Parameter Estimation Algorithms for Kinetic Modeling from Noisy Data

Fabiana Zama[1], Dario Frascari[2], Davide Pinelli[2], and A.E. Molina Bacca[2]

[1] Department of Mathematics, University of Bologna `fabiana.zama@unibo.it`,
[2] Department of Civil, Chemical, Environmental and Materials Engineering (DICAM), University of Bologna

Abstract. The aim of this work is to test the Levemberg Marquardt and BFGS (Broyden Fletcher Goldfarb Shanno) algorithms, implemented by the matlab functions `lsqnonlin` and `fminunc` of the Optimization Toolbox, for modeling the kinetic terms occurring in chemical processes of adsorption. We are interested in tests with noisy data that are obtained by adding Gaussian random noise to the solution of a model with known parameters. While both methods are very precise with noiseless data, by adding noise the quality of the results is greatly worsened. The *semi-convergent* behaviour of the relative error curves is observed for both methods. Therefore a stopping criterion, based on the Discrepancy Principle is proposed and tested. Great improvement is obtained for both methods, making it possible to compute stable solutions also for noisy data.

Keywords: Parameter Estimation, Non-linear differential models, Quasi-Newton methods, Discrepancy Principle.

1   Introduction

An important topic in many engineering applications is that of estimating parameters of differential models from partial and possibly noisy measurements. For example the removal of pollutants from surface water and groundwater requires the optimization of partial differential models where the dispersion, mass transfer and reaction terms are estimated from data in column reactor experiments [1], [2], [3].

We define here the constrained optimization problem connected to the estimation of a parameter, defined by q, in a differential model represented by $c(u, q)$, named state equation, whose solution $u(q)$ is called state variable:

$$\min_q J(u, q) \quad s.t. \quad c(u, q) = 0 \quad \text{ODE-PDE model}$$

By pointing out the implicit dependence of u on the parameter q, the problem is usually presented in its reduced form $\min_q \widehat{J}(q)$ where $\widehat{J}(q) \equiv J(u(q), q)$ represents the fit to the given data.

By defining the observation operator $\mathcal{C}$, that maps the state variable $u \in U$ into the measurements space Y, we obtain the measurements $y \in Y$. The data fidelity term $\widehat{J}(q)$ is defined as: $\widehat{J}(q) \equiv \|F(q) - y\|$ where $F(q) \equiv \mathcal{C}(u(q))$.

Hence the final problem consists in the minimization of the distance between the data and the computed approximation ($F(q)$) measured in a norm dependent on the model of the data noise. The discrete finite dimensional optimization problem is obtained by defining the vector parameter $\mathbf{q} \in \mathbb{R}^P$ and computing the noisy data $\mathbf{y}^\delta \in \mathbb{R}^N$ by sampling y at N points and adding a noise term.

Depending on the type of noise present in the data the discrete minimization problem can be defined in the suitable L_p norm $1 \le p < \infty$. In case of Gaussian random noise the L_2 norm is the optimal choice, obtaining the following nonlinear least squares problem:

$$\min_{\mathbf{q}} \frac{1}{2} \|F(\mathbf{q}) - \mathbf{y}^\delta\|_2^2. \tag{1}$$

Since the present paper focuses on the L_2 norm we define $\| \cdot \| \equiv \| \cdot \|_2$. It is well known that such problems are ill-posed in the sense that noise present in the data leads to poor solutions, hence some form of regularization needs to be introduced. Among the most common ways to regularize problem (1) is the introduction of a suitable regularization constraint, taking into account the smoothness of the solution (see [8] and reference therein). However the application of such methods to the estimation of several parameters of a differential model is quite challenging and has a high computational cost. A more practical way consists in exploiting the possibly semi-convergent behavior of the iterative methods used to solve (1) and compute stable solutions by means of a suitable stopping criterion. The aim of this work is to test the iterative methods implemented by the functions `fminunc` and `lsqnolin` of the Matlab Optimization Toolbox and evaluate their efficiency in the solution of problem (1) with noisy data, verifying the *semi-convergence* in presence of medium-high noise. To this purpose we define a test problem where the state equation is a system of two time dependent differential equations, representing the dynamic evolution of the liquid and solid phases of Polyphenolic compounds [4]. The noisy measurements $\mathbf{y}^\delta$ are obtained by adding Gaussian noise to the solution of the state equation with given reference parameters $\mathbf{q}_{true}$. We observe a progressive worsening of the results of both functions while increasing the level of noise in the data. The analysis of the relative errors at each iteration shows that the error curve has the typical *semi-convergent* behaviour: it decreases in the first steps and, after reaching a minimum value, it start to increase, reaching errors possibly higher than those at the initial step. Hence a change in the convergence conditions and tolerances can improve the solutions. Although the semi-convergence of Descent, Gradient and Simultaneous Iterative Reconstruction Technique (SIRT) methods is well understood [11, 10, 9], the same does not apply to Gauss Newton, Levemberg Marquardt or quasi Newton Methods such as the BFGS (Broyden Fletcher Goldfarb Shanno). Therefore after the a posteriori verification of such behaviour, we define a stopping rule based on the Morozov's Discrepancy Principle [7], that is proved to be suitable for descent, gradient and SIRT methods. By suitably

defining the `options` structure, it is possible to modify the stopping conditions of the matlab optimization functions and evaluate the improvement obtained by computing the solution reached by the application of the stopping rule in case of data with different noise levels. Moreover we can evaluate the efficiency of our stopping rule compared to the optimal solution obtained by minimizing the relative error.

In section 2 we define the state equation and analyze the details of its numerical solution. The optimization methods implemented by the functions `fminunc` and `lsqnolin` are outlined in section 3, together with the proposed stopping rule. Finally in section 4 we report the numerical results and the conclusions.

2   The Adsorption Model

We describe here the differential problem (state equation) used as a test problem. It consists of a system of two time dependent differential equations representing the dynamic evolution of the liquid and solid phases of Polyphenolic compounds [4]. In the hypotheses of not negligible mass transfer and Langmuir adsorption isotherm the liquid phase u is modeled by a convection, diffusion and reaction equation while the solid phase concentration v is characterized by the absence of any dispersion and convection:

$$\begin{cases} \delta_v \frac{\partial u}{\partial t} = -\nu \frac{\partial u}{\partial z} + D \frac{\partial^2 u}{\partial z^2} - R(u,v,\boldsymbol{\theta}) + f_u \\ \delta_v \frac{\partial v}{\partial t} = R(u,v,\boldsymbol{\theta}) + f_v \end{cases} \quad z \in [0,L] \quad (2)$$

where the adsorption isotherm is given by

$$R(u,v,\boldsymbol{\theta}) = \theta_1 \left(u - \frac{\theta_2 v}{\theta_3 - v} \right)$$

The parameters to be identified are $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ while the dispersion coefficient D, the interstitial velocity coefficient ν and the retardation factors δ_u, δ_v are assumed to be known. The spatial domain is given by the height L of the column reactor. Dirichlet boundary conditions are assumed in $z = 0$ while convective flux is assumed in $z = L$ for the liquid phase. Using the Method of Lines, problem (2) is tranformed in the following system of Nonlinear Ordinary Differential Equations:

$$\begin{cases} U'(t) = KU(t) - G(t,\boldsymbol{\theta}) \\ V'(t) = G(t,\boldsymbol{\theta}) \end{cases}, \quad G(t,\boldsymbol{\theta}) \in \mathbb{R}^M, \quad (3)$$

where M is the number of intervals in the spatial domain, $U(t) = (u_1(t), \ldots, u_M(t))$, $V(t) = (v_1(t), \ldots, v_M(t))$ are the discrete solutions at time t and G is the discrete isotherm at time t of components $G_i = R(u_i(t), v_i(t), \boldsymbol{\theta})$, $i = 1, \ldots, M$. The matrix $K \in \mathbb{R}^{M \times M}$ is the tridiagonal matrix obtained by applying the second order finite differences approximation of the spatial derivatives in (2). It is well known that such systems tend to become very stiff at the increasing of

the spatial resolution M, hence a suitable implicit solver is required. In our experiments we use the matlab function `ode15s`, which implements variable order $(1 - 5)$ method and variable step size, being therefore the most accurate solver available in Matlab ODE-suite for stiff problems.

3 Iterative Regularization

In this section we define the stopping rule applied to the iterative methods used by `fminunc` and `lsqnolin` functions to solve the test problem obtained by the model described in section 2. We start by a brief outline of the iterative numerical methods tested in the numerical experiments.

The first method, Levemberg Marquardt, is specific of the non linear Least squares minimization while the second method, BFGS (Broyden Fletcher Goldfarb Shanno) quasi Newton method, is applied to more general nonlinear minimization problems. (See [5] [6] for details). Both methods compute a sequence of approximate solutions of (1), $\{\mathbf{q}(k)\}$, $k = 0, 1, \ldots$ by the following update relation:

$$\mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} + \alpha_k \mathbf{s}_k \tag{4}$$

where $\alpha_k \in (0, 1]$ is a damping parameter used to guarantee the decrease of the residual norm $\|F(\mathbf{q}^{(k)}) - \mathbf{y}^\delta\|_2$. In the case of Levemberg Marquardt the direction $\mathbf{s}_k$ is computed by solving the linear system obtained by the first order conditions of the linear approximation of the residual at $\mathbf{q}^{(k+1)}$:

$$\left((J_F^{(k)})^t J_F^{(k)} + \lambda_k I \right) \mathbf{s}_k = -(J_F^{(k)})^t (F(\mathbf{q}^{(k)}) - \mathbf{y}^\delta), \quad \lambda_k \geq 0 \tag{5}$$

where $J_F^{(k)}$ is the Jacobian matrix $(J_F^{(k)})_{i,j} = \partial F_i(\mathbf{q}^{(k)})/\partial \mathbf{q}_j$. If the parameter λ_k is zero we have the Gauss Newton Method, otherwise, to overcome possibly singular Jacobians, a diagonal positive matrix is added by means of a small scalar parameter λ_k (Levemberg Marquardt Method). Notice that (5) is the first order condition of the following constrained minimization problem:

$$\min_{\mathbf{s}} \|J_F^{(k)} \mathbf{s} + \mathbf{r}_k\|_2^2, \ s.t. \ \|\mathbf{s}\|_2^2 \leq \Delta_k \tag{6}$$

where $\mathbf{r}_k = \mathbf{y}^\delta - F(\mathbf{q}^{(k)})$ and Δ_k is the level of smoothness required by $\mathbf{s}$ and can be computed by the Trust Region method (see algorithm 4.1 in [6]).

In absence of data noise the following stopping rule is used to stop the iterations:

$$G_F \mathbf{s}^{(k)} < \tau_F \text{ and } \|G_F\|_\infty < 10(\tau_F + \tau_X), \quad \|s^{(k)}\|_\infty < \tau_X \tag{7}$$

where $G_F = 2(J_F^{(k)})^t (F(\mathbf{q}^{(k)}) - \mathbf{y}^\delta)$ and τ_F, τ_X are tolerance parameters.

The second family of methods is that of Quasi Newton methods where the direction $\mathbf{s}_k$ used in the update step (4) is computed by solving the following linear system:

$$H^{(k)} \mathbf{s}_k = -\nabla_{\mathbf{q}} \widehat{J}(\mathbf{q}^{(k)})$$

where $H^{(k)}$ is the approximate Hessian matrix, whose value is updated by adding a rank one update term. The BFGS method uses the following update term $H^{(k+1)} = H^{(k)} + S(\mathbf{q}^{(k)})$, where

$$S(\mathbf{q}^{(k)}) = \frac{v_k v_k^t}{v_k^t \mathbf{s}_k} - \frac{H^{(k)} \mathbf{s}_k \mathbf{s}_k^t H^{(k)}}{\mathbf{s}_k^t H^{(k)} \mathbf{s}_k}, \quad v_k = \nabla_{\mathbf{q}}(\widehat{J}(\mathbf{q}^{(k+1)})) - \nabla_{\mathbf{q}}(\widehat{J}(\mathbf{q}^{(k)}))$$

The initial Hessian approximation is chosen as $H^{(0)} = \gamma I$ where $\gamma > 0$ is relative to the scaling of the variables. The stopping criterion applied in this case is:

$$\|\nabla \widehat{J}(\mathbf{q}^{(k)})\|_\infty < \tau_F (1 + \|\nabla \widehat{J}(\mathbf{q}^{(0)})\|_\infty), \quad \max_i \left(\frac{|q_i^{(k+1)} - q_i^{(k)}|}{1 + |q_i^{(k)}|} \right) < \tau_X \quad (8)$$

The main computational effort in each iteration is the computation of the Jacobian matrix. The approach used here is that of finite difference approximation. Although it is not optimal for precision and computational cost, it is simple and readily available in matlab software optimization tools. As reported in section 4, in case of noiseless data, the sequences $\{\mathbf{q}^{(k)}\}$ of both methods converge to good approximate solutions of (1). On the other hand, with noisy data we observe bad results for both algorithms. A thorough analysis reveals that the error curve has the typical *semi-convergent* behaviour: it decreases in the first steps and then start to increase giving a completely wrong solution. An improvement can be obtained by heuristically increasing the tolerances τ_F and τ_X, so as to decrease the number of steps. This strategy would require specific tolerance values for each noise level, which is unknown.

More systematic stopping rules can be obtained by means of the Morozov's Discrepancy Principle (MDP) that proposes to stop (4) at the d-th iteration as soon as the residual norm approximates the data noise:

$$\|F(\mathbf{q}^{(d)}) - \mathbf{y}^\delta\| \simeq \delta, \quad (9)$$

The main drawback is the need to estimate the noise δ. In order to overcome this difficulty we exploit the decreasing behaviour of the residual norm. We observe that the decrease of the residual norm is fast in the first iterations and tends to become slower and slower as the iterations increase. Hence computing the decrease rate (measured as the difference of the residual norms in two successive steps: d and $d-1$) we stop as soon as it becomes sufficiently small, compared to the initial decrease rate. Therefore we propose the following stopping rule (SR_d)

$$|R_d - R_{d-1}| < \tau |R_1 - R_0|, \quad \tau > 0 \quad (10)$$

where R_d is the residual norm at d-th step: $R_d = \frac{1}{2}\|F(\mathbf{q}^{(d)}) - \mathbf{y}^\delta\|_2$ and the parameter τ represents the ratio between the change in the residual norm at step d and that at the first step. The optimal parameter should stop the iterations as soon as the noise starts to deteriorate the solution, causing an increase of the relative error. Hence the optimal value depends on the method, the data and the noise as well. Improvement in (10) could be obtained by adding information

about the maximum relative change in the solution.However in this application, using only information about the residual norm, it is possible to obtain great improvement in the results. In all our experiments we used the following values for the tolerance τ: 10^{-1} for the Levemberg Marquardt method and 10^{-5} for the BFGS method.

4   Numerical Experiments

The numerical experiments reported in this section are carried out on a test problem obtained by the model equations (2) where the terms f_u and f_v are defined by the known solutions:

$$u(t,z) = e^{(-\pi^2 t)}(\sin(\pi z^2))\cos(0.5\pi z^2), \quad v(t,z) = e(-\pi^2 t)(\sin(\pi z))$$

The reference parameter vector $\mathbf{q}_{true}$ has elements $[\theta_1, \theta_2, \theta_3] = [1, 2, 3]$. The retardation factors are $\delta_u = 1, \delta_v = 2.2$, and the spatial domain $[0.2, 0.8]$ is discretized using N_z uniform spaced samples. The differential system (3) is solved in the time interval $[0, 0.1]$ by means of the matlab function `ode15s` with tolerance parameters `AbsTol=RelTol=`10^{-10}.

The measurements $\mathbf{y} = F(\mathbf{q}_{true})$ are defined on a uniformly spaced grid of $N_t \times N_z$ points on the time space domain and are computed by solving (3) on an oversampled space domain: $\simeq 3N_z$ points. In order to compare the results ($\mathbf{q}$) computed by the different methods, we evaluate the Parameter Relative Error (PRE) and Residual Norm ($ResN$) defined as follows:

$$PRE = \frac{\|\mathbf{q} - \mathbf{q}_{true}\|}{\|\mathbf{q}\|} \quad ResN = \|F(\mathbf{q}) - \mathbf{y}\| \tag{11}$$

The reported results are computed on a PC Intel(R) equipped with 4 i5 processors 5.8 GB Ram, using Matlab R2010a.

In the first experiment we compare the results obtained by the Levemberg Marquardt and BFGS methods without data noise. The Levemberg Marquardt method with parameter λ (5) is implemented by the matlab function `lsqnonlin` with the option `'Algorithm',{'levenberg-marquardt',`λ `}`. In this experiment we choose the constant value $\lambda = \varepsilon$ (machine epsilon) throughout all the iterations and we call this method $LM(\varepsilon)$.

The BFGS method is implemented by the matlab function `fminunc` setting the option `{ 'HessUpdate', 'bfgs' }`.The starting value $\mathbf{q}^{(0)}$ is chosen at a relative distance δ_q from $\mathbf{q}_{true}$, i.e. $\mathbf{q}^{(0)} = \mathbf{q}_{true}(1 \pm \delta_q)$. The iterations are stopped using the standard convergence stopping rules (7) and (8) with $\tau_F = \tau_X = 10^{-6}$.

In table 1 we report the parameter error (PRE) and Residual norm ($NRes$) obtained by solving the problem with $N_t \times N_z$ samples ranging in the domain $[20, 40] \times [20, 40]$ and with an initial relative distance δ_q given by 25% and 40%. The computational cost is evaluated by the total number of function evaluations `fval` and by the number of iterations k. We observe that $LM(\varepsilon)$ performs much

$\delta_q(\%)$	N_t	N_z	$LM(\varepsilon)$			BFGS		
			PRE	$ResN$	$k(\texttt{fval})$	PRE	$ResN$	$k(\texttt{fval})$
	20	20	1.0455e-12	3.0782e-26	5(24)	4.8683e-03	6.2636e-09	11(56)
	20	30	2.0644e-11	1.9521e-24	5(24)	1.7381e-03	1.1973e-09	9(52)
25	30	30	1.3760e-12	4.5015e-25	5(24)	1.7167e-03	1.7694e-09	10(48)
	40	30	3.1294e-10	3.0582e-22	5(24)	4.5629e-05	1.6661e-12	20(92)
	40	40	1.3050e-12	2.1200e-25	5(24)	7.3953e-05	5.9466e-12	20(92)
	20	20	2.9505e-12	1.8642e-26	7(32)	5.3876e-04	8.7967e-11	22(108)
	20	30	1.4647e-11	5.3618e-25	7(32)	7.5150e-05	2.4042e-12	25(112)
40	30	30	4.7820e-12	3.2652e-25	7(32)	5.2751e-05	1.7968e-12	25(112)
	40	30	4.4779e-11	8.5756e-24	7(32)	3.8518e-05	1.2619e-12	25(112)
	40	40	1.3728e-11	1.0526e-24	7(32)	1.3826e-04	2.5034e-11	24(108)

Table 1. Test with noiseless measurements. Levemberg Marquardt $LM(\varepsilon)$ with $\lambda = \varepsilon$. BFGS with $H_0 = I$

better than BFGS for precision and computational complexity. Furthermore it is more robust in terms of dependence on the initial parameter estimate δ_q. We see that an initial relative error of 25% or 40% doesn't affect much the errors of $LM(\varepsilon)$ while BFGS is more precise for smaller values of δ_q. The behavior of the error curves plotted in figures 1 confirms the faster convergence of $LM(\varepsilon)$ compared to BFGS.

In the second experiment we introduce Gaussian random noise of level $\delta \in [10^{-4}, 10^{-1})$ and estimate the parameters starting from the noisy data $\mathbf{y}^\delta$ defined as follows:

$$\mathbf{y}^\delta = \mathbf{y} + \delta \|\mathbf{y}\| \boldsymbol{\eta}, \qquad \|\eta\| = 1 \tag{12}$$

We run this analysis for all the cases in table 1 but, as an example, we report in table 2 the details of the case $N_t = N_z = 40$ with an initial parameter estimate $\delta_q = 40\%$. The error parameters reported in table 2 show that both methods are

δ	$LM(\varepsilon)$			BFGS		
	PRE	$ResN$	k (fval)	PRE	$ResN$	k (fval)
1e-4	1.1224e-3	1.1849e-5	7(32)	1.2120e-3	1.7211e-03	21(104)
1e-3	1.3231e-2	1.1852e-3	7(32)	1.3275e-2	1.7229e-02	24(112)
5e-03	1.1362e-01	2.9605e-02	5(24)	1.1415e-01	8.6031e-02	24(108)
1e-2	6.2798e-2	1.1855e-01	6(41)	5.7114e-2	8.6031e-02	24(108)
5e-2	1.2936e+4	2.9593	8(36)	1.7283e+1	8.6013e-01	32(208)

Table 2. Noisy data: $N_t = N_z = 40$, $\delta_q = 40\%$

very sensitive to noise, even if BFGS seems to be more stable than $LM(\varepsilon)$ when $\delta \geq 1.e - 2$. Focusing on the BFGS method we observe the *semi-convergence* by plotting the PRE values at each iteration (see figures 2). The PRE curves obtained with noise $\delta = 0.01$ and $\delta = 0.1$ are plotted in figures 2, where the iteration at which the relative error is minimum is represented by a red star and

the iteration d obtained by rule SR_d (10) by a green circle. By changing the exit condition as in (10), with tolerance $\tau = 10^{-5}$, we obtain a great improvement for the noisy data, as reported in table 3. The relative errors obtained in column 5 are very close to the minimum value (column 2). The exit condition of the `fminunc` function is changed by setting the `OutputFcn` field in the `options` structure to a user defined `mfile.m` function that implements the stopping rule (10) .

	Best			SR_d		
δ	PRE	$ResN$	k	PRE	$ResN$	d
5e-03	5.7449e-02	8.6034e-02	11	5.7449e-02	8.6034e-02	10
1e-02	5.7087e-02	1.7215e-01	11	5.7114e-02	1.7215e-01	11
5e-02	7.0872e-02	8.6022e-01	11	7.1133e-02	8.6022e-01	11
1e-01	5.6863e-02	1.7213	5	7.7457e-02	1.7211	9

Table 3. BFGS results with best and stopping rule $DR_d : N_t = N_z = 40$, $\delta_q = 40\%$

Concerning the Levemberg Marquardt method, it is possible to improve the results in case of noisy data by using a larger initial value λ_0 and a suitable strategy to update it during the computation. The matlab function `lsqnonlin` implements the following strategy:

$$\lambda_k = \max(0.1\lambda_{k-1}, \texttt{eps}) \tag{13}$$

where `eps` is the machine epsilon ε. Hence we repeat the experiment with noisy data choosing the default value $\lambda = 0.05$ and refer to this method as $LM(.05)$. Comparing the columns 2 in tables 2 and 4, we observe an improvement of the errors when $\delta > 1.e - 3$. Also in this case we have *semi-convergence* because the update formula (13) is not optimal. Actually the value of λ_k should be updated by taking into account the constraint in problem (6) (see [3] for details about a possible implementation). By applying the stopping rule (10) with $\tau = 0.1$ we observe a more stable behaviour of the error (table 4 column 8), which is always better than the standard stopping rule (7) and very close to the minimum value (column 6). The PRE curves obtained with noise $\delta = 0.01$ and $\delta = 0.1$ are plotted in figures 3, where the iteration at which the relative error is minimum is represented by a red star and the iteration d obtained by rule SR_d (10) by a green circle.

Finally the global evaluation of the two methods in all the cases reported in table 1 shows that the Levemberg Marquardt Method is usually slightly better than BFGS in terms of mean PRE (see figure 5).

5   Conclusions

The present work reports tests of the Levemberg Marquardt and BFGS algorithms for modeling the kinetic terms, occurring in chemical processes of adsorption, in the presence of noisy data. The *semi-convergent* behavior of both

δ	$LM(.05)$				min PRE		SR_d	
	PRE	$ResN$	k	fval	PRE	k	PRE	d
1e-3	2.2742e-2	5.3695e-1	6	28	9.8952e-3	5	2.1302e-2	3
5e-3	5.2977e-2	5.4327e-1	6	28	6.8027e-3	4	2.1448e-2	3
1e-2	1.1323e-1	5.6172e-1	6	28	6.7842e-3	4	1.8100e-2	3
5e-2	3.6183e+2	1.0134	11	48	5.6674e-2	3	5.6674e-2	3

Table 4. Levemberg Marquardt results with $\lambda_0 = 0.05$ case $N_t = N_z = 40$, $\delta_q = 40\%$

methods is observed in presence of noise, confirming the need to introduce a suitable stopping criterion. A stopping rule, based on the behavior of the residual norm, is presented and the good performance is reported by the experimental tests.
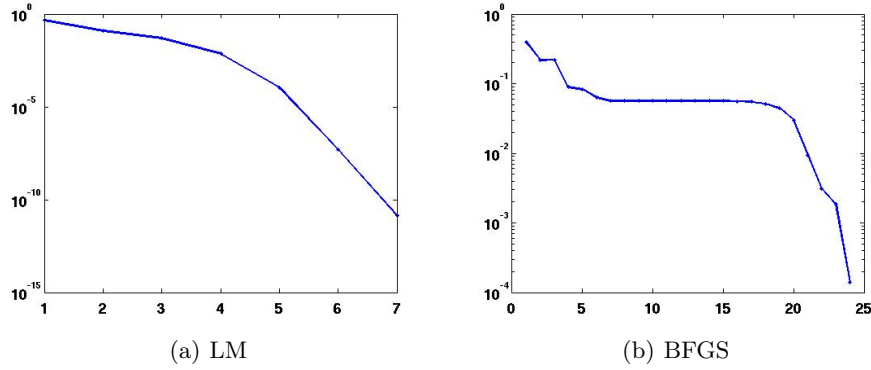

(a) LM   (b) BFGS

Fig. 1. LM and BFGS convergence plots: $\delta_q = 40\%$, $N_t = N_z = 40$

References

1. R. Ciavarelli, M. Cappelletti, S. Fedi, D. Pinelli, and D. Frascari. Chloroform aerobic cometabolism by butane-growing rhodococcus aetherovorans bcp1 in continuous-flow biofilm reactors. *Bioprocess and Biosystems Engineering*, 35(5):667–681, 2012.
2. Dario Frascari, Martina Cappelletti, Stefano Fedi, Angelo Verboschi, Roberta Ciavarelli, Massimo Nocentini, and Davide Pinelli. Application of the growth substrate pulsed feeding technique to a process of chloroform aerobic cometabolism in a continuous-flow sand-filled reactor. *Process Biochemistry*, 47(11):1656 – 1664, 2012. {ACB} 2011+ Bioenergy+Env. Biotech+Ind. Biotech.
3. F. Zama and R. Ciavarelli and D. Frascari and D. Pinelli. Numerical parameters estimation in models of pollutant transport with chemical reaction. In Dietmar Hömberg and Fredi Tröltzsch, editors, *System Modeling and Optimization*, volume
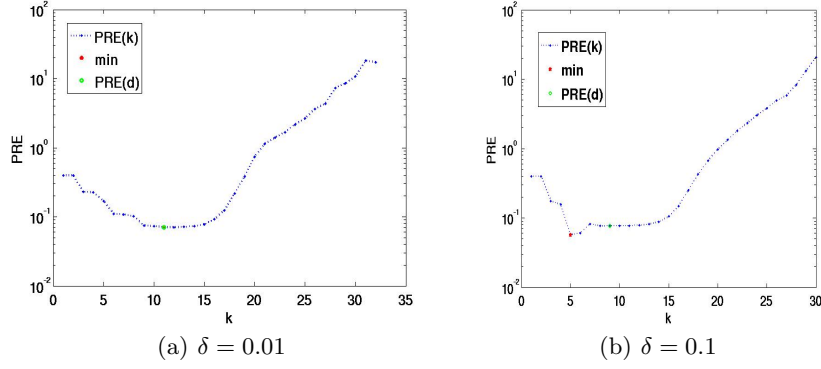
(a) $\delta = 0.01$ (b) $\delta = 0.1$

Fig. 2. BFGS convergence plots with noisy data, $\delta_q = 40\%$, $N_t = N_z = 40$

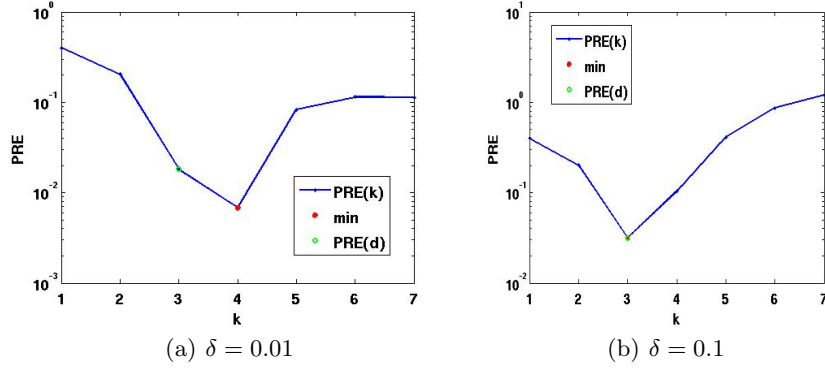(a) $\delta = 0.01$ (b) $\delta = 0.1$

Fig. 3. Levemberg Marquardt ($LM(.05)$) $\delta_q = 40\%$, $N_t = N_z = 40$

391 of *IFIP Advances in Information and Communication Technology*, pages 547–556. Springer Berlin Heidelberg, 2013.

4. D. Frascari and A. E. Molina Bacca and F. Zama and L. Bertin and F. Fava and D. Pinelli. Olive mill wastewater valorisation through phenolic compounds adsorption in a continuous flow column. *Chemical Engineering Journal*, 283:293 – 303, 2016.
5. R. Fletcher. *Practical Methods of Optimization*. Wiley, 2013.
6. J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.
7. O. Scherzer. The use of Morozov's Discrepancy Principle for Tikhonov Regularization for solving nonlinear ill-posed problems. *Computing*, 51(1):45–60, 1993.
8. F. Zama. Parameter Identification by Iterative Constrained Regularization. *Journal of Physics : Conference Series*. IOP publishing, 2016 (to appear).
9. T. Elfving and T. Nikazad and P. C. Hansen Semi-convergence and relaxation parameters for a class of SIRT algorithms. *Electronic Transactions on Numerical Analysis*. V. 37, pp. 321-336, 2010.
10. H.W.Engl, M.Hanke, A.Neubauer, Regularization of Inverse Problems. *Kluwer*. Dordrecht 1996
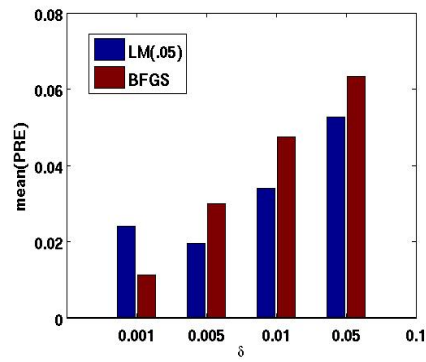
Fig. 4. Mean relative error for all the cases in table 1 with noise $\delta \in [10^{-3}, 5 \cdot 10^{-2}]$

11. M.Hanke Conjugate Gradient Type Methods for Ill-Posed Problems.   Pitman Research Notes in Mathematics, Longman House, Harlow, Essex, 1995. ISBN 0 582 27370 6