Computers & Geosciences 36 (2010) 1200–1209

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

CXTFIT/Excel–A modular adaptable code for parameter estimation, sensitivity analysis and uncertainty analysis for laboratory or field tracer experiments ☆

Guoping Tang [a,*], Melanie A. Mayes [a], Jack C. Parker [b], Philip M. Jardine [c]

[a] Environmental Sciences Division, Oak Ridge National Laboratory, P.O. Box 2008, MS-6038, Oak Ridge, TN 37831, USA
[b] Department of Civil and Environmental Engineering, University of Tennessee, Knoxville, TN 37996, USA
[c] Department of Biosystems Engineering and Soil Science, University of Tennessee, Knoxville, TN 37996, USA

ARTICLE INFO

Article history:
Received 6 September 2009
Received in revised form
23 December 2009
Accepted 4 January 2010

Keywords:
Equilibrium/nonequilibrium convection dispersion equation
Weighted nonlinear least squares
Penalty function
Monte Carlo analysis
Response surface

ABSTRACT

We implemented the widely used CXTFIT code in Excel to provide flexibility and added sensitivity and uncertainty analysis functions to improve transport parameter estimation and to facilitate model discrimination for multi-tracer experiments on structured soils. Analytical solutions for one-dimensional equilibrium and nonequilibrium convection dispersion equations were coded as VBA functions so that they could be used as ordinary math functions in Excel for forward predictions. Macros with user-friendly interfaces were developed for optimization, sensitivity analysis, uncertainty analysis, error propagation, response surface calculation, and Monte Carlo analysis. As a result, any parameter with transformations (e.g., dimensionless, log-transformed, species-dependent reactions, etc.) could be estimated with uncertainty and sensitivity quantification for multiple tracer data at multiple locations and times. Prior information and observation errors could be incorporated into the weighted nonlinear least squares method with a penalty function. Users are able to change selected parameter values and view the results via embedded graphics, resulting in a flexible tool applicable to modeling transport processes and to teaching students about parameter estimation. The code was verified by comparing to a number of benchmarks with CXTFIT 2.0. It was applied to improve parameter estimation for four typical tracer experiment data sets in the literature using multi-model evaluation and comparison. Additional examples were included to illustrate the flexibilities and advantages of CXTFIT/Excel. The VBA macros were designed for general purpose and could be used for any parameter estimation/model calibration when the forward solution is implemented in Excel. A step-by-step tutorial, example Excel files and the code are provided as supplemental material.

Published by Elsevier Ltd.

1. Introduction

CXTFIT is widely used for tracer experiment data analysis to investigate transport of solutes in the subsurface. The original FORTRAN version (Parker and van Genuchten, 1984b) reads ASCII data files with observations, initial parameter guesses, and control variables. Parameters for equilibrium or nonequilibrium convection dispersion models (CDE or MIM) are estimated using a nonlinear least squares method. Optimized values, standard errors, and correlations for parameters and predicted concentrations are output for the fitted model. Toride et al. (1995) added solutions for first order decay and zero order production for MIM, nonuniform initial conditions for CDE and MIM and developed a user-friendly interface. Nützmann et al. (2005) implemented Visual CXTFIT in Microsoft Excel with similar input/output structure and functions.

The above versions of CXTFIT have a number of limitations. First, the parameters that can be estimated are hard-wired and are not always convenient to the experiment or study objectives. For example, average pore velocity v, dispersion coefficient D, retardation factor R, and pulse volume T_0, are invertible parameters in existing CXTFIT. However, these parameters are not independent, e.g., $v=q/\theta$, $D=\lambda v$, $R=1+\rho_b k_d/\theta$ and $T_o=qt_o/\theta L$ where q is Darcy velocity, θ is effective porosity, λ is dispersivity, ρ_b is bulk density, k_d is partition coefficient, and t_o is pulse duration. All of the invertible parameters are dependent on θ and model predictions are often very sensitive to θ (e.g., Tang et al., 2009). If θ is not accurately known, or is suspected to be different from the measured total porosity, inversion results may be inaccurate. Because sensitivity is essential for the identification of transport mechanisms and the estimation of parameters, and it is useful for

☆ Code available from server at http://www.iamg.org/CGEditor/index.htm.

* Corresponding author. Tel.: +1 865 574 7314; fax: +1 865 576 8646.
E-mail address: tangg@ornl.gov (G. Tang).

0098-3004/$ - see front matter Published by Elsevier Ltd.
doi:10.1016/j.cageo.2010.01.013

experimental design and the diagnosis of misfit and overfit (Tang et al., 2009), the option to output sensitivity is necessary.

Second, CXTFIT was designed to estimate one set of parameters for one solute at a time. Take Fig. 7.9 in Toride et al. (1995) for example, D and mobile water fraction θ_m/θ were first estimated from ^3H_2O data to be $D=15.57$ cm^2 d^{-1} and $\theta_m/\theta=0.823$ with a 95% confidence interval of 7.129–24.01 cm^2 d^{-1} and 0.760–0.886, respectively. The D and θ_m/θ estimates were assumed to be identical for ^3H_2O and boron (B) and were subsequently used to estimate the mass transfer coefficient ω and fraction of equilibrium sorption sites (f) of B. While this logic is commonly used and may simplify parameter estimation for B, the impact of the uncertainty in D and θ_m/θ on the estimate of ω and f for B is ignored. Fitting data for both ^3H_2O and B simultaneously using the same assumptions (identical D and θ_m/θ for ^3H_2O and B) might be preferable because the uncertainties in all the parameters could be quantified (Tang et al., 2009). Such a strategy requires more flexibility in setting up the parameter estimation problem than what is currently possible in CXTFIT.

Third, CXTFIT uses an ordinary nonlinear least squares method based on the assumption of normally distributed error with identical variance for all observations. In practice, measurement errors may differ for different concentration levels or for different chemicals in the same experiment. Parameters in CXTFIT are assumed to be either known or unknown and hence they are either fixed or estimated (with or without constraints) in the inverse solution. A weighted least squares formulation with a penalty function can incorporate observation uncertainties and prior parameter information to improve parameter estimation. Data and/or parameter transformations can also make the optimization faster and more stable (Doherty, 2004). This flexibility is useful when many parameters are estimated and/or when uncertainty in a few parameters is significant.

Finally, the nonlinear squares method in CXTFIT is solved by a gradient method for efficiency and the estimated parameter confidence intervals are based on a linear approximation assumption. When many parameters are estimated, the results can be dependent on the initial guess and the optimization can be trapped in a local minimum, resulting in inaccurate confidence intervals and sensitivities. In these cases, it is desirable to check the response surface to examine the correlation of two or three highly correlated parameters (Hill and Tiedeman, 2007) and observe the shape and range of the confidence intervals of parameters (Press et al., 1992). In the case of many parameters, Monte Carlo analysis can also help determine if the optimized value is a global minimum and to analyze global sensitivity and uncertainty of parameters with respect to model fits.

The objective of this work is to implement a version of CXTFIT in Excel to achieve greater flexibility in setting up parameter estimation problems and to provide additional tools for sensitivity analysis, uncertainty analysis, model evaluation, and model comparison. Unlike Visual CXTFIT by Nützmann et al. (2005) and CXTFIT by Parker and van Genuchten (1984b) and Toride et al. (1995), we implement the direct solutions as math functions so that users can easily set up direct and inverse problems according to individual needs and preferences. We add functions with user-friendly interfaces for sensitivity analysis, error propagation, response surface calculation, and Monte Carlo analysis and implement the weighted least squares method with a penalty function to incorporate observation error and parameter prior information in the inversion. Therefore, CXTFIT/Excel has far greater flexibility and more uncertainty analysis functions than previous versions. This was shown to be important for improving data interpretation for increasingly complex column experiments (Tang et al., 2009). While our previous work focused on using multi-model evaluation and comparison to improve column experimental data interpretation, this work focuses on implementation and application of CXTFIT/Excel as a tool. Additional examples from the literature are analyzed in this work to illustrate the potential advantages of CXTFIT/Excel.

The code is implemented in Microsoft Excel 2003 under Microsoft Windows XP Professional and verified by a number of benchmarks in CXTFIT 2.0 (Toride et al., 1995). The source code is available together with a step-by-step tutorial to set up and use CXTFIT/Excel from scratch ("Tutorial"). Excel files for these benchmarks, other examples, and blank Microsoft Excel files for both Excel 2003 and 2007 are also provided. Issues related to the two versions of Excel are also addressed in the Tutorial.

2. Theory

The CDE model may be written as (Parker and van Genuchten, 1984b)

$$R\frac{\partial c}{\partial t}=D\frac{\partial^2 c}{\partial x^2}-v\frac{\partial c}{\partial x}-\mu c+\gamma \tag{1}$$

subject to specified initial and boundary conditions where c is the resident concentration in the pore water [ML^{-3}], μ is a first order decay coefficient or irreversible adsorption coefficient, [T^{-1}], γ is a production rate [ML^{-3}T^{-1}], x is the distance from the inlet, and t is the time. Analytical solutions are given by Parker and van Genuchten (1984b).

The dimensionless MIM is (Toride et al., 1995)

$$\beta R\frac{\partial C_m}{\partial T}=\frac{1}{P}\frac{\partial^2 C_m}{\partial X^2}-\frac{\partial C_m}{\partial X}-\omega(C_m-C_{im})-\mu_m C_m \tag{2a}$$

$$(1-\beta)R\frac{\partial C_{im}}{\partial T}=\omega(C_m-C_{im})-\mu_{im}C_{im} \tag{2b}$$

where C_m and C_{im} are the reduced aqueous concentrations in mobile and immobile zones, β and ω are the mobile water fraction and dimensionless mass transfer coefficient, μ_m and μ_{im} are the decay coefficients, P is the Peclet number, and X and T are the dimensionless coordinate and time. Definitions of these dimensionless parameters and analytical solutions are given in Toride et al. (1995).

The least squares method adjusts M parameters (p_1,p_2 , … , p_M) in the CDE or MIM solution $c(x,t,p_1,p_2,\ldots p_M)$ to match N observations (c_1,c_2, … , c_N). With the standard deviation σ_{pj} for mean prior estimate $\overline{p}_j$ and σ_{ck} for observation c_k, the objective function considered is

$$Obj=\sum_{k=1}^{N}[c_k-c(x_k,t_k,p_1,\ldots,p_M)]^2/\sigma_{ck}^2+\lambda_p{}^2\sum_{j=1}^{M}(\overline{p}_j-p_j)^2/\sigma_{pj}^2 \tag{3}$$

with λ_p^2 as a relative penalty weight. Depending on observation error distribution or assumed distribution of parameters, measurement c_k and its corresponding prediction or parameter p_j in Eq. (3) may be a transformed variable (e.g., log-transformed). Minimization of the objective function is achieved using Microsoft Excel Solver. When $\sigma_{ck}=1$ and the second term is ignored, the weighted least squares method with penalty function reduces to the ordinary least squares method.

The Jacobian matrix, $J_{kj}=\partial c(x_k,t_k,p_1,p_2,\ldots p_M)/\partial p_j$, is approximated by finite difference as

$$J_{kj}=[c(x_k,t_k,p_1,p_2,\ldots,(1+\Delta)p_j,\ldots p_M)-c(x_k,t_k,p_1,p_2,\ldots,p_j,\ldots p_M)]/(\Delta p_j), \tag{4}$$

with a perturbation Δ of 1% (Toride et al. 1995). The scaled sensitivity, $dss_{kj}=J_{kj}p_j/\sigma_{ck}=\partial c(x_k,t_k,p_1,p_2,\ldots p_M)/\partial \ln(p_j)/\sigma_{ck}$, is often

used for the comparison of sensitivity for parameters of different units and the composite scaled sensitivity is

$$css_j = \left(\sum_{k=1}^{N} ss_{kj}{}^2 / N \right)^{1/2}$$ (Hill and Tiedeman, 2007).

The covariance matrix is approximated by

$$\mathbf{Cov} = \left[\mathbf{J}^T \mathbf{C}_c \mathbf{J} + \lambda_p{}^2 \mathbf{C}_p \right]^{-1} \quad (5)$$

with the diagonal observation covariance matrix $C_c(k,k) = 1/\sigma_{ck}^2$, and diagonal parameter covariance matrix $C_p(j,j) = 1/\sigma_{pj}^2$. The standard error of parameter p_j is approximated by $s_{pj} = Cov(j,j)^{1/2}$, the coefficient of variation (CV) is s_{pj}/p_j, and the correlation between p_j and p_l is approximated by $r(p_j,p_l) = Cov(j,l)/s_{pj}s_{pl}$.

For the ordinary least squares method where observation variance is unknown, the standard deviation for all observations is approximated by *RMSE*, or $s_c = [Obj/(N-M)]^{1/2}$ and the penalty term is zero.

To propagate the uncertainty in parameters, for example, p_j, p_l, and p_m, to a prediction $c(x,t,p_1,p_2,\ldots p_M)$ at location x and time t, the variance is approximated by

$$s^2_{propagation} = \begin{bmatrix} \partial c/\partial p_j & \partial c/\partial p_l & \partial c/\partial p_m \end{bmatrix} \begin{bmatrix} Cov_{jj} & Cov_{jl} & Cov_{jm} \\ Cov_{lj} & Cov_{ll} & Cov_{lm} \\ Cov_{mj} & Cov_{ml} & Cov_{mm} \end{bmatrix} \begin{bmatrix} \partial c/\partial p_j \\ \partial c/\partial p_l \\ \partial c/\partial p_m \end{bmatrix} \quad (6)$$

For the ordinary least squares method, total prediction uncertainty for a prediction $c(x,t,p_1,p_2,\ldots p_M)$ can be approximated by

$$s^2_{prediction} = s_c^2 + s^2_{propagation} \quad (7a)$$

or

$$s^2_{prediction} = s_c^2 \left[1 + 1/N + (t-\bar{t})^2 / \sum_{k=1}^{N} (t_k - \bar{t})^2 + (x-\bar{x})^2 / \sum_{k=1}^{N} (x_k - \bar{x})^2 \right] + s^2_{propagation}(t_h) \quad (7b)$$

with $\bar{t}$ and $\bar{x}$ as the mean time and distance, respectively, for the calibration data set (Neter et al., 1996). Eq. (7b) accounts for the effect of the closeness of the prediction from the observations: the farther away the prediction location and time from that of the observations, the greater the uncertainty prediction.

Confidence intervals for estimated parameters and model predictions are approximated with Student's t distribution for two-sided confidence α (e.g., 95%) with $N-M$ degrees of freedom. For example, the confidence intervals for p_j is $p_j \pm t(1-\alpha/2, N-M)s_{pj}$.

3. Implementation

3.1. Model setup

As described by Wraith and Or (1998) and illustrated in Fig. 1, setting up a parameter estimation problem in a spreadsheet involves input of parameters, observations, and formulae. The parameters may either be fixed (Step 1) or estimated with or without constraints (Step 2). The observations are at various times and/or locations (Step 3). Formulae are required for direct model predictions, residuals (Step 4), objective function (Step 5), and/or parameter confidence intervals (Step 6). Of note are the

Fig7.9a: Tritium effluent from Glendale clay loam (exp3-1, van Genuchten, 1974)
Fit ^3H_2O data using nonequilibrium convection dispersion equation.

Objective function cell

Step 1 Parameters to be fixed			Step 5			Step 6		
Column length (cm)	L	30	SSR	0.0074	= SUMSQ (E18:E53)	N	36	= COUNT (D18:D53)
Average velocity (cm/d)	v	37.5	R^2	0.977	= 1-H5/VAR (C18:C53)/(M5-1)	M	3	= COUNT (E11:E13)
Pulse duration (-)	T_0	3.102	RMSE	0.015	= SQRT (H5/(M4-M5))	Confid.	95%	
						t value	2.035	= TINV ((1-M6),M4-M5)

Parameter range

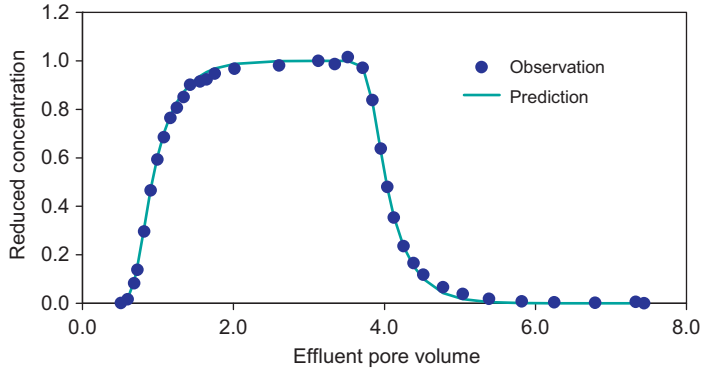
		Initial gue & estimate	Constraint Min	Constraint Max	Estimate Std. dev.	Correlation matrix λ	Correlation matrix β	Correlation matrix ω	Confidence interval Lower	Confidence interval Upper	
Dispersivity (cm)	λ	0.414	0.001	1000	0.094	1.000	0.942	-0.905	0.224	0.605	= E11+$M$7*H11
Mobile water fraction (-)	β	0.822	0.001	0.999	0.028	0.942	1.000	-0.981	0.766	0.878	
Mass transfer coefficient (-)	ω	0.873	0.001	1000	0.243	-0.905	-0.981	1.000	0.379	1.366	

=MIM(1, B18, $E$7, $E$5/$E$11, $E$12, $E$13)

Prediction range

Step 3 Observations Pore volume	Observation	Step 4 Pred.	Residual
0.5120	0.0010	0.0016	0.0006
0.5990	0.0160	0.0199	0.0039
0.6860	0.0820	0.0918	0.0098
0.7300	0.1380	0.1548	0.0168
0.8170	0.2960	0.3125	0.0165
0.9040	0.4650	0.4721	0.0071
0.9920	0.5930	0.6045	0.0115
1.0790	0.6850	0.7028	0.0178
1.1660	0.7640	0.7757	0.0117
1.2530	0.8060	0.8304	0.0244
1.3400	0.8500	0.8719	0.0219
1.4280	0.9010	0.9039	0.0029
1.5580	0.9150	0.9375	0.0225
1.6460	0.9230	0.9535	0.0305
1.7540	0.9470	0.9678	0.0208
2.0160	0.9670	0.9870	0.0200
2.6040	0.9810	0.9984	0.0174
3.1250	1.0000	0.9998	-0.0002

=D35-C35



Formula input | Data input | Formula input | Guess/estimate | Output

Fig. 1. Assembling a simple parameter estimation problem using spreadsheet (light green for data input, light blue for formula input, bright yellow for initial estimates to be overwritten after optimization, and light yellow for optimization output. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

objective function cell (e.g., H5), the parameter range (e.g., E11:E13) that contains values of parameters to be estimated and the prediction range (e.g., D18:D53) that contains the predictions corresponding to observations. They can be placed anywhere in the same sheet as long as the ranges (parameter or prediction) are continuous in a column. Initial parameter values are put in the parameter range and replaced by optimized values by Solver. Further details about setting up and solving a parameter estimation problem are available in the Tutorial.

Most of the functions used in these formulae are available in Excel except those for model predictions. Even though it is possible to input some simple analytical solutions for the transport equation as formulae (e.g., Wraith and Or, 1998), the general solutions to Eqs. (1) and (2) are too complicated. Therefore, we code the analytical solutions to Eqs. (1) and (2) as VBA functions CDE and MIM to simplify formula input for model predictions.

Fig. 2 shows the setup for a parameter estimation problem utilizing prior information and observation errors. Prior parameter estimates are in cells E8:E15 and standard deviations for the estimates are in cells F8:F15 with a "column offset" (in Microsoft Excel VBA jargon) of −3 and −2 from the parameter range (H8:H15). They are given to the left of the parameter range because the calculated standard error and correlation matrix are expected to be output to the right, by default, after lower and upper bounds (F11:G13). Similarly, the standard deviations for observations are input in D18:D53, which is on the left of the prediction range and the Jacobian matrix and error propagation are expected to output to the right with specified offset. Note that the objective function (I4) is modified to include the penalty

				SSR for predictions	35.8744	= SUMSQ (E18:E83)
Column length (cm)	L	30		SSR for parameters	0.1133	= SUMSQ (F8:F15)
Velocity (cm/d, ^3H_2O)	v	37.5	^3H_2O	Penalty weight	10.0000	← Penalty cell
Velocity (cm/d, B)	v	38.5	B	Objective function	37.0078	= I1+I2*I3

		Prior estimate	Std dev	Weighted Residual	Estimate	Min	Max	Estimated Std. dev.	Correlation matrix T_0 (^3H_2O)	T_0 (B)	R (B)	λ
Pulse duration (-)	T_0 (^3H_2O)	3.102	0.03	-0.0874	3.105	2	4	0.007	1.000	0.001	0.000	0.029
Pulse duration (-)	T_0 (B)	6.494	0.06	0.2642	6.477	6	7	0.020	0.001	1.000	0.037	0.029
Retardation factor (-)	R (B)	3.900	0.20	-0.1878	3.937	3	5	0.057	0.000	0.037	1.000	0.001
Dispersivity (cm)	λ	0.774	10.00	0.0117	0.657	0.0001	1000.0	0.160	0.029	0.029	0.001	1.000
Mobile water fract. (-)	θ_m/θ	0.879	1.00	0.0022	0.877	0.0001	0.999	0.040	0.011	0.026	0.000	0.906
Mass trans. coef. (-)	ω (^3H_2O)	0.586	10.00	0.0116	0.471	0.0001	1000.0	0.294	-0.122	-0.025	0.000	-0.851
Mass trans. coef. (-)	ω (B)	0.558	10.00	-0.0048	0.607	0.0001	1000.0	0.091	-0.015	-0.054	-0.108	-0.529
Site fraction (-)	f	0.520	1.00	0.0188	0.501	0.0001	0.999	0.022	0.016	-0.093	-0.301	0.243

= (E15-H15)/F15

Pore volume	Reduced Conc.	Observ std dev	Prediction	Weighted residual
0.5120	0.0010	0.03	0.0047	0.1250
0.5990	0.0160	0.03	0.0319	0.5292
0.6860	0.0820	0.03	0.1076	0.8519
0.7300	0.1380	0.03	0.1667	0.9571
0.8170	0.2960	0.03	0.3113	0.5089
0.9040	0.4650	0.03	0.4638	-0.0394
0.9920	0.5930	0.03	0.5990	0.2007
1.0790	0.6850	0.03	0.7033	0.6114
1.1660	0.7640	0.03	0.7801	0.5360
1.2530	0.8060	0.03	0.8354	0.9797
1.3400	0.8500	0.03	0.8755	0.8485
1.4280	0.9010	0.03	0.9053	0.1417
1.5580	0.9150	0.03	0.9361	0.7044
1.6460	0.9230	0.03	0.9510	0.9318
1.7540	0.9470	0.03	0.9645	0.5835
2.0160	0.9670	0.03	0.9839	0.5617
2.6040	0.9810	0.03	0.9973	0.5439
3.1250	1.0000	0.03	0.9995	-0.0177

= (E18-C18)/D18

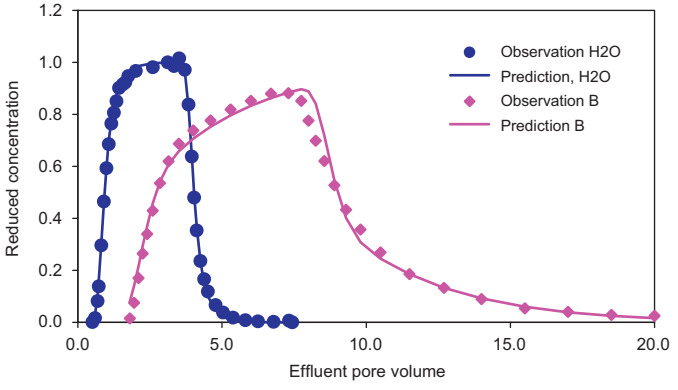


Fig. 2. Hypothetical example for application of weighted least squares method with a penalty function to incorporate prior information in parameters.

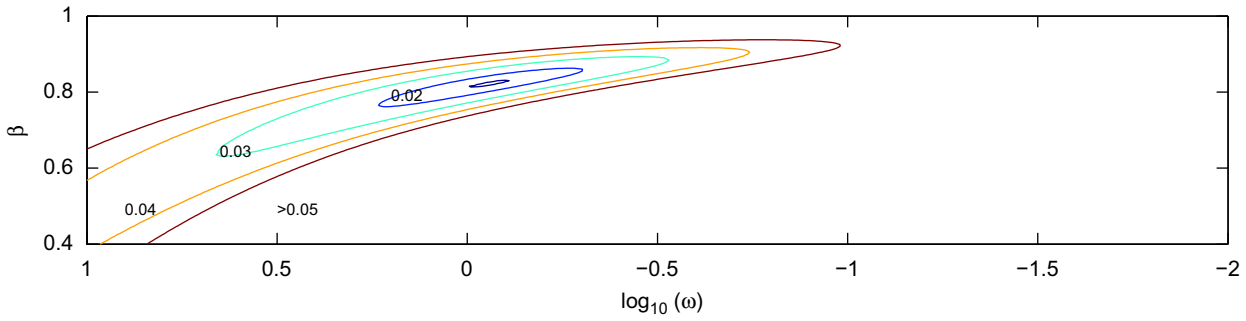


Fig. 3. Response surface contours for mobile water fraction (β) and mass transfer coefficient (ω) for application shown in Fig. 1.

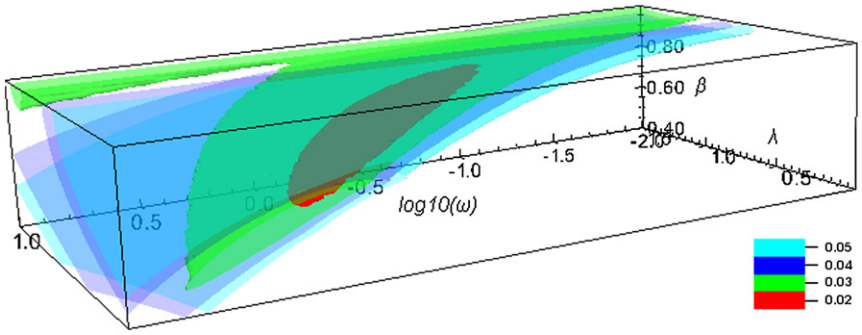


Fig. 4. Response surface contour for dispersivity (λ), mobile water fraction (β) and mass transfer coefficient (ω) for application shown in Fig. 1.

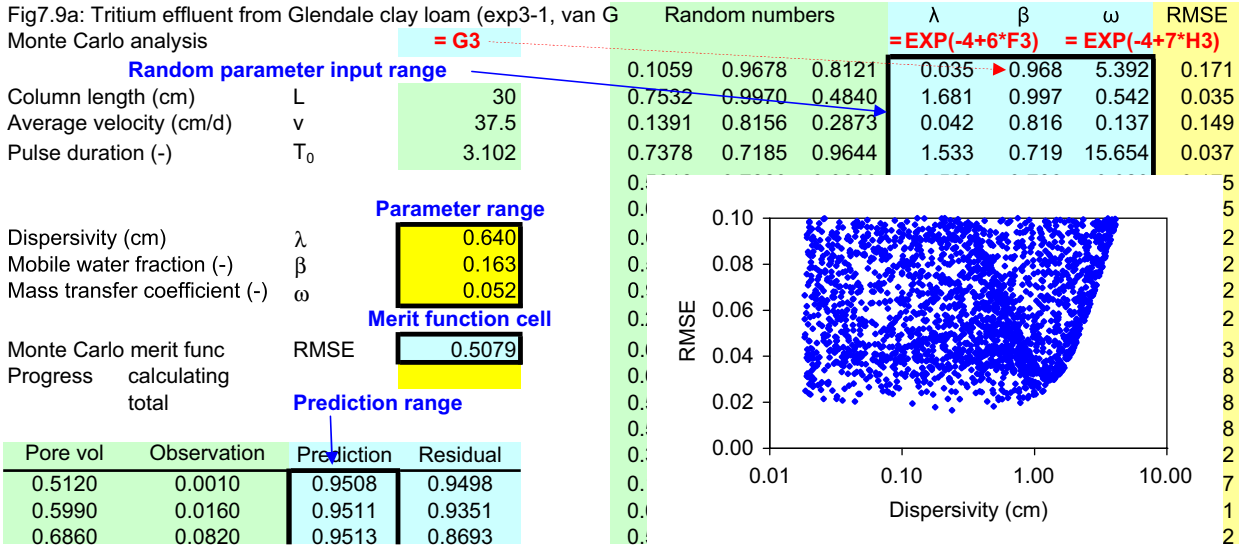


Fig. 5. Monte Carlo analysis. Light green for data input, light blue for formula input, bright yellow for parameter inputs for forward prediction. Random numbers are created using=RAND( ) in column F and pasted into columns G, H, and I as values. Random values in column J, K, and L are produced according to desired distribution for parameters. In this example, ln(λ) is assumed to be uniformly distributed between -4 and 2. Red dot in chart is optimized value from Fig. 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

function (I2) corresponding to the sum of squared deviations from prior parameter estimates (G8:G15) as well as the prediction residuals (I1).

With the forward solutions and the merit function (which, as a measure for goodness-of-fit, can be the same as or different from the objective function) set up similarly to the inverse problem and a number of values for two selected parameters (e.g., β and ω in Fig. 1) in a continuous range in a row (e.g., F2:AT2) and a column (e.g., E3:E42) in an Excel sheet, the merit function can be calculated and recorded for each pair of parameters. Then the contour of merit function can be plotted to show the response surface and check the confidence intervals of parameters (Fig. 3; Press et al., 1992; Hill and Tiedeman, 2007). Multiple sheets can be used to vary one additional parameter (say, λ in Fig. 1), and the results can be exported to a graphical software to plot 3-dimensional contours (Fig. 4). These plots vividly show if the optimized value is a global minimum, sensitivity of parameters with respect to predictions and uncertainty in the parameters. This example is given in the Tutorial.

When more parameters are estimated, Monte Carlo analysis may be used to check if the solution obtained by Microsoft Excel Solver is a global minimum and to show the global uncertainty and sensitivity of the merit function with respect to the parameters (Tang et al., 2009). Fig. 5 shows a setup for Monte Carlo analysis for the example shown in Fig. 1 with the same parameter values and model prediction assembly. RMSE is used as the merit function. The RMSE values are plotted against a large number of random parameter values to show the global uncertainty and sensitivity of the merit function for the parameter (Sambridge and Mosegaard, 2002). In our example, one thousand sets of random values for the parameters are created using the Microsoft Excel random function RAND( ) in columns G, H, I. Assuming ln(λ), β, and ln(ω) are uniformly distributed between -4 and 6, 0 and 1, and -4 and 7, these random values are further transformed in columns J, K, and L for the three parameters (for example, using formula=EXP($-4+6*$G3) in cell J3 of Fig. 5). Note that users can use other programs to produce their own samples and paste them in the random parameter input range (Fig. 5). The flat response of RMSE to dispersivity in Fig. 5 shows that dispersivity is relatively insensitive between values of 0.02 and 0.9 cm. An example Monte Carlo problem with step-by-step setup details is given in the Tutorial. More information on Monte Carlo analysis methodology with advanced sampling methods is available in, e.g., Sambridge and Mosegaard (2002).

3.2. Functions CDE and MIM

The analytical solution for Eq. (1) is coded in the function CDE(x, t ,t_0, v, D, [R, μ, γ, *strFluxConc*, c_i, c_o]) where arguments in [ ]

are optional, t_0 is the duration when influent with tracer of concentration c_0 is applied, and c_i is the initial uniform tracer concentration in the column. The calculated value is flux concentration if *strFluxConc* is "flux" or not specified. Otherwise, it yields resident concentrations (Parker and van Genuchten, 1984a). If the optional arguments are not specified, the default values ($R=1$, $\mu=0$, $\gamma=0$, *strFluxConc*="flux", $c_i=0$ and $c_0=1$) will be used. If the value for an optional argument at a later position, e.g. c_i, is different from its default value, then all of the optional arguments before c_i argument should be explicitly assigned (e.g., R, μ, γ, and *strFluxConc*) (see Fig. 8 for example).

This function can be used for both dimensional and dimensionless time, location, and concentration so long as the parameters are of consistent and appropriate units. For the example in Fig. 1, if Eq. (1) is used in the dimensionless form as

$$R\frac{\partial C}{\partial T}=\frac{1}{P}\frac{\partial^2 C}{\partial X^2}-\frac{\partial C}{\partial X}-\mu^E C+\gamma^E, \tag{8}$$

where μ^E and γ^E are the normalized first order decay coefficient and production rate as defined by Toride et al. (1995), the formula input in cell D18 (Fig. 1) is

=CDE(1.0,B18,E$7,1.0,E$11/E$5).

In this formula, $X=1.0$ (dimensionless distance from the inlet boundary); cell B18 is T; cell E$7 is T_0; 1.0 corresponds to the coefficient of 1 in Eq. (8) for the advection term; E$11/E$5 is $1/P=\lambda/L$. The rest of the optional arguments are not explicitly assigned because the default setting is sufficient for this example. If Eq. (1) is followed, the input formula is

=CDE(E$5,B18*E$5/E$6,E$7*E$5/E$6,E$6,E$11*E$6).

These formulae can be simplified and made more meaningful by defining Excel names (Jelen and Syrstad, 2004 or Tutorial) for the cells and/or values. For example, if we define cells E5, E7 and E11 as local Excel names *L*, T_0, and *Lambda*, 1.0 as *X*, the formula for dimensional case is

=CDE(*X*,B18,T_0,1.0,*LAMBDA*/*L*).

If we put=*LAMBDA*/*L* in another cell and define it with a name *INVP*, then the formula becomes

=CDE(X,B18,T0,1.0,INVP).

The analytical solution for Eq. (2) is programmed in function MIM(*X*, *T*, T_0, *P*, β, ω, [*R*, μ_m, μ_{im},*strFluxConc*,*strMobile*, C_i, C_o,*bln*, *strNI*, *tol*,*nP*]) where *strMobile* is "mobile" to solve for C_m and other value for C_{im}, *bln* is a Boolean ('true' or 'false') for ln-transformation during numerical integration (Parker and van Genuchten, 1984b; Toride et al., 1995), *strNI* is a variable to identify the numerical integration method, *tol* is the tolerance for adaptive integration and *nP* is the number of quadrature points for the Chebyshev quadrature. T_0, C_i, and C_o are dimensionless form of t_0, c_i, and c_o. If the optional arguments in [...] are not specified, the default values ($R=1$, $\mu_m=0$, $\mu_{im}=0$, *strFluxConc*="flux", *strMobile*="mobile", *tol*=0.00001, *strNI*="Chebyshev", *bln*=FALSE, and nP=75) will be used. Note that the input for string arguments, such as "flux", are case insensitive. The default numerical integration method is the Chebyshev method with 75 integration points, which is the same as CXTFIT 2.0 (Toride et al., 1995). This is expected to be sufficient for most problems and users do not have to specify these optional arguments for numerical integration. However, when computation accuracy and/or efficiency are needed, three adaptive integration methods: Romberg (Parker and van Genuchten, 1984b), Simpson, and Lobbato quadrature (Gander and Gautschi, 2000) are also available. The argument *strNI* can be set as any of the"Chebyshev", "Romberg", "Simpson", and "Lobatto". A detailed example on how to use these options is given in the Tutorial.

If MIM is used for the example in Fig. 1, the formula input in cell D18 is

=MIM(1,B18,E$7,E$5/E$11,E$12,E$13),

or simply

=MIM(1,B18,T0,L/lambda,beta,omega)

if cells E5, E7, E11, E12, and E13 are defined by names *L*, T_0, *lambda*, *beta*, and *omega*. The arguments are dimensionless distance from the inlet, effluent pore volume (dimensionless time, B18), dimensionless pulse duration (E$7 or T_0), Peclet number (E$5/E$11 or *L*/*lambda*), mobile water fraction (E$12 or *beta*), and mass transfer coefficient (E$13 or *omega*).

3.3. Macros and interfaces (dialogs and menu)

Once the parameter estimation problem is set up, Solver can be used to adjust the values in the parameter range to minimize the value in the objective function cell. Then, the Jacobian and covariance matrices may be computed and error propagation performed according to Eqs. (4–6). We develop the macros: Optimize, GetJacobianMatrix, Analyze, Propagate, CalculateResponse, and MonteCarloAnalyze to automate optimization, Jacobian matrix calculation (Eq. (4)), covariance matrix calculation (Eq. (5), output as a standard error vector and correlation matrix as in CXTFIT), error propagation (Eq. (6)), response surface calculation and Monte Carlo analysis, respectively. The macro Solve combines Optimize and Analyze to perform parameter estimation like CXTFIT. The Jacobian matrix is calculated whenever Solve, Analyze, or Propagate are executed. The macro GetJacobianMatrix is provided for the case of sensitivity analysis when none of these three macros is used.

Dialogs are designed for user's convenience to specify macro input and output locations. A menu CXTFIT is added to the Excel menu for Excel 2003 or Add-Ins tab for Excel 2007 so that users can execute these macros in a manner similar to the Microsoft Solver function. For example, the Solve Dialog (Fig. 6) can be invoked by selecting the menu CXTFIT→Solve… to specify the objective function cell, parameter range, and prediction range. Then, clicking the Solve button will invoke the macro Solve to optimize the objective function and calculate and output the standard deviations and correlation matrix next to the parameter constraints (Fig. 1).

By default, the two columns next to the parameter range are used as lower and upper bounds (i.e., constraints) for the parameter estimation. More advanced options, such as input of prior information for parameters (e.g., Fig. 2), perturbations for the Jacobian matrix calculation (Δ in Eq. (4)), standard deviations of observations (Fig. 2) and the Jacobian matrix and error propagation output can be specified by clicking the option button in the

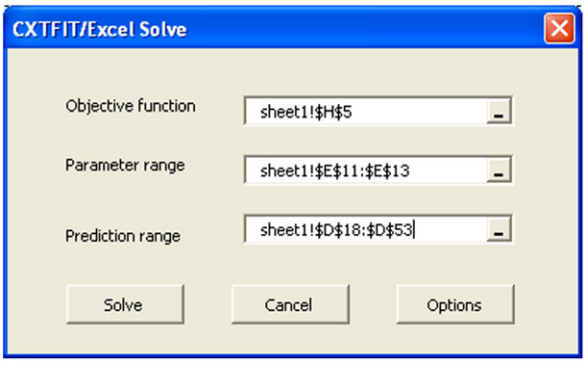


Fig. 6. Interface for macro Solve. Selections (settings) are for example in Fig. 1.

Solve Dialog (Fig. 6) to open the Solve Option Dialog (Fig. 7) to specify offsets for parameter and observation standard deviations.

CXTFIT/Excel defines a number of Excel local (sheet level) range and constant names (Jelen and Syrstad, 2004) for the inputs and options that are specified through these interfaces (dialogs). For example, Excel range names ObjFuncCell, ParameterRange, and PredictionRange are defined to refer to the objective function cell, the parameter range, and the prediction range after the Solve button in Fig. 6 is clicked. In addition, Excel constant names OffsetParaStd and OffsetPredStd are defined as -2 and -1 for the offset for the parameter and observation standard deviation from the parameter and prediction range after clicking the OK button in Fig. 7. After these names are defined, they will show up in the dialogs when opened (similar to Solver dialog). The user can redefine these names and redo the analysis through the interfaces, or use standard Excel procedures (define Excel names and run the macros) without using the interfaces. Predefined names for CXTFIT/Excel are summarized in Table 1 of the Tutorial.

4. Applications

CXTFIT/Excel was used to solve a number of benchmark problems in CXTFIT 2.0 (Toride et al., 1995) to verify the code and to provide examples to facilitate transition for CXTFIT to CXTFIT/Excel (see Tutorial). The results are very close for both forward and inverse solutions. CXTFIT/Excel was also applied to analyze four column experiment data sets for multiple model choices and parameterizations in Tang et al. (2009) to illustrate how multi-model evaluation and comparison can help in data analysis. We will briefly summarize these earlier applications before considering additional examples from the literature to further illustrate the increased flexibility and improved uncertainty analysis functions for analyzing column experimental data.

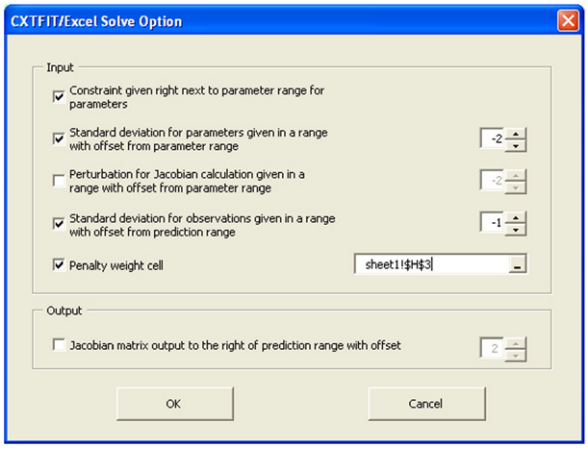


Fig. 7. Interface for options for macro Solve. Settings correspond to example in Fig. 2.

4.1. Applications in Tang et al. (2009)—Example 7.9 in Toride et al. (1995)

Fig. 1 demonstrates the application of CXTFIT/Excel to fit ^3H_2O data with the nonequilibrium convection dispersion model. Breakthrough curves for ^3H_2O and B data are simultaneously optimized assuming dispersivity and mobile water fraction are identical for the two species (Fig. 2). If measurement error for ^3H_2O and B and prior estimate uncertainty for the pulse duration and retardation factor can be reasonably estimated, this information can be employed to condition the inversion (Fig. 2). Unlike analyses with prior CXTFIT programs, dispersivity, rather than dispersion coefficient is estimated in Fig. 1 and the site fraction for B is also estimated in Fig. 2. Virtually any parameterization may be employed with a user-specified transformation. For example, $ln(\omega)$ value can be put in cell E13 in the parameter range in Fig. 1 to be estimated as long as EXP(E13) is used as ω in the formula in the prediction range. Similarly, other Excel functions and user-defined functions can be used for the transformation of parameters according to the assumed distribution of parameter values.

To further check the correlation between β and ω as shown in Fig. 1, the response surface is calculated using the macro CalculateResponse and plotted in Fig. 3 with λ fixed at optimized value. The minimum RMSE is 0.015 for MIM and 0.031 for CDE, with MIM predictions matching the observations slightly better at the end portion of both rising and failing limbs of the breakthrough curve (see Fig. 9 in Tang et al., 2009). The confidence region is quite wide even when λ is fixed at the optimized value.

Repeating the calculations for a number of λ values in multiple sheets (see Tutorial for details), the 3-D contour with respect to the three parameters is shown in Fig. 4. Around the optimized values in the center the RMSE 0.02 iso-surface forms a deformed ellipsoid that is similar to the oval region in Fig. 3 but much wider. For RMSE 0.03 the 3-D confidence region is different from the 2-D region:the 2-D response surface (Fig. 3) shows that ω can be estimated to be between $10^{-0.5}$ and $10^{0.7}$ if λ is fixed at the optimized value, while the 3-D response surface suggests that ω may not be identifiable unless λ or β is independently determined. The fitting error (RMSE) can be less than or equal to 0.03 for any ω values when β is close to 1 and λ is between 1.3 and 1.5 (Fig. 4). An equilibrium model may describe the observations sufficiently well considering uncertainties in the observations.

Similarly, Monte Carlo analysis results in Fig. 5 suggest that the minimum obtained in Fig. 1 is likely to be the global minimum and that the response surface is flat, so the objective function and first order uncertainty estimates are not very sensitive to dispersivity in the range from 0.1 to 1 m. Overall, the nonequilibrium model does match the ^3H_2O observations a little better, but the uncertainty ω is such that the nonequilibrium process may reduce into an equilibrium process, particularly if the average observation error is greater than 0.03.

Figs. 1–2 demonstrate the flexibility in estimating different parameters and incorporation of observation errors and prior

Table 1
Comparison of dispersivity estimate $\pm$ standard error for separate and simultaneous fits for application 4.3.

Dispersivity (cm)	Saturated case		Unsaturated case	
	Separate fit	Simultaneous fit	Separate fit	Simultaneous fit
@ 11 cm	0.0628 ± 0.0021	0.0668 ± 0.0080	0.1383 ± 0.0066	0.1429 ± 0.0144
@ 17 cm	0.0503 ± 0.0021	0.0505 ± 0.0059	0.1545 ± 0.0090	0.1552 ± 0.0175
@ 23 cm	0.0440 ± 0.0013	0.0441 ± 0.0050	0.1716 ± 0.0123	0.1696 ± 0.0188

Fig.7-3a: Steady saturated flow in a sand column (Toride et al. 1995)

SSR 0.07117

		Estimate	Min.	Max.	Std. Error	Correlation matrix v	λ@11	λ@17	λ@23
Average velocity	v	2.500	0.01	100	0.002	1.000	0.018	-0.005	-0.032
Dispersivity @ 11 cm	λ	0.067	0.01	100	0.004	0.018	1.000	0.000	-0.001
Dispersivity @ 17 cm	λ	0.050	0.01	100	0.003	-0.005	0.000	1.000	0.000
Dispersivity @ 23 cm	λ	0.044	0.01	100	0.002	-0.032	-0.001	0.000	1.000

x	t	Obs.	Pred.	Residual
11.0	2.52	0.0000	0.0000	0.0000
11.0	2.68	0.0002	0.0000	0.0002
11.0	2.93	0.0005	0.0001	0.0004
11.0	3.18	0.0013	0.0015	-0.0002
11.0	3.35	0.0027	0.0064	-0.0037
11.0	3.43	0.0040	0.0116	-0.0076
11.0	3.60	0.0141	0.0338	-0.0197
11.0	3.68	0.0258	0.0519	-0.0261
11.0	3.77	0.0375	0.0798	-0.0423
11.0	3.85	0.0641	0.1123	-0.0482
11.0	3.93	0.0956	0.1522	-0.0566
11.0	4.02	0.1418	0.2059	-0.0641
11 0	4 10	0 1920	0 2606	-0 0686

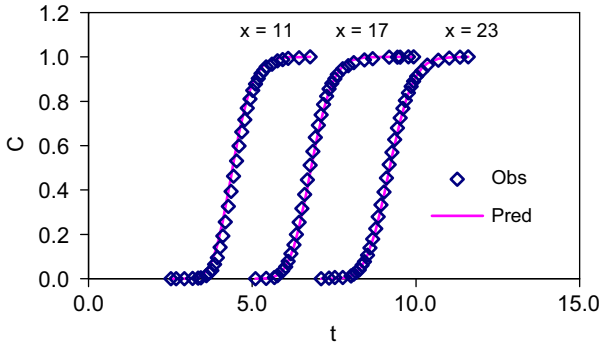


Fig. 8. Simultaneous estimation of dispersivity at different depths with fixed velocity.

Sensitivity analysis

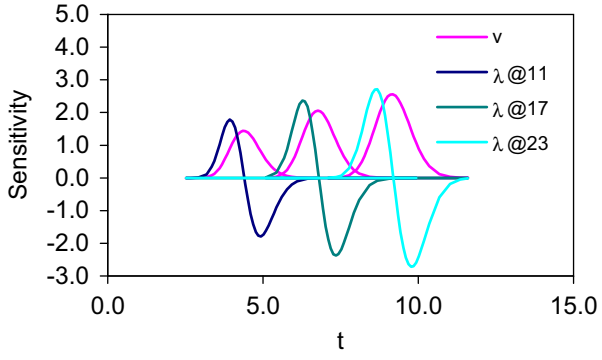
v	2.500
λ@11 cm	0.067
λ@17 cm	0.050
λ@23 cm	0.044

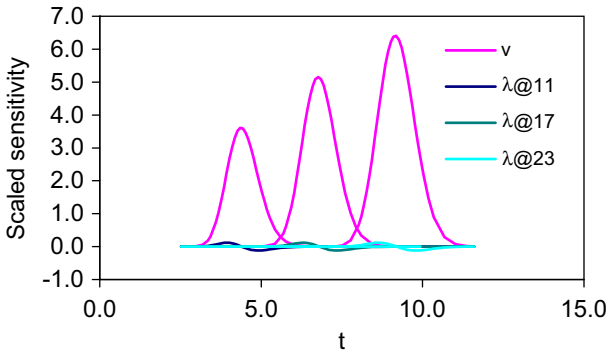
= SQRT (SUMSQ (I9:I115)/COUNT (I9:I115))

Composite scale sensitivity 2.952 0.043 0.044 0.045

= C$3*E9

				Jacobian matrix (sensitivity)				Scale sensitivity			
x	t	Obs.	Pred.	v	λ@11	λ@17	λ@23	v	λ@11	λ@17	λ@23
11.0	2.52	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
11.0	2.68	0.0002	0.0000	0.0001	0.0004	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000





11.0	4.35	0.3930	0.4591	1.4417	0.2970	0.0000	0.0000	3.6049	0.0198	0.0000	0.0000

Fig. 9. Sensitivity analysis for example in Fig. 8.

information in CXTFIT/Excel, while the functions of response surface calculation and Monte Carlo analysis are illustrated in Figs. 3 and 4. Furthermore, Tang et al. (2009) showed that CXTFIT/Excel can facilitate multi-model (particularly, the physical nonequilibrium model and the chemical nonequilibrium model) comparison, and significantly improved the reliability and rigorousness of the data analysis for four column experiments in the literature. Excel files for the four examples are also provided with the code.

4.2. *Example Fig. 7.3 in Toride et al. (1995)*

In this example, transport parameters were estimated separately at different depths, yielding v estimates of 2.45, 2.51, and 2.51 cm d^{-1} for saturated flow, and 0.258, 0.254, and 0.249 cm d^{-1} for unsaturated flow at depths of 11, 17, and 23 cm, respectively (Toride et al., 1995). This is not consistent with the assumption of steady state flow rate and uniform water content.

Using CXTFIT/Excel, a single value of v is estimated while dispersivity is allowed to vary with distance (Fig. 8). While dispersivity estimates differ only slightly for the two fitting procedures, parameter uncertainty is significantly greater for the simultaneous fit than that for the separate fit (Table 1), suggesting that the artifice of fitting to data subsets may underestimate the actual parameter uncertainty.

CXTFIT/Excel results indicate that the sensitivity for both v and λ increase with increasing depth in the saturated case (Fig. 9). The scaled sensitivity shows that v is much more sensitive than λ, with a composite scaled sensitivity of 2.95 for v (cell J17 in Fig. 9) and about 0.04 for λ. For this reason, even though v estimate at 11 cm for the separate fit (2.45) is only 2% less than that for the simultaneous fit (2.50), the difference in λ is about 6% (Table 1). More significantly, the coefficient of variation for λ estimate (about 6%) is far greater than that for v estimate (0.09%, Fig. 8). This analysis shows that good control and/or measurement of velocity is important for reliable quantification of dispersivity in this experiment. In addition, both the parameter and prediction ranges can be redefined to include any parameter and any prediction for the sensitivity analysis in CXTFIT/Excel. This can be useful for experimental design.

For the same example, propagation of uncertainty in v and λ estimates on model predictions is calculated and output next to the prediction range (Fig. 10) using the macro Propagate. Prediction sensitivity increases with depth and the confidence band becomes significantly wider even though the error in velocity is only 1%. A prediction error of 3% (RMSE=0.03) is employed for the model fitting error in this illustration, t value is set at 2, and Eq. (7a) is used to calculate the prediction uncertainty. The first order error propagation (Eq. (6)) at $x=11$ is compared with Monte Carlo analysis (Fig. 11) where v and λ values are randomly selected between mean plus/minus twice the standard deviation for CV of 1% and 10%. The first order error propagation is close to that by Monte Carlo analysis for CV of 1%. For CV of 10%, the first order method overestimates the error propagation in the breakthrough limb where the prediction is most sensitive to the parameters (Fig. 9). The prediction uncertainty analysis functions enable CXTFIT/Excel to investigate the effect of uncertainty in model parameters and their co-variances on model predictions, not only for in-sample data used for calibration, but more importantly for out-of-sample predictions, for example at greater travel times and distances than considered during calibration.

5. Summary

The analytical solutions for the one-dimensional equilibrium and nonequilibrium convection dispersion equations (CDE and MIM) were coded as VBA functions, which may be used as ordinary math functions for the forward solutions in Excel. Objective function minimization, sensitivity analysis, uncertainty analysis, error propagation, response surface calculation, and Monte Carlo analysis functions were programmed into VBA subroutines, which can be used as macros. User-friendly interfaces were designed to provide for flexibility and ease of use to define input and output for the macros. The code was verified by a number of benchmarks in CXTFIT 2.0 (Toride et al., 1995). Implementation of CXTFIT in Excel provides improved flexibility in setting up parameter estimation problems, conducting sensitivity and uncertainty analysis, and facilitating model evaluation and comparison. Improvements over previous versions include the capability for simultaneous fitting parameters for multi-tracer data sets, incorporating observation uncertainty and prior information on parameter estimates and exerting no limitations on model parameterization—i.e., any primary or derived variable that occurs within the model with desired transformation (e.g., log-transformed, inverted, etc.) can be estimated with sensitivity and uncertainty quantification. Ordinary Excel users who can use math functions as formulae input will be able to employ the full potential of CXTFIT/Excel to significantly improve both simple and

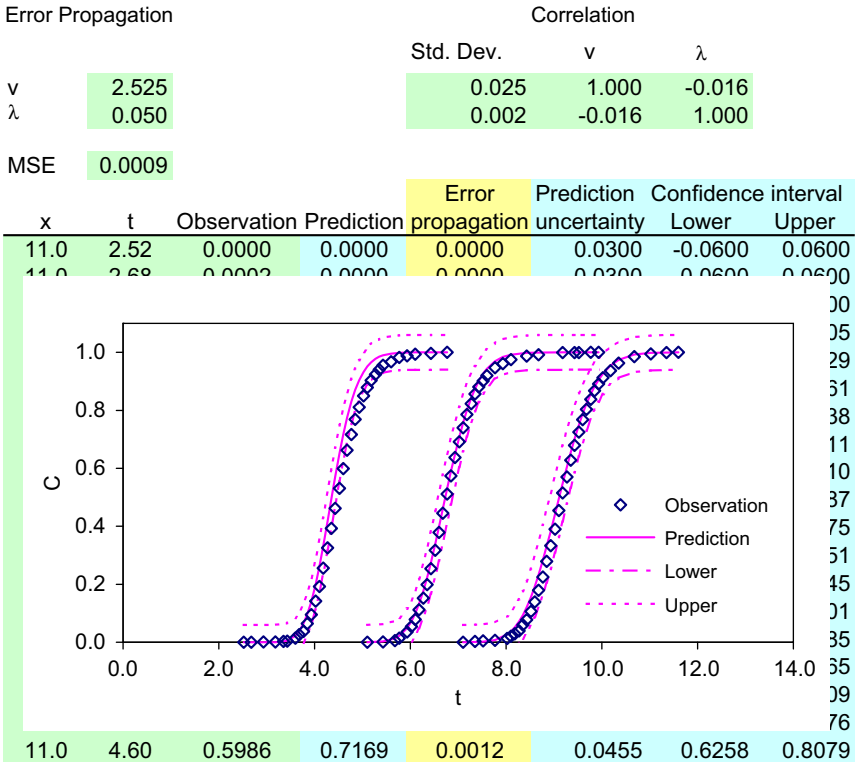


Fig. 10. Error propagation and prediction uncertainty for example in Fig. 8.

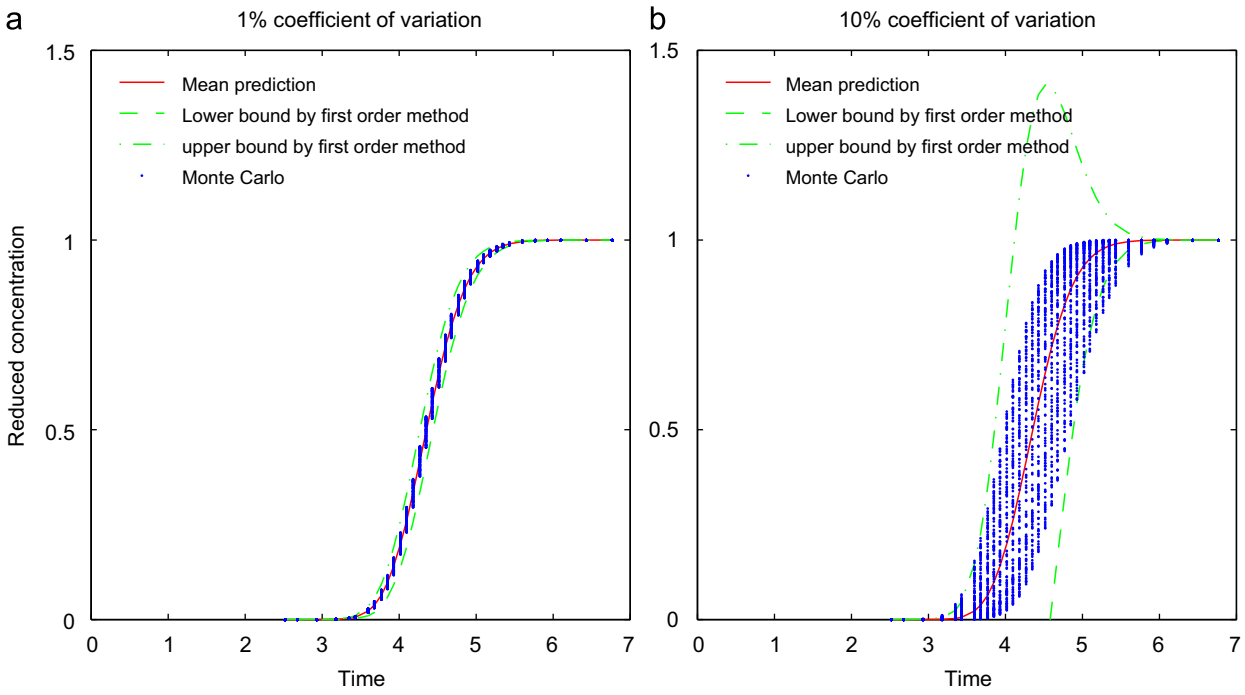


Fig. 11. Comparison of first order error propagation at $x=11$ in λ. 10 with Monte Carlo analysis results. Parameters are sampled randomly between $(1 \pm 2CV)$ times mean value following a uniform distribution.

complicated tracer experiment data interpretation, which will enable us to better understand the processes that control contaminant migration and to better predict the fate and transport of contaminants in the subsurface.

Furthermore, the CXTFIT/Excel optimization and error analysis macros and interfaces can be applied to solve virtually any problem for which the direct solution can be formulated either directly in Excel or coded into an Excel/VBA function. For example, Parker et al. (2008) incorporated a VBA function for transport in a 2-D aquifer with time-dependent source dissolution in CXTFIT/Excel to calibrate model parameters from field-scale monitoring data and perform forward predictions of uncertainty in long-term remediation performance. CXTFIT/Excel can easily be used to estimate saturated and unsaturated hydraulic properties from field and laboratory experiments, to fit sorption and kinetic models to batch or stirred-tank reactors, or to analyze many other calibration/error analysis problems.

Acknowledgements

We appreciate the review comments provided by Dr. Philip Meyer and another anonymous reviewer, which significantly helped improve this paper. This research was funded by the U.S. Department of Energy, Office of Science, Office of the Biological and Environmental Research. Oak Ridge National Laboratory is managed by the University of Tennessee-Battelle, LLC, under contract DE-AC05-00OR22725 with the US DOE.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.cageo.2010.01.013.

References

Doherty, J., 2004. PEST: model independent parameter estimation–user manual, version 5, Watermark Numerical Computing 336pp.
Gander, W., Gautschi, W., 2000. Adaptive quadrature—revisited. BIT 40 (1), 84–101.
Hill, M.C., Tiedeman, C.R., 2007. Effective Groundwater Model Calibration—with Analysis of Data, Sensitivities, Predictions, and Uncertainty. Wiley, Hoboken, New Jersey 455pp.
Jelen, B., Syrstad, T., 2004. VBA and Macros for Microsoft Excel. Sams Publishing, Indianapolis, Indiana 550pp.
Neter, J., Kutner, M.H., Nachtsheim, C.J., Wasserman, W., 1996. Applied Linear Regression Models 3rd Edn. IRWIN, Chicago, Illinois 720pp.
Nützmann G., Holzbecher E., Wiese B., Licht E., Knappe A., 2005. Visual CXTFIT—a user friendly simulation tool for modelling one-dimensional transport, sorption and degradation processes during bank filtration. In: Proceedings of the Fifth International Symposium on Management of Aquifer Recharge ISMAR5, Berlin, Germany, pp. 409–414.
Parker, J.C., van Genuchten, M.Th, 1984a. Flux-averaged and volume-averaged concentrations in continuum approaches to solute transport. Water Resources Research 20, 866–872.
Parker, J.C., van Genuchten, M.Th, 1984b. Determining Transport Parameters from Laboratory or Field Tracer Experiments, Bulletin 84–3. Virginia Agriculture Experiment Station, Blacksburg, Virginia 95pp.
Parker, J.C., Park, E., Tang, G., 2008. Dissolved plume attenuation with DNAPL source remediation, aqueous decay and volatilization—Analytical solution, model calibration and prediction uncertainty. Journal of Contaminant Hydrology 102, 61–71.
Press, W.H., Teukolsky, S.A., Vetterling, W.T., 1992. Numerical Recipes in Fortran: the Art of Scientific Computing. Cambridge University Press, New York, NY 992pp.
Sambridge, M., Mosegaard, K., 2002. Monte Carlo methods in geophysical inverse problems. Reviews of Geophysics 40 (3), 1–29.
Tang, G., Mayes, M.A., Parker, J.C., Yin, X.L., Watson, D.B., Jardine, P.M., 2009. Improving parameter estimation for column experiments by multi-model evaluation and comparison. Journal of Hydrology 376 (3–4), 567–578. doi:10.1016/j.jhydrol.2009.07.063.
Toride, N., Leij, F.J., van Genuchten, M.T.h, 1995. The CXTFIT Code for Estimating Transport Parameters from Laboratory or Field Tracer Experiments, Version 2.0. U.S. Salinity Laboratory, Agricultural Research Services, U. S. Department of Agriculture, Riverside, CA 121pp.
Wraith, J.M., Or, D., 1998. Nonlinear parameter estimation using spreadsheet software. Journal of Natural Resources and Life Sciences Education 27, 13–19.