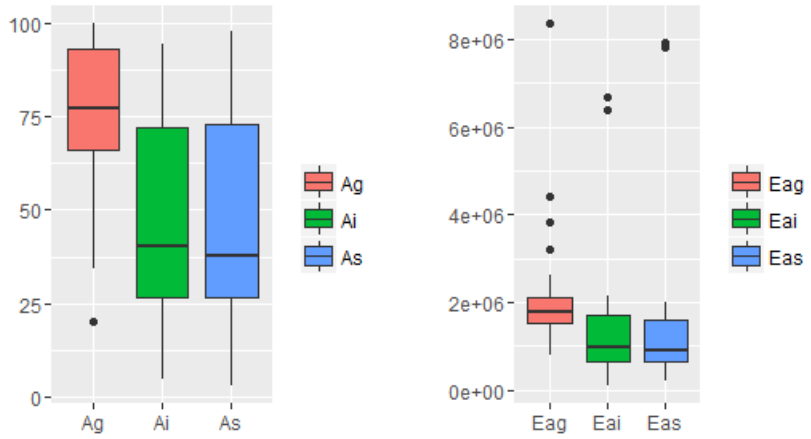

Figure 2: Box and Whisker Plots of Resulting Coefficients

The box plots show us there is a wide variation in the ranges of possible coefficients found by the ODE solver. We hoped to see tight ranges for the box plots signifying a small range of possibilities for the coefficients, thus giving a single number for the resulting coefficient. What we found instead were clear linear relationships with similar RMSE values between the corresponding coefficients for each section of the ODE (i.e. Eag and Ag). The scatter plots of the relationships between the results in Figure 3 show these linear relationships.

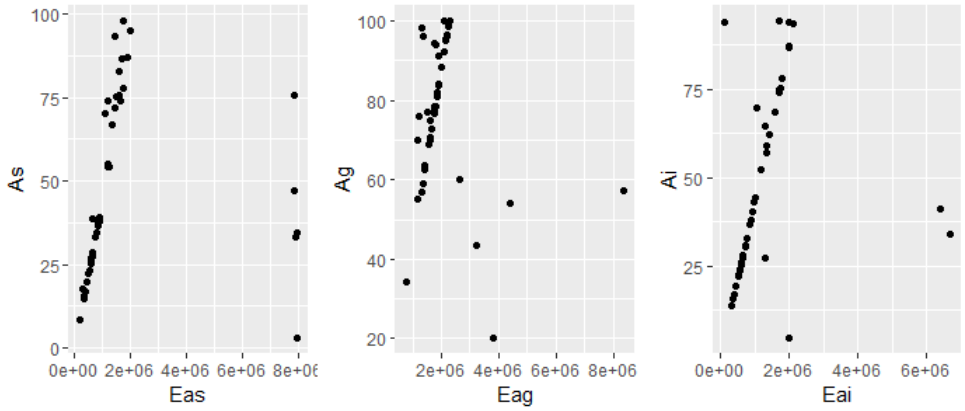

Figure 2: Scatter plots Showing Linear Relationship between Coefficients

When examining the quality of each of the points, the points not along these linear lines had higher RMSE values than those points not on the line. The points on the line all had very similar RMSE values so no specific regions of the line could be pointed out as better than another region. The results show us that our method is capable of finding feasible regions of answers for the coefficients in our application problem.

4.1 Implications for Solid-State Kinetics

The linear regions of feasibility give researchers a more narrowed search space when looking for the set values for the coefficients. The method as a whole could be used by researchers attempting to find regions of feasibility for Solid-State Kinetics applications with stiff ODEs. Our method also provides an inexpensive way to find these regions of feasibility in stiff ODEs with experimental data. Researchers now will be able to search these linear relationships between coefficients for more approximate numbers.

5 Conclusion

Given a stiff ODE model with unknown coefficients and a set of experimental data we were able to implement a 2-stage search for feasible values. In the first stage, implementing a polynomial regression of the experimental data, then using the derivative of this polynomial provided us with a tool to compare the outputs of our ODE by using the root mean squared error as a metric of similarity. In the first stage, the PSO software attempts to find coefficients with low corresponding RMSE values. This allows the PSO and ODE solvers in the second stage to search a reduced region of feasible solutions. This feasibility region was unattainable with traditional analytical techniques. Within this feasible region found in the first stage, standard ODE solvers in conjunction with PSO were able to output reasonable feasible solutions to our stiff ODE with RMSE values under 0.03. The solutions given by the ODE solver in the solid state kinetics problem were scattered but gave us valuable information about the relationships between the coefficients. These linear relationships, instead of giving us single values like we had hoped for, have instead given us a smaller search space which we can focus on for later searches. This technique could easily be applied to other stiff ODEs to find regions of feasibility given a vast search space. In conclusion the method we have proposed has given us solutions to a very stiff ODE, but in our application problem, despite giving us valuable information on the linearity of the relationships between the coefficients, our results did not give us singular figures.

5.1 Limitations

Our method is capable of finding feasible regions of coefficients for very stiff ODE's but has not been shown to predict specific points. The method is only shown to be able to narrow the search space. The method also relies on the

accuracy of the derivative of the polynomial regression, because of this the true accuracy of trying to fit to experimental data is limited. This model should only be accepted if traditional methods for finding coefficients for stiff ODE's have failed.

5.2 Future Research

Future research on the method itself would be focused on running the model focusing on the linearity between the corresponding activation energies and pre-exponential factors in our application problem. To do this we propose fitting a line to the linear relationship in the coefficients we have found thus far and re-using our method to search on that specific line. Finding points on these lines and determining their RMSE values could help us determine points on this line with lower corresponding RMSE values than others. Doing this would lead us to more specific regions with the true values of the activation energies and the pre-exponential factors.

References

[1] Edmund Burke and Graham Kendall. *Search methodologies: introductory tutorials in optimization and decision support techniques*. English. OCLC: 861617253. 2013. ISBN: 978-1-4614-6940-7. URL: http://dx.doi.org/10.1007/978-1-4614-6940-7 (visited on 06/23/2016).

[2] M. H. Carpenter et al. "Fourth-Order Runge–Kutta Schemes for Fluid Mechanics Applications". en. In: *Journal of Scientific Computing* 25.1 (Oct. 2005), pp. 157–194. ISSN: 0885-7474, 1573-7691. DOI: 10.1007/

s10915-004-4637-3. URL: http://link.springer.com/10.1007/s10915-004-4637-3 (visited on 06/23/2016).

[3] Steven C. Chapra and Raymond P. Canale. *Numerical methods for engineers*. 6th ed. Boston: McGraw-Hill Higher Education, 2010. ISBN: 978-0-07-340106-5.

[4] Francesco Contino et al. "Simulations of Advanced Combustion Modes Using Detailed Chemistry Combined with Tabulation and Mechanism Reduction Techniques". en. In: *SAE International Journal of Engines* 5.2 (Apr. 2012), pp. 185–196. ISSN: 1946-3944. DOI: 10.4271/2012-01-0145. URL: http://papers.sae.org/2012-01-0145/ (visited on 06/23/2016).

[5] Jesse Fosheim and Luke Venstrom. *The Solid-State Kinetics for the Reduction of Cobalt Oxide in Argon*.

[6] J. Kennedy and R. Eberhart. "Particle swarm optimization". In: 4 (Nov. 1995), 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968.

[7] Octave Levenspiel. *Chemical reaction engineering*. 3rd ed. New York: Wiley, 1999. ISBN: 978-0-471-25424-9.

[8] D. Marquardt. "An Algorithm for Least-Squares Estimation of Nonlinear Parameters". In: *Journal of the Society for Industrial and Applied Mathematics* 11.2 (June 1963), pp. 431–441. ISSN: 0368-4245. DOI: 10.1137/0111030. URL: http://dx.doi.org/10.1137/0111030 (visited on 07/27/2016).

[9] Jorge J. Moré. "The Levenberg-Marquardt algorithm: Implementation and theory". In: *Numerical Analysis: Proceedings of the Biennial Conference Held at Dundee, June 28–July 1, 1977*. Ed. by G. A. Watson. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 105–116. ISBN: 978-3-540-35972-2. URL: http://dx.doi.org/10.1007/BFb0067700.

[10] R. Palumbo et al. "Solar thermal decoupled water electrolysis process I: Proof of concept". en. In: *Chemical Engineering Science* 84 (Dec. 2012), pp. 372–380. ISSN: 00092509. DOI: 10.1016/j.ces.2012.08.023. URL: http://linkinghub.elsevier.com/retrieve/pii/S0009250912005180 (visited on 04/19/2016).

[11] Thomas Pregger et al. "Prospects of solar thermal hydrogen production processes". en. In: *International Journal of Hydrogen Energy* 34.10 (May 2009), pp. 4256–4267. ISSN: 03603199. DOI: 10.1016/j.ijhydene.2009.03.025. URL: http://linkinghub.elsevier.com/retrieve/pii/S0360319909004352 (visited on 04/19/2016).

[12] C. Runge. "Ueber die numerische Auflösung von Differentialgleichungen". In: *Mathematische Annalen* 46.2 (1895), pp. 167–178. ISSN: 1432-1807. DOI: 10.1007/BF01446807. URL: http://dx.doi.org/10.1007/BF01446807.

[13] A. Sandu et al. "Benchmarking stiff ODE solvers for atmospheric chemistry problems II: Rosenbrock solvers". In: *Report on computational mathematics* 85 (1996). URL: http://oai.cwi.nl/oai/asset/125/0125A.pdf (visited on 06/23/2016).

[14] Lawrence F. Shampine and Charles William Gear. "A user's view of solving stiff ordinary differential equations". In: *SIAM review* 21.1 (1979), pp. 1–17. URL: http://epubs.siam.org/doi/abs/10.1137/1021001 (visited on 06/23/2016).