TranKin

A software package for enhancement of time resolution in transient kinetic methods

Version 1.1.0

Soheil Soltani
Division of Chemical Reaction Engineering,
Department of Chemical Engineering
Chalmers University of Technology, Göteborg, Sweden
http://www.chalmers.se/en/staff/Pages/soheil-soltani.aspx

December 21, 2015

License

TranKin is licensed under the terms of the MIT license:

To Mozhan, Jon and Oana

Preface

Analysis of transient kinetic data requires a fairly high temporal resolution. For instance, kinetic analysis of ultrafast systems involves processes with time scales in the order of pico- or femto seconds, which necessitates achieving an extremely high resolution in time. Therefore, the response time of the analyzers is generally desired to be lower than the kinetic time scale to obtain the time-resolved information of the system. However, dispersion effects in various parts of the experimental setup can smear the concentration signal so that the response time of the detector becomes larger than the time scale of the kinetic processes. Under these circumstances, temporal resolution can be a major limitation of transient kinetic methods.

A signal restoration method is therefore required in order to retrieve the true response of the system as the solution to a one-dimensional deconvolution problem. However, the inverse problem of finding the true signal is an ill-posed problem in the presence of experimental uncertainties (noise), since arbitrarily small perturbations in the measurements result in arbitrarily large oscillation in the solution. Therefore, a regularization method is required to filter out these magnified perturbations. TranKin is a software package that has been primarily developed to solve the above-mentioned problem using the Tikhonov regularization method, yet it can be regarded as a general-purpose solver for one-dimensional deconvolution problems.

Chapter 1 of this monograph gives a short introduction to the topic, which is followed by the physics of hydrodynamic dispersion briefly recapped in Chapter 2. Chapter 3 summarizes the mathematical background of the inverse problem. Finally, the software package is described in Chapter 4, which is accompanied by a short tutorial on how to use the package.

Acknowledgments. I am grateful to my advisers at Chalmers, Prof. Bengt Andersson and Doc. Ronnie Andersson, who have always inspired me and supported my research. My deepest thanks go to Mozhan Soltani and Jon Kristensen for their invaluable assistance and motivating discussions. And last, but foremost, Goli and Oana for their endless love.

Soheil Soltani
Gothenburg, August 2014

Contents

1 Introduction

Despite the generality of the mathematical background and numerical analysis of inverse problems, the features of the practical issues where an inverse problem arises can be very different from case to case. The main objective of this work is to address hydrodynamic dispersion and time-lag in laboratory reactor systems in order to enhance temporal resolution of transient measurements.

Perhaps the problem with smeared concentration measurements is first understood when running a calibration-like experiment in an empty reactor setup, in which a known concentration signal (often a step change from 0 to a certain concentration of a specific gas) is dosed at the inlet. In the absence of any catalytic or reactive material that can interact with the gas phase species, the same step signal is expected to be observed as the measured signal in the outlet. Nevertheless, the result is often a monotonically increasing function of time that slowly reaches the inlet concentration after certain elapsed time. Hydrodynamic dispersion is known to be responsible for the observed phenomenon as it gives rise to a distribution of the residence time of fluid elements. Since signal distortion will also affect the results of experiments including chemical reactions, it is necessary to characterize the reactor setup with respect to these smearing effects in order to properly interpret the results of transient experiments. To have this fulfilled, there is also a need for a method to retrieve the true response of the reactor system for quantitative data analysis. To this end, TranKin has been developed under a permissive and business-friendly open-source license. While being capable of handling one-dimensional, unconstrained deconvolution problems in general, its prime focus is on solving deconvolution of transient kinetic data. The following sections include a more in-depth description of these issues.

2 Hydrodynamic dispersion

Let us consider a linear, time-invariant (LTI) system that represents a laboratory setup as depicted in Figure 1. The reactor system can be thought of as an operator that maps the input signal, $x(t)$, on to the output observation, $y(t)$.

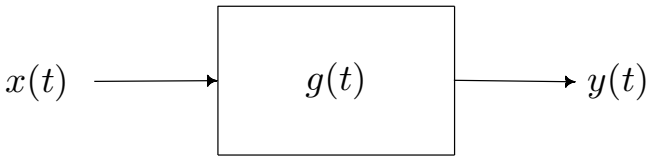

Figure 1: Block diagram of the reactor setup.

We note that the forward problem is governed by the convolution of the

input signal with the transfer function of the system, $g(t)$, according to Eq. 1:

$$\int_a^b g(t-t')x(t')dt' = y(t), \tag{1}$$

where g is the transfer function of the system, x is the unknown input and y is the measured output concentration.

Generally speaking, all the possible sources of signal distortion which have a linear effect on the output signal can be lumped into a transfer function which is also referred to as the kernel of the convolution equation. Therefore, determination of the kernel is an essential part of the characterization study.

As an example, we shall consider the reactor system that has been investigated in an earlier study, with a predominant laminar flow field throughout the setup (the system is normally operated at a volumetric flow rate of 3500 mL/min at NTP) [8]. The major source of dispersion was shown to be the gas cell of the detector. In addition to that, the reactor tube was introduced as the second most important contributor to signal distortion. In the coming sections, characterization of hydrodynamic dispersion in each of these parts of the setup is shortly described.

2.1 Dispersion in tubular parts

In laminar flows through cylindrical tubes, the parabolic velocity profile is the main source of axial dispersion of a concentration signal. Furthermore, molecular diffusion may play a relatively important role under certain conditions. Diffusion in the axial direction contributes to axial dispersion, while diffusion towards the walls helps maintain a more uniform concentration profile. The relative importance of these processes have been thoroughly investigated in earlier studies [1, 2, 3, 9].

The reactor tube in the above-mentioned case study is made of quartz, which is 80 cm long with an inner diameter of 22 mm. The analysis has shown that dispersion in the quartz tube can be described by the Aris-Taylor equation which predicts a dispersion coefficient of $3.34 \times 10^{-3}\ m^2/s$. Consequently, residence time distribution (RTD) of the quartz tube can be described by a Gaussian function with a variance of $0.66\ s^2$. Figure 2 shows the thus-obtained transfer function of the quartz tube, having a mean residence time of $\sim$1.7 s. Assuming a unitary Heaviside function as the input concentration signal and a uniform velocity profile at the inlet of the quartz tube, the evolution of the concentration profile in the outlet can be constructed according to Eq. 1. Figure 3 illustrates the calculated step-response of the quartz tube.

2.2 Dispersion in a gas cell

The analysis of residence time distribution becomes more complex if the geometry differs from a cylindrical tube, and numerical methods are required to calculate the response of the system. In the former study, temporal evolution of a unitary Heaviside function at the inlet of the gas cell was obtained from a computational fluid dynamics (CFD) simulation. The simulated output signal has been differentiated with respect to time to obtain the kernel representing

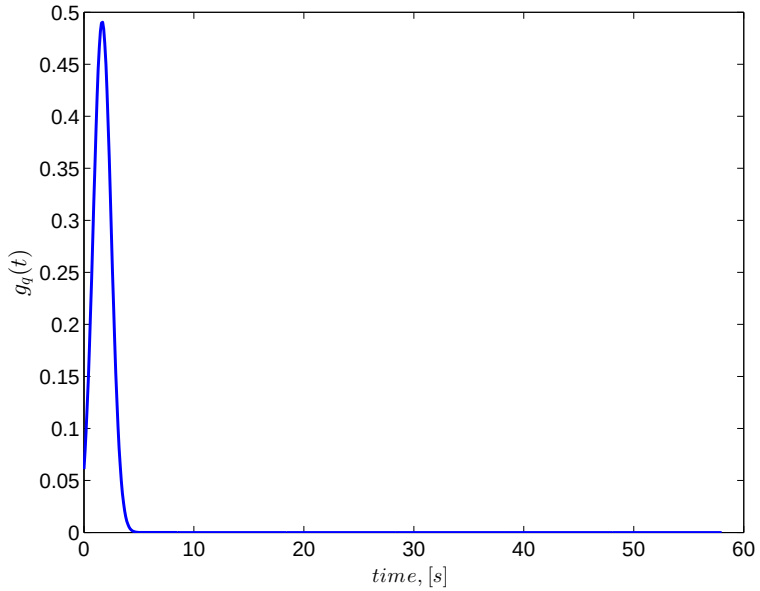

Figure 2: Residence time distribution of the quartz tube.

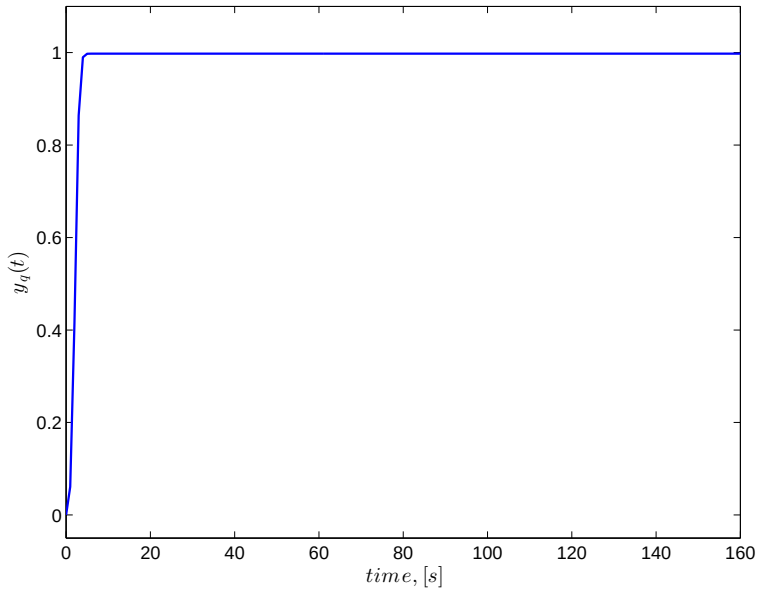

Figure 3: Step-response of the quartz tube.

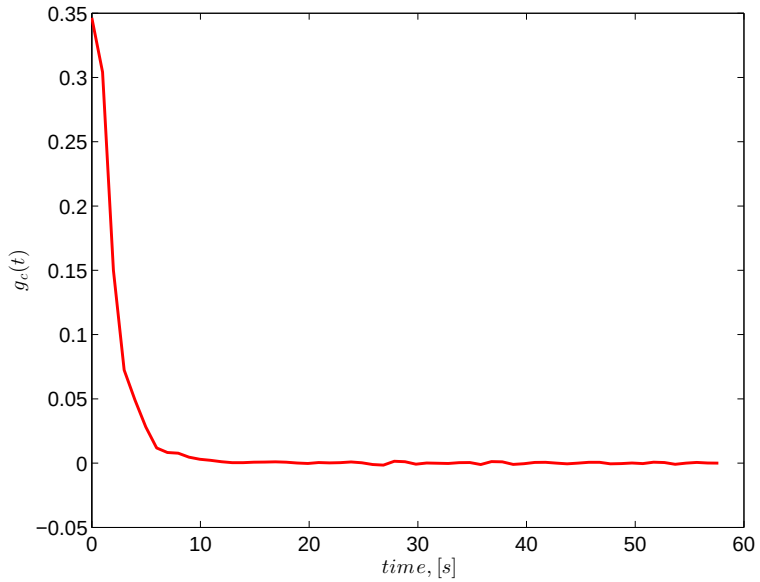

Figure 4: RTD of the gas cell.

the RTD of the gas cell:

$$g(t) = \begin{cases} u(t), & when\ z = \delta(t - t_0), \\ \frac{du(t)}{dt}, & when\ z = H(t), \end{cases} \tag{2}$$

where δ is the Dirac delta function, and H is the Heaviside step function.

Figure 4 depicts the residence time distribution of the gas cell, indicating a more complex flow field compared to a simple pipe system.

3 The inverse problem

In this section, the inverse problem of deducing the unperturbed concentration signal is addressed, and the ill-posed characteristics of the problem are briefly described.

In Section 2, the 1-dimensional convolution operation was introduced through Eq. 1, which is a typical example of the Fredholm integral equation of the first kind. The kernel and the right-hand side of Eq. 1 are assumed to be known, and the unknown signal x is to be deduced such that the following residual norm is minimized:

$$argmin \left\{ \|A_\tau x - y_\delta\|^2 \right\}. \tag{3}$$

Here, A_τ is an approximation to the convolution operator that is derived, in its discretized form, by transforming the kernel into a lower-triangular, toeplitz matrix. y_δ represents the measured signal, and δ, τ denote the error in specifying the right-hand side (i.e. $y(t)$) and the kernel, respectively. However, the encountered inverse problem is known to be ill-posed if the right-hand side (and possibly also the kernel) is perturbed by experimental noise. Under these

circumstances, arbitrarily small perturbations in the measurements result in arbitrarily large oscillations in the calculated solution. Therefore, a regularization method is required in order to dampen out these oscillations [4].

We shall use the Tikhonov regularization method in conjunction with the discrepancy principle for calculating the optimum value of the regularization parameter. The theoretical background has been outlined in previous studies; therefore, only the most important governing equations are mentioned here, and the interested reader is referred to other references for further information [5, 8, 6, 7].

Tikhonov's method introduces a regularization parameter as a weighting factor on minimizing the residual norm versus the solution norm as

$$M^{\alpha}[x] = \|A_{\tau}x - y_{\delta}\|^2 + \alpha\|x\|^2. \tag{4}$$

Consequently, the regularized solution to Eq. 4 reads

$$x_{\eta}^{\alpha} = (A_{\tau}^{*}A_{\tau} + \alpha I)^{-1}A_{\tau}^{*}y_{\delta}, \tag{5}$$

where I is the identity matrix, and the superscript $*$ signifies the adjoint operator. Using singular value decomposition (SVD) of A_{τ}, Eq. 5 can be written as

$$x_{\eta}^{\alpha} = \sum_{i=1}^{m}\left(\frac{\sigma_i}{\sigma_i^2 + \alpha}\right)\gamma_i^T y_i \theta_i, \tag{6}$$

where γ_i, θ_i are the elements of the orthogonal vectors of A_{τ}, and σ_i are the singular values of it. The key parameter that determines the optimum extent of filtering in Eq. 3-5 is the regularization parameter α. The only available method that derives α in correspondence to the experimental uncertainties (i.e. δ, τ) is the generalized discrepancy principle. Let us define discrepancy as

$$\rho_{\eta}^{\kappa}(\alpha) = \|A_{\tau}x_{\eta}^{\alpha} - y_{\delta}\|^2 - \left(\delta + \tau\|x_{\eta}^{\alpha}\|\right)^2. \tag{7}$$

It is desired to find the value of α corresponding to the root of Eq. 7, i.e. $\rho_{\eta}(\alpha) = 0$. It has been shown that the Newton's method can be used to find the root of Eq. 7, and if the kernel is exact, i.e. $\tau = 0$, the convergence of the Newton's method is guaranteed [10]. In case the kernel is experimentally determined, the convergence of the Newton's method is not guaranteed; therefore, an under-relaxation factor ω has been implemented in the software to provide a satisfactory convergence of the algorithm. Nevertheless, the use of a more robust root-finding procedure such as the bisection method is recommended under these circumstances. Derivative of the discrepancy functional is needed in the Newton's routine which can be obtained from

$$\left(\rho_{\eta}^{\kappa}(\alpha_n)\right)' = \left((A_{\tau}^{*}A_{\tau} + \alpha I_n)^{-1}x_{\eta}^{\alpha}, x_{\eta}^{\alpha}\right)\left(\alpha + \frac{\tau\delta}{\sqrt{\|x_{\eta}^{\alpha}\|^2}} + \tau^2\right), \tag{8}$$

where I_n is the identity matrix [10].

Convergence of the Newton's method requires the starting value of the regularization parameter to be fairly close to the final solution. Therefore, the

following lemma has been implemented in the software that attempts to compute a suitable starting guess to initialize the iterations [10]:

$$\bar{\alpha} = \|A_\tau\| \left(\tau + \frac{\sqrt{\tau^2 + (\tilde{\alpha} + \tau^2)(C^2 - 1)} + \tau}{C^2 - 1} \right), \tag{9}$$

where $1 < C < \frac{\|y_\delta\|}{\delta}$; $C = constant$, and $\tilde{\alpha}$ is obtained from $\tilde{\alpha} = \frac{\|A_\tau\|^2 C\delta}{\|y_\delta\| - C\delta}$.

4 Software package

The software can be downloaded from its official repository at:
https://github.com/soheil-soltani/TranKin.

The package consists of three MATLAB files (*.m files): i) TranKin ii) initial_guess and iii) frootf. The dissection of the different parts of the code is given in this section.

For a proper installation, it is important to keep the *.m files and the data file(s) in the same folder.

4.1 TranKin

This is the main function (entry point) that the user should run. It loads the input data, controls the procedure of finding the regularization parameter and computation of the corresponding solution and performs post-processing of the results.

Block 1 initializes the algorithm by requesting the user to provide the name of the data file that contains the time series at which data has been acquired, the measured concentration signal and the kernel. These variables are denoted by t, y and g, respectively. It is important to note that the data file should contain the aforementioned variables as separate columns having the same length. The data file should be a .txt file[1], and it must be located in the same folder as the installation folder of the software.

In block 2, the kernel is first rearranged into a lower-triangular toeplitz matrix denoted with K. Consequently, Eq. 1 can be written in operator form as $y = Kx$. Using singular value decomposition routine (`svd`), the operator K is next decomposed into matrices G, $Sigma$ and T. G and T are matrices with orthonormal columns, and $Sigma$ is a square matrix with non-negative real elements on its main diagonal. The off-diagonal elements of $Sigma$ are excluded so that the resulting vector consists solely of the singular values of the operator K.

The uncertainties (error) in the measurements and the kernel are defined in block 3 by requesting the user to specify the values of δ and τ. The former represents the uncertainties in the measured signal y, and the latter is the error in specifying the kernel.

The procedure of calculating the regularization parameter is controlled in block 4. The user can choose to enter the value of the regularization parameter or to let the software calculate it. It should be noted that the regularization parameter accepts a value on the interval $(0, \infty)$; otherwise, the user is notified with an error message and the program stops.

[1] For other file-formats, the main code must be adapted accordingly.

If the user leaves the calculation of the regularization parameter to the software, he/she has to choose between either the Newton or the bisection method (see block 4-a). If the Newton method is chosen, the user will be asked for the starting point of the root-finding routine in the next step (block 4-b). Again, the user may choose to enter the initial guess from the interval $(0, \infty)$ or to let the *initial_guess* routine calculate it. The subroutine *initial_guess* is explained in the following section.

If the bisection method is chosen, the user will be asked to specify an interval that includes the regularization parameter (block 4-c). This is done by entering two numbers that are enclosed in brackets and are separated by a space: e.g. [first_number second_number]. Alternatively, the user may press <Enter> to let the software calculate a suitable interval.

In the next step, the root-finding procedure *frootf* is called to calculate the regularization parameter, which finalizes block 4. Using the calculated regularization parameter, the regularized solution is computed in block 5. In the final step (block 6), the user is asked to confirm post-processing of the results, which concludes the main routine *TranKin*. The user can choose to plot the results by confirming the message: "Plot the results?[Y/N]...".

4.2 initial_guess

This function includes the implementation of the proposed lemma (see Eq. 9) to closely approximate the initial guess of the regularization parameter [10]. This can be very important with respect to the convergence behavior of the Newton root-finding routine.

4.3 frootf

This function implements both the Newton and the bisection roof-finding schemes for calculating the regularization parameter as the root of the discrepancy functional (Eq. 7).

Block 1 initializes the subroutine. Block 2 ensures that a positive value for discrepancy is calculated using the initial guess for the regularization parameter. This is a necessary step for both root-finding schemes. If a negative discrepancy is encountered, the initial guess is multiplied by a factor 2 until the condition is satisfied. This is confirmed by notifying the user: "Correct starting value was computed". Next, the algorithm distinguishes wether the Newton method has been chosen or the bisection method. In the first case, the user is requested to provide an under-relaxation factor from the interval $(0, 1]$. Under-relaxation factor is strongly dependent upon the features of the problem; however, a trial value between 0.8-0.95 may be recommended. It should be noted that if the entered value is not within the specified interval, an error occurs, and the program stops. In case of convergence problems, the under-relaxation factor should be reduced.

Block 3 contains the implementation of the Newton's formula through the calculation of the discrepancy functional and its derivative. 3-a handles the special case of $\tau = 0$ (i.e. an exact kernel), while 3-b handles a general case ($\tau \neq 0$, i.e. an approximate kernel). Block 4 contains the implementation of the bisection method. In 4-a, the user-specified interval is used to start off iterations. Using the bisection method requires the sign of the function being

investigated to be different at the upper- and the lower-bound of the interval. This condition is checked in block 4-a for $\tau = 0$ and in block 4-b for a more general case of $\tau \neq 0$.

Alternatively, calculating the initial interval can be assigned to the algorithm. This case is handled in part d both for $\tau = 0$ and for $\tau \neq 0$. The procedure is as follows. The algorithm uses the initial guess calculated for the Newton method, for which it was already ensured that a positive value for discrepancy is calculated. Next, the algorithm keeps halving the intial guess until a negative value for discrepancy is encountered. This value is then set to be the lower bound of the interval. The upper bound is obtained by multiplying the lower bound by 2. Thereby, the root is trapped in a close neighbourhood.

Once the algorithm finds the initial interval [a b], the condition of $\rho(a)\rho(b) < 0$ is further controlled to be satisfied. If the algorithm fails to find the inital interval successfully, the user is notified with the following error message: "Incorrect interval. Software failure". The user is kindly requested to report this to the author for further investigations.

After this step, the initial interval is known and the bisection method is iterated until the root of Eq. 7 is found. This concludes the function *frootf*.

After the convergence criterion is met, the user is informed about the number of iterations and the time it took for the subroutine to converge, the initial guess that was used to start off the iterations and the calculated regularization parameter.

4.4 Tutorial

In this section, we shall demonstrate the software in the MATLAB environment using an example consisting of synthetic data. To this end, we start first by defining a time series representing the sequence at which our imaginary data has been acquired:

```
t = 0:1:160;
```

Next, we define a step signal with the same length as t corresponding to the unknown input signal x. We want this signal to start exactly from 0:

```
x = heaviside(t);
x(1) = 0;
```

We assume a well-mixed vessel with a time constant of 3 s as the system through which the inlet stream flows; therefore, the kernel can be defined as:

```
g = 1/3*exp(-t./3);
```

We now define the smeared observations according to the convolution of x and g as the imaginary data:

```
y = filter(g,1,x);
```

We shall further assume an uncertainty of 0.003 to be present in our measurements by superimposing a randomly distributed noise with mean 0 and standard

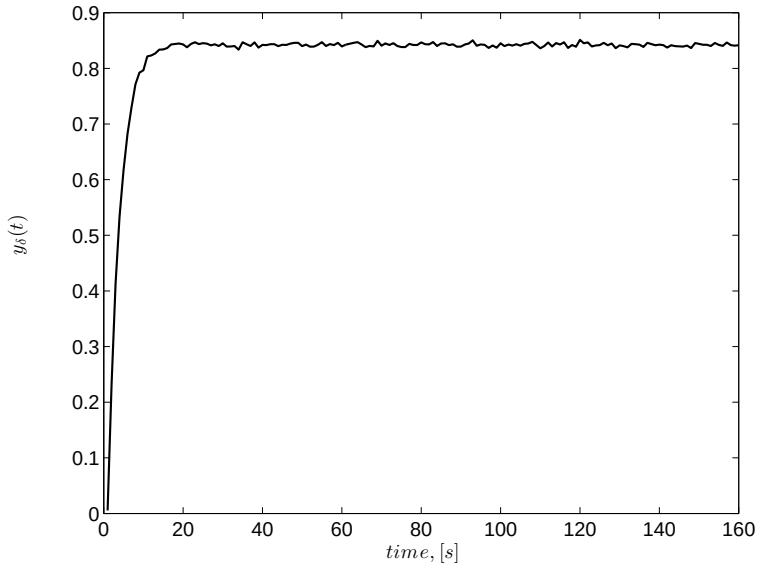

Figure 5: The measured signal corrupted with random noise.

deviation of 0.003:

```
y = y + 0.003.*randn(1,161);
```

This will render the data as depicted in Figure 5. In this elementary example, we let the kernel remain exact. The generated data including the time series, the corrupted signal and the defined kernel are stored as column-vectors in the suplementary data file (i.e. the data.txt file in the software's directory):

```
data = [ t' y' g' ];
save('data.txt', 'data', '-ascii');
```

The software is now called by executing the main code: `TranKin`. As the first step, the name of the data file, i.e. `data.txt`, is supplied to initialize the program. Next, the errors are declared as $\delta = 0.003$ and $\tau = 0$. Let the software calculate the regularization parameter by pressing the "Enter" key for the next enquiry of the algorithm. Choose the Newton method and set the under-relaxation to 0.9 to start off the iterations. The algorithm converges after around 230 iterations, resulting in a regularization parameter of 0.00026. Confirm "`Plot the results?[Y/N]...`" to observe the results. The input signal x is restored as shown in Figure 6, and the convergence monitor as illustrated in Figure 7 indicates a satisfactory solution.

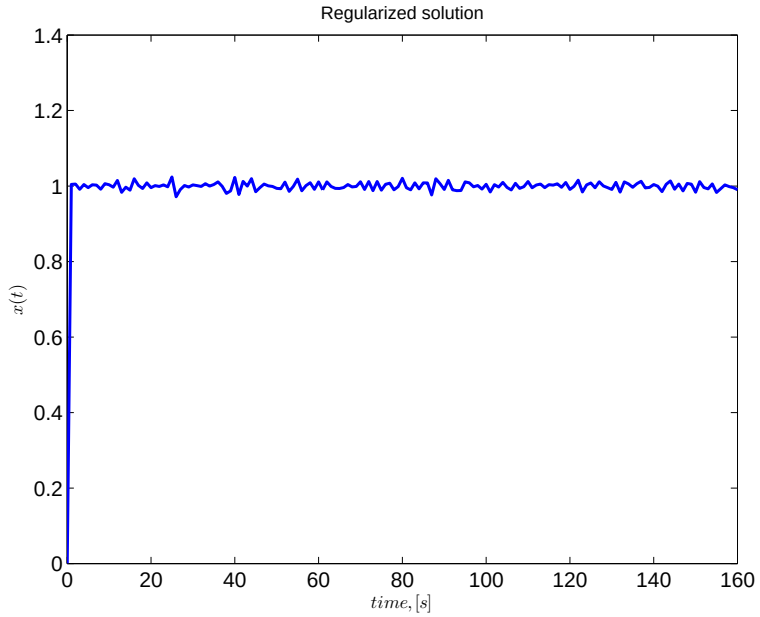

Figure 6: The reconstructed signal.

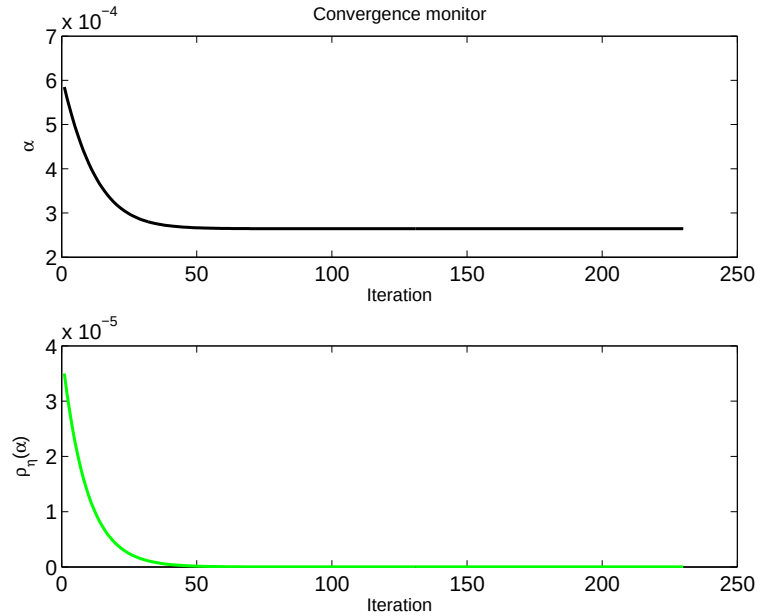

Figure 7: Convergence monitor.

References

[1] Ananthakrishnan.V, Gill, W. N., Barduhn, A. J.: Laminar Dispersion in Capillaries .I. Mathematical Analysis. Aiche Journal. **11**(6), 1063 (1965)

[2] Aris, R.: On the dispersion of a solute in a fluid flowing through a tube. Proceedings of the Royal Society of London Series a-Mathematical and Physical Sciences. **235**(1200), 66–77 (1956)

[3] Bournia, A., Houghton, G., Coull, J.: Dispersion of Gases in Laminar Flow through a Circular Tube. Proceedings of the Royal Society of London Series a-Mathematical and Physical Sciences. **261**(1305), 227 (1961)

[4] Groetsch, C.W.: The theory of Tikhonov regularization for Fredholm equations of the first kind. Pitman, Boston (1984)

[5] Morozov, V.A.: Methods for solving incorrectly posed problems. Springer-Verlag, New York, (1984)

[6] Soltani, S., Andersson, R., Andersson, B.: Enhancement of time resolution in transient kinetics. Chemical Engineering Journal. DOI 10.1016/j.cej.2014.10.098, **264**, 188–196 (2014).

[7] Soltani, S., Andersson, R., Andersson, B.: Time resolution in transient kinetics, in: L. Beilina (Ed.) Inverse Problems and Applications, Springer, ISBN 978-3-319-12498-8 (2015).

[8] Soltani, S., Wang-Hansen, C., Andersson, R., Andersson, B.: CFD characterization of monolithic reactors for kinetic studies. Canadian Journal of Chemical Engineering. DOI 10.1002/cjce.22022, **92**(9), 1570 (2014).

[9] Taylor, G.: Dispersion of Soluble Matter in Solvent Flowing Slowly through a Tube. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences. **219**(1137), 186–203 (1953)

[10] Tikhonov, A.N., Goncharsky, A.V., Stepanov, V.V., Yagola, A.G.: Numerical Methods for the Solution of Ill-Posed Problems. Springer, Dordrecht (1977)