Fitting Supercritical Fluid Extraction Models by means of a Derivative-free Software *

Esdras P. Carvalho † Julián Martínez ‡ J. M. Martínez § Feodor Pisnitchenko ¶

Abstract

General Supercritical Fluid Extraction (SFE) models are given by systems of partial differential equations (PDE). The SFE process consists of two steps: extraction and separation. In this work we deal with the extraction step. In this step, a solvent flows through a column filled with a solid substrate fixed bed, which contains the components to be removed. Since there are two phases inside the column, we describe mass balance equations for fluid (solvent) and solid phases, in the following way:

$$\frac{\partial Y}{\partial t} + U\frac{\partial Y}{\partial h} = \frac{\partial}{\partial h}\left(D_{aY}\frac{\partial Y}{\partial h}\right) + \frac{J(X,Y)}{\varepsilon}, \tag{1}$$

$$\frac{\partial X}{\partial t} = \frac{\partial}{\partial r}\left(D_{aX}\frac{\partial X}{\partial r}\right) - \frac{J(X,Y)}{1-\varepsilon}\frac{\rho}{\rho_S}. \tag{2}$$

The unknowns of the system of equations (1-2) are $X = X(h,r,t)$, the extract concentration in the solid phase, and $Y = Y(h,t)$, the extract concentration in the fluid phase. The variables h, r and t are, respectively, the one-dimensional (space) coordinate of a column with total height equal to H, the radial coordinate of each solid particle and the time. The following quantities appear in (1-2): ε is the extraction bed porosity; ρ, the solvent density (kg/m^3); ρ_S, the solid density (kg/m^3); U, the solvent velocity (m/s), which may be writen as $U = \frac{4Q_{CO_2}}{\pi D_b^2 \varepsilon\rho}$, where D_b (m) is the diameter of the extraction column and Q_{CO_2} (kg/s) is the solvent mass flow rate; D_{aY}, the extract diffusivity in the fluid phase (m^2/s); D_{aX}, the extract diffusivity in the solid phase (m^2/s); and $J(X,Y)$, the interfacial mass transfer flux wich depends on ρ, ρ_S and the *extract solubility in the fluid phase* (Y^*). The definition of $J(X,Y)$ also involves adjustable parameters $c_1,\ldots,c_n$. For solving these coupled equations we need initial conditions on $X(h,r,t=0)$ and $Y(h,t=0)$ and boundary conditions on $X(h,r=0,t), Y(h=0,t)$ and $\frac{\partial Y}{\partial h}(h=H,t)$. The system of equations (1-2) does not possess an analytical solution. Nevertheless, obtaining (numerical) solutions of (1-2) may be important for practical purposes. For obtaining reliable simulations under different conditions of, say, pressure and temperature, one should have good estimates of the constants $c_1,\ldots,c_n$ involved in the definition of $J(X,Y)$. The estimation process for these constants is necessarily based on some empirical observation that, in this case, takes the form of an extraction curve, which is defined by

$$E(t) = Q_{CO_2}\int_0^t Y(H,s)ds. \tag{3}$$

In practical terms, one has m observed values $E(t_1),\ldots,E(t_m)$ and, using these data and the model (1-2) one should estimate the parameters $c_1,\ldots,c_n$. We considered several optimization approaches (Augmented Lagrangean method, PDE-Constrained Optimization, Newton Method) to estimate the parameters but they did not work. So we have been led to use the "black-box" approach by means of which the parameters to be estimated are the only independent variables of the optimization problem and the discretization scheme is a part of the objective function evaluation. In principle, the derivatives of the objective function with respect to parameters may be obtained using recursive (direct or reverse) differentiation. However, the discontinuity of some of functions prevents reliable derivative computations and, so, we decided to rely on derivative-free software. We choose the BOBYQA software by M. J. D. Powell. BOBYQA seems to possess the correct characteristics for reliable fitting of our parameters. This software admits bounds on the independent variables and its strategy consists on minimizing iteratively quadratic models of the objective function, built using stable minimum-norm variation principles. Although its behavior is expected to be better when the objective function is defined as a sum of squares of the errors, it can be used too when the goal is to minimize the (nonsmooth) sum of absolute error values. Using BOBYQA we got very interesting results.

*This work was supported by Fundação Araucária

†Department of Mathematics, State University of Maringá, Brazil, e-mail:epcarvalho@uem.br
‡Food Engineering Department, University of Campinas, Brazil, e-mail: julian@fea.unicamp.br.
§Department of Applied Mathematics, University of Campinas, Brazil, e-mail:martinez@ime.unicamp.br.
¶Center for Mathematics, Computation and Cognition, Federal University of ABC, Brazil, e-mail: feodor@ufabc.edu.br