New Deconvolution Methods for Well Test and Production Data Analysis

M. Andrecut

University of Calgary, BI547, 2500 University Drive NW,
Calgary, Alberta, T2N 1N4, Canada

Email address: mandrecu@ucalgary.ca

Abstract

The deconvolution method has received much attention recently, and is becoming one of the major tools for well test and production data analysis. Here, we present several new deconvolution algorithms, which we believe that are relevant and can be an important addition to the existing efforts made in this field. We show that the solution of the deconvolution problem can be successfully represented as a linear combination of exponential basis functions. We present three deconvolution algorithms. The first two algorithms are based on regularization concepts borrowed from the well-known Tikhonov and Krylov methods, while the third algorithm is based on the stochastic Monte Carlo method.

Introduction

Deconvolution is a mathematical method that can be used to convert the pressure data response from a variable-rate test, or production sequence, into a constant-rate equivalent pressure profile for the entire duration of the test or production history, thus making the data more useful for interpretation [1-9]. The deconvolution method allows one to extract more information from well test and production data than is possible by using conventional analysis methods [10]. For example, the Bourdet derivative plot [11], displays the pressure behavior only for a specific flow period of a test, while the result of deconvolution can give the transient pressure behavior for a group of flow periods. Therefore, the deconvolved system response is defined on a longer time interval, and reveals characteristics of the transient behavior that otherwise would not be observed using the conventional methods. Due to these reasons, the pressure-rate deconvolution problem has received considerable interest over the past decades. Unfortunately, applying deconvolution to well test and production data analysis is a difficult challenge because it requires the solution of an ill-conditioned problem. This ill-conditioned problem, combined with the errors that are inherent in pressure and rate measurements, makes the development of robust (error-tolerant) deconvolution algorithms a real-challenge. During the past several decades, a large number of algorithms have been proposed [1-8]. However, only a few, recently developed algorithms [6-8], appear to be somewhat more robust [9].
The goal of this work is not to analyse the performance of the existing algorithms, but to identify and explore several new algorithms, which we believe that are relevant and can be important additions to the existing efforts made in this field. We are interested in simple and robust algorithms, which do not require too many parameters for tuning, and can be easily implemented in a software product.

In this article we show that the solution of the deconvolution problem can be successfully represented as a linear combination of exponential basis functions. This approach is justified by the fact that in well-test and production data analysis we are interested in the transient response, which for a stable linear system is a linear combination of exponential functions.

The first two algorithms are based on regularization concepts borrowed from the well-known Tikhonov [12] and Krylov [13-14] methods, while the third one is a new stochastic Monte Carlo algorithm.

Pressure-rate deconvolution model

In the well test theory, the input signal is usually a step function in rate created by closing a flowing well or an injection well, by opening a well previously shut-in or by injecting in a well previously closed. The corresponding output signal is the change in pressure created by the change in rate. Alternatively, the input signal could be the wellhead bottomhole pressure and the output signal would then be the change in the production rate. The way the input signal is created is not important as far as well test analysis is concerned. What is important is the quality of both measured input and output signals. In this section we give a short discussion of the pressure-rate deconvolution model in the framework of linear systems theory [15-16].

In general, a linear system is a functional transformation, L, which converts an input signal, $f(t)$ to an output signal, $g(t)$:

$$g(t) = L[f(t)]$$

and which follows the principles of superposition

$$L[f_1(t) + f_2(t)] = L[f_1(t)] + L[f_2(t)]$$

and amplitude scaling

$$L[\alpha f(t)] = \alpha L[f(t)],$$

where α is a scalar. The amplitude scaling also implies that the output of the system is zero when there is no input:

$$L[0] = 0$$

Here, we consider systems that are:

- time-invariant, i.e., the functionality of L is not time dependent.
- causal, i.e., the output at time t_0 depends only on values of the input for $t \le t_0$.
- stable, i.e., every non-infinite input produces a non-infinite output.

These simple rules, defining the linear systems, provide far-ranging and very useful constraints on the mathematical characterization of the system.

Several idealized input functions are of special importance in analyzing systems: the Dirac impulse function and the Heaviside unit step function. The Heaviside step function is defined as:

$$\Theta(t) = \begin{cases} 0 & t < 0 \\ 1 & t \ge 0 \end{cases},$$

while the Dirac impulse function is its derivative:

$$\delta(t) = \frac{d}{dt}\Theta(t).$$

The Dirac function has the following fundamental properties:

$$\int_{\tau-\varepsilon}^{\tau+\varepsilon} f(t)\delta(t-\tau)dt = f(\tau), \quad \forall \varepsilon > 0$$

$$\delta(t-\tau) = 0 \quad if \quad t \ne \tau .$$

The impulse response of a system is the output produced by a Dirac impulse input

$$h(t) = L[\delta(t)]$$

Similarly, we have the step response, or the response to the unit step function

$$H(t) = L[\Theta(t)].$$

In general, we are interested in the response of a system not to these special functions, but to an arbitrary input. The trick is to approximate this arbitrary input as a sequence of step functions, and then to use the superposition principle to obtain the overall response. It is necessary to assume that the input function $f(t)$ is continuous, so that the approximation is meaningful. Suppose the input starts at $t = 0$, and time is discretized into bins Δ units apart. The first step in the approximation has height

$f(0)$, a constant, and the additional step at $t = k\Delta$ has height $f(k\Delta) - f((k-1)\Delta)$. The signal $f(t)$ can thus be approximated by:

$$f(t) = f(0)\Theta(t) + \sum_{k=1}^{K}[f(k\Delta) - f((k-1)\Delta)]\Theta(t - k\Delta) .$$

The approximate output is then given by

$$g(t) = L[f(t)]$$

$$= f(0)L[\Theta(t)] + \sum_{k=1}^{K}[f(k\Delta) - f((k-1)\Delta)]L[\Theta(t - k\Delta)]$$

$$= f(0)H(t) + \sum_{k=1}^{K}[f(k\Delta) - f((k-1)\Delta)]H(t - k\Delta)$$

$$= f(0)H(t) + \sum_{k=1}^{K}\frac{f(k\Delta) - f((k-1)\Delta)}{\Delta}H(t - k\Delta)\Delta$$

Now, we introduce a limiting process in which the time steps become vanishingly close, so that $\Delta \to d\tau$, $k\Delta \to \tau$ and $K\Delta \to t$. Thus, the sum becomes an integral and we have:

$$g(t) = f(0)H(t) + \int_0^t \frac{df(\tau)}{d\tau}H(t-\tau)d\tau$$

$$= f(t)H(0) - \int_0^t f(\tau)\frac{d}{d\tau}H(t-\tau)d\tau$$

$$= -\int_0^t f(\tau)\frac{d}{d\tau}L[\Theta(t-\tau)]d\tau$$

$$= -\int_0^t f(\tau)L[\frac{d}{d\tau}\Theta(t-\tau)]d\tau$$

$$= \int_0^t f(\tau)L[\delta(t-\tau)]d\tau$$

$$= \int_0^t f(\tau)h(t-\tau)d\tau$$

$$= \int_0^t f(t-\tau)h(\tau)d\tau$$

because $H(0) = 0$.

Now, let us assume that the input signal $f(t)$ corresponds to the flow rate $q(t)$ and the output signal is the difference between the initial pressure p_i and the pressure $p(t)$ at an elapsed time t. Therefore, the impulse response $h(t)$ corresponds to the time derivative of the unit (constant) rate pressure $H(t) = p_u(t)$:

$$h(t) = \frac{dp_u}{dt} .$$

With these assumptions we obtain the following convolution integral:

$$p(t) = p_i - \int_0^t q(t-\tau)h(\tau)d\tau .$$

The above integral equation is known as Duhamel's principle and it represents the foundation of well test analysis [1-11]. This is also equivalent to a Volterra integral equation of the first kind [17].

The crucial step in well test analysis is to calculate the logarithmic derivative of the unit step response [1-11]:

$$G(t)=\frac{dH(t)}{d\ln t}=t\frac{dH(t)}{dt}=t\frac{d}{dt}L[\Theta(t)]=tL[\frac{d}{dt}\Theta(t)]=tL[\delta(t)]=th(t),$$

from measurements of rate and pressure drop. Up to multiplication with time, the task is to estimate the impulse response $h(t)$ from the convolution integral. Therefore, the objective of the deconvolution process is to recover the impulse response $h(t)$, given the observed measurements of pressure and flow rate. This leads to the *inverse problem* in which the time dependent signal $h(t)$ must be extracted from the output signal $p(t)$ given the flow rate $q(t)$.

Inverse problems are a notorious source of ill-posed problems [18-19]. The numerical solution of these problems involves some kind of discretization, which in turn originates a class of problems known as discrete ill-posed problems, which are very ill-conditioned. The fact that ill conditioning is an intrinsic feature of these problems makes it necessary to develop special numerical methods to treat them. Also, there are a number of issues that have to be considered before proceeding with the deconvolution of well-test data: (i) the deconvolution will produce a meaningful system response function only if the test pressure and rate data are consistent with the superposition model (ii) the superposition principle is valid only for linear systems, governed by linear equations (iii) the linearity implies single-phase flow in the reservoir (in the case of gas reservoir one has to use pseudo-pressure transform to linearize the fluid flow problem). Therefore, it is important to use only the portions of pressure data that are of good quality and are consistent with the superposition principle. It is important to note that a good deconvolution method must be able to point out the inconsistencies in the data. Therefore, the deconvolution can be also used for detecting inconsistencies between rate and pressure data.

Another important aspect deals with the solution itself. It is well known that in general the response $y(t)$ of a stable linear system to a general input $f(t)$ is the sum of a homogenous component and a particular solution $y_p(t)$ [15-16]:

$$y(t)=\sum_{j=1}^{J}x_j\exp(\lambda_j t)+y_p(t),$$

where J is the system order, λ_j are the roots of the distinct characteristic equation (the system's eigenvalues) and x_j are the constants depending on the initial condition. For a stable system all the terms $\exp(\lambda_j t)$ must decay to zero, this means that all the eigenvalues must be negative $\lambda_j<0$. Also, it is convenient to consider the total response $y(t)$ in two regions, an initial transient region in which the exponential homogeneous components $\exp(\lambda_j t)$ must be considered, followed by the steady-state region when the homogeneous solution components have all decayed to the point of becoming insignificant. In the steady-state region, only the particular response component $y_p(t)$ of the response is considered.

In well test and production data analysis, we are interested only in the transient response, and therefore we should be able to represent the solution as a sum of exponential basis functions:

$$h(t)=\sum_{j=1}^{J} x_j \exp(\lambda_j t) .$$

Unfortunately, we do not know anything about the system itself. We do not know the dimensionality J and we do not know the eigenvalues λ_j. Also, there is no method to find these quantities from the rate and pressure measurements. However, this may not be so important because the solution of the inverse problem is non-unique: several different systems may exist which, for identical input signals, provide identical output signals [18]. This non-uniqueness is a property of the inverse problem that cannot be avoided, and it has significant implications on the methodology for well test analysis. Apparently, it seems that this property is not desirable. However, our approach to the deconvolution problem exploits exactly this property. Instead of searching for a specific system, we design a general system capable to generate the type of responses we are looking for.

We consider a system with a given dimensionality J for which the eigenvalues λ_j are distributed such that by adjusting the coefficients x_j one can generate the class of responses we are looking for. In the well-test and production data analysis, the logarithmic derivative $G(t)=th(t)$ is represented on a log-log plot. Therefore, we seek a distribution of J negative eigenvalues $\lambda_j<0$ such that the functions:

$$\varphi_j(t)=t\exp(\lambda_j t), \quad j=1,...,J$$

give a good decomposition of the logarithmic derivative in the log-log space. Based on several numerical experiments we have concluded that the following set of eigenvalues:

$$\lambda_j=-\alpha(j/J)^\gamma, \quad j=1,...,J$$

where α is a time scaling constant, give a good decomposition. In Fig. 1 we give the graphical representation of the functions $\varphi_j(t)$, for $J=15$, $\gamma=3$ and $\alpha=1$. One can see that the whole time interval is almost uniformly covered by these functions. This means that we can write the solution of the deconvolution problem as a linear composition of exponential functions. Our numerical experiments have shown that for small dimensionality values, $J\approx 10\div 100$, the distribution parameter should be $\gamma\approx 3$. This value decreases by increasing J. For, large $J\approx 1000$ the parameter value should be $\gamma=1$. Also, we have found that a small number of exponential basis functions, typically $J\approx 20$ are enough to represent almost any type of well test responses. The constant α is set to $\alpha=10^\gamma T^{-1}$, where T is the time length of the test, in this case $T=10^3$, $\gamma=3$.

Discrete ill-conditioned Volterra equation

The numerical solution of the deconvolution problem requires the discretization of the continuous Volterra equation [17]. Here, we consider that the solution can be represented as a linear composition:

$$h(t)=\sum_{j=1}^{J} x_j \psi_j(t) ,$$

where

$$\psi_j(t)=\exp[-\alpha(j/J)^\gamma t], \quad j=1,...,J$$

are the exponential basis functions, and

$$x = (x_1, ..., x_J)^T$$

is the unknown vector of coefficients (here, T stands for the transposition of vectors and matrices). This form of the solution leads to the following equation:

$$p(t) = p_i - \sum_{j=1}^{J} x_j \int_0^t q(t-\tau)\psi_j(\tau)d\tau \ .$$

The evaluation of this equation at discrete moments of time:

$$\{t_n \mid n = 0,1,..., N\}$$

yields a system of equations which can be written as following:

$$\Delta p_n = \sum_{j=1}^{J} a_{nj} x_j \ ,$$

where

$$\Delta p_n = p_i - p_n$$

and

$$a_{nj} = \int_0^{t_n} q(t_n - \tau)\psi_j(\tau)d\tau, \quad j = 1,..., N, \quad n = 0,1,..., N$$

are the elements of the sensitivity matrix elements.

Now let us consider the case where the initial pressure is not known, and is treated as a new unknown coefficient $x_0 \equiv p_i$. This leads to the following system:

$$p_n = \sum_{j=0}^{J} b_{nj} x_j \ ,$$

where:

$$b_{nj} = \begin{cases} -\int_0^{t_n} q(t_n - \tau)\psi_j(\tau)d\tau & j = 1,..., N, \quad n = 0,1,..., N \\ 1 & j = 0, \quad n = 0,1,..., N \end{cases} .$$

In a matrix form we have:

$$wx = y \ ,$$

where

$$w = \begin{cases} a & \text{if the initial pressure is known} \\ b & \text{otherwise} \end{cases} ,$$

$$y = \begin{cases} \Delta p & \text{if the initial pressure is known} \\ p & \text{otherwise} \end{cases} .$$

The integral in the sensitivity matrix elements can be approximated by the following simple rectangular rule:

$$a_{nj} = \int_0^{t_n} q(t_n - \tau)\psi_j(\tau)d\tau \approx \sum_{i=0}^{n-1} q_{n-i}\psi_{ji}\Delta_i \ ,$$

where

$$q_{n-i} = q(t_n - t_i), \quad n, i = 0,1,..., N \ ,$$

$$\psi_{ji} = \psi_j(t_i), \quad j = 1,..., J, \quad i = 0,1,..., N \ ,$$

$$\Delta_i = t_{i+1} - t_i, \quad i = 0,..., N-1 \ .$$

The solution of the discrete Volterra equation is:

$$h_n = \sum_{j=1}^{J} x_j \psi_{jn} \ .$$

Also, the reconstructed pressure signal is:

$$P_n = p_i - \sum_{i=0}^{n-1} q_{n-i} h_i \Delta_i \ .$$

The discrete form of the logarithmic derivative must be calculated in the middle of the interval $[t_n, t_{n+1}]$:

$$G_n = 0.5(t_n + t_{n+1})h_n, \quad n = 0,...,N-1 ,$$

because of the rectangular approximation of the integrals in the sensitivity matrix. The discrete ill-conditioned problem has the following non-desirable properties:

- the linear system is usually over-determined $J < N$;
- the system's matrix w has a large cluster of small singular values, which increases with the dimension problem;
- standard inversion method fail to produce a meaningful solution.

Least squares problem

Instead of solving the ill-conditioned system $wx = y$, the least squares problem consists in solving the optimization problem [19-21]:

$$\min_x \|wx - y\|^2 ,$$

which gives an approximate solution (here $\|\cdot\|$ is the Euclidean norm). The least squares problem has the advantage that the solution with the minimum norm is unique and satisfies the system of normal equations:

$$w^T wx = w^T y$$

An important tool for the analysis of the least squares problem is the Singular Value Decomposition (SVD) of the sensitivity matrix w:

$$w = U\Gamma V^T ,$$

where

$$U^T U = I ,$$

$$V^T V = VV^T = I ,$$

$$\Gamma = diag(\gamma_i) .$$

Here, γ_i are the (non-zero) singular values (eigenvalues), the columns of U are the left singular vectors and the columns of V are the right singular vectors of w. Also, I is the identity matrix. Replacing $w = U\Gamma V^T$ in the system of equations yields the solution:

$$x = V\Gamma^{-1}U^T y = \sum_i \frac{u_i^T y}{\gamma_i} v_i \ .$$

The norm of the least squares solution is therefore given by

$$\|x\| = \sqrt{\sum_i \left(\frac{u_i^T y}{\gamma_i} v_i \right)^2} \ .$$

This norm will not be too large as long as $\left|u_i^T y\right| / \gamma_i < 1$. Also, in discrete ill-conditioned problems the singular vectors have more sign changes in their components as the singular values decrease, i.e. the high frequency components corresponds to small singular values. Therefore, in order to have a solution, $\left|u_i^T y\right|$ must decay to zero faster than γ_i (Picard condition).

Now, let us assume that we are interested in recovering the solution x from $\bar{x}$, which is the solution of

$$\min_x \|a\bar{x} - \bar{y}\|^2$$

where

$$\bar{y} = y + \eta$$

with η a random vector representing the noise. The solution $\bar{x}$ is obviously given by:

$$x = \sum_i \frac{u_i^T y}{\gamma_i} v_i + \sum_i \frac{u_i^T \eta}{\gamma_i} v_i$$

Thus, $\bar{x}$ consists of two terms: one is the actual solution x of the unperturbed problem, and the other is the contribution from the noise. The difficulty of finding x comes from the noise contribution. Since it is not reasonable to expect that η satisfies the Picard condition, it is possible that $|u_i^T \eta|$ increases or becomes constant, causing the ratio $|u_i^T \eta| / \gamma_i$ to blow up, making impossible to recover the solution x.

Tikhonov method

Numerical regularization techniques seek to approximate the exact (unknown) solution of an ill-conditioned problem by the solution of a related well-known problem that includes meaningful information about the solution to the original problem. Here, we consider the well-known Tikhonov regularization method where the additional information is usually expressed as a constraint of the form [12]:

$$\|\sigma x\| \le \varepsilon ,$$

where σ is typically the identity operator I or a discretized derivative operator, and $\varepsilon > 0$ is a small positive quantity. This constraint is used to control the smoothness and the size of the approximate solution. To justify the use of this constraint, one must assume that the exact solution is smooth or that it has small norm. Regularization is also known as smoothing because it tries to damp non-smooth components in the approximate solution. When non-smooth components correspond to small singular values, those components are magnified by the noise. Regularization can also be regarded as a multi-objective optimization problem where one tries to balance the quality of the approximation and the effect of perturbations on the solution. Associated with any regularization method there is a parameter, namely the regularization parameter, which controls how much perturbation effect is allowed. Therefore the regularization of the ill-conditioned system $wx = y$ can be formulated as following:

$$\min_x \{\|wx - y\|^2 + \mu^2 \|\sigma x\|^2\} ,$$

where $\mu > 0$ is the regularization parameter. This is equivalent of solving simultaneously two systems of equations:

$$w^T wx = w^T y$$

with a weight 1, and

$$\sigma x = \varepsilon$$

with a weight μ. Therefore we can write

$$w^T wx - w^T y + \mu(\sigma x - \varepsilon) = 0 ,$$

or equivalently

$$(w^T w + \mu\sigma) x = w^T y + \mu\varepsilon .$$

One can neglect the residual $0 < \mu\varepsilon << 1$ (which is a small quantity) and thus the solution of the regularized problem is given by:

$$x = (w^T w + \mu\sigma)^{-1} w^T y .$$

Note that

$$w^+ = \lim_{\mu \to 0} (w^T w + \mu\sigma)^{-1} w^T$$

is the pseudo-inverse of w. Therefore for small values of μ one can recover meaningful information on the solution of initial system, which is obviously given by:

$$x = w^+ y .$$

The numerical implementation of the Tikhonov regularization method requires the calculation of the inverse matrix $(w^T w + \mu\sigma)^{-1}$. In our implementation we have used the LU inversion method, which seems to work quite well in this case [22]. Also, we have implemented a simple pre-conditioning method by computing the solution as following [21]:

$$x = (A^{-1} w^T w + \mu r)^{-1} A^{-1} w^T y ,$$

where

$$A = diag(w^T w)$$

is the pre-conditioning matrix. This procedure helps very much, if the matrix $w^T w$ is very ill-conditioned, by making all the diagonal coefficients equal to one.

Krylov method

Another approach to regularization is given by the so-called Krylov subspace projection methods [13-14]. These are iterative methods, developed for symmetric and positive definite linear systems and therefore they can be applied to the normal equations. Krylov subspace projection methods are non-stationary, meaning that the relationship between iterations is not given by a fixed matrix. From this point of view, Krylov methods are a major innovation in numerical linear algebra. The basic methods of this type are the Lanczos method and the conjugate gradient method. These two methods are essentially the same method applied in different ways. The Lanczos method generates an approximate tridiagonalization of $W = w^T w$, which then can be used to solve a tridiagonal system. The obtained solution is equivalent to the conjugate gradient solution. The conjugate gradient method has an advantage, because it needs less storage. Therefore, our choice is the conjugate gradient method, for which we give a short description of our implementation.

We consider a linear system of the form:

$$Wx = z ,$$

where

$$W = w^T w$$

is a symmetric positive definite matrix and

$$z = w^T y$$

is a given vector.

We say that two non-zero vectors u and v are conjugate with respect to W if:

$$u^T W v = 0 .$$

Since W is symmetric and positive definite, the left-hand side defines an inner product:

$$(u, v)_W = u^T W v .$$

So, two vectors are conjugate if they are orthogonal with respect to this inner product. Suppose that $\left\{d^{(k)} \mid k = 1,2,...\right\}$ is a sequence of mutually conjugate directions. Then $\left\{d^{(k)}\right\}$ form a basis, so we can expand the solution of $Wx = z$ in this basis:

$$x \approx x^{(k)} = \alpha_1 d^{(1)} + \alpha_2 d^{(2)} + ... + \alpha_k d^{(k)}.$$

This gives the following method for solving the system $Wx = z$. First we find a sequence of conjugate directions $\left\{d^{(k)}\right\}$ and then we compute the coefficients $\{\alpha_k\}$. This is the essence of the conjugate gradient method. At the k iteration step, the conjugate gradient method finds a vector in the Krylov subspace

$$span\{d^{(1)}, d^{(2)}..., d^{(k)}\}$$
$$= span\{d^{(1)}, Wd^{(1)}, ..., W^{k-1}d^{(1)}\}$$
$$\equiv K_k(W, d^{(1)}, k),$$

which minimizes the functional

$$f(x) = \frac{1}{2}x^T Wx - z^T x.$$

Obviously, we have:

$$\frac{d}{dx}f(x) = Wx - z.$$

The above derivative vanishes if x is the solution of $Wx = z$. If we take another derivative

$$\frac{d^2}{dx^2}f(x) = W,$$

we obtain the matrix W which is positive definite ($W = w^T w$), so that $f(x)$ can only increase if the vector x moves away from $W^{-1}z$. This shows that $W^{-1}z$ is the global minimum of the functional $f(x)$. The sequence of vectors $\{x^{(k)}\}$ lies in increasingly large Krylov subspaces. Because the conjugate gradient method minimizes the functional $f(x)$ over these subspaces, this sequence of vectors tends to approach the solution $x = W^{-1}z$. When the iteration count k is equal to the size of the system then, as a consequence of the Cayley-Hamilton theorem [20-21], the Krylov subspace becomes equal to the whole system space, and the vector $x^{(k)}$ must be equal to $W^{-1}z$. If z lies in a proper invariant subspace of the matrix W, then exact convergence occurs sooner.

The implementation of conjugate gradient method should avoid forming the matrix $W = w^T w$, since doing so may introduce large rounding errors. Thus, any product Wv should be calculated in two steps: $u = wv$ and then $w^T u$.

When the iterations start approximating singular vectors associated to small values, contributions from noise appear and the solution start to diverge at that point. This behaviour is known as *semi-convergence*. It is necessary to stop the iteration before the effect of noise appears. Thus, the number of iterations plays the role of the regularization parameter.

The semi-convergence behavior suggests the following stopping criterion: monitor the residual norm and stop when it starts to increase. This is unfortunately not a very good stopping criterion. When the method reaches the minimum point, the residual becomes zero, and if the method is applied for one more step then a division by zero will result. It seems that one must stop immediately when the residual is zero. To complicate things, accumulated errors in the recursive calculation of the residual may yield a false zero residual. Usually, however, one wishes to stop before convergence is complete. Because the error term is not available, it is customary to stop when the norm of the residual falls below a specified value; often, this value is some small fraction $\varepsilon > 0$ of the initial residual:

$$\left\| r^{(k)} \right\| < \varepsilon \left\| r^{(0)} \right\|$$

In this case, the small parameter ε plays the role of the regularization parameter. Below we give the pseudo-code of our implemented version of the conjugate gradient method:

```
1. Initialize $\varepsilon$ and $k_{\max}$
2. $x \leftarrow 0$
3. $u \leftarrow wx$
4. $r \leftarrow w^T z - wu$
5. $d \leftarrow r$
6. $\varepsilon \leftarrow \varepsilon \|r\|$
7. $k \leftarrow 1$
8. while ($\|r\| < \varepsilon$ and $k < k_{\max}$) do
    a) $u \leftarrow wd$
    b) $\beta \leftarrow r^T r$
    c) $\alpha \leftarrow \beta / (u^T u)$
    d) $x \leftarrow x + \alpha d$
    e) $u \leftarrow wx$
    f) $r \leftarrow w^T z - wu$
    g) $\gamma \leftarrow r^T r / \beta$
    h) $d \leftarrow r + \gamma d$
    i) $k \leftarrow k + 1$
9. return $x$
```

The vector d is the search direction, since x is obtained from previous approximation by adding a scalar multiple of d. The vector r is the residual error.

Monte Carlo method

One of the main criticisms of the above discussed regularization methods is that one cannot impose constraints on the solution. For example, we would like to be sure that the solution is always positive. Therefore, we would like to solve the following problem:

$$\min_x \|wx - y\|^2 ,$$

subject to

$$x_j \geq 0, \quad j = 1,..., J .$$

Obviously, this problem cannot be solved with the above discussed regularization methods. In order to solve this problem we consider a new stochastic Monte Carlo approach. Our Monte Carlo method is extremely simple and efficient. The only parameters are the amount of error allowed ε, the maximum number of random iterations M and the number of trials K. This is a stochastic method, in which the solution is calculated using a built in mechanism for averaging over a number of iterations and a number of trials. The algorithm also has a built in mechanism for convergence. Our Monte Carlo method has the following pseudo-code:

```
1. Initialize $\varepsilon$, $M$ and $K$
2. $x \leftarrow 0$
3. $\varepsilon \leftarrow \varepsilon \|y\|$
4. For $k = 1,...,K$ do
    a. $r \leftarrow y$
    b. For $m = 1,...,M$ do
        i. $j \leftarrow \mathrm{random}\{1,...,J\}$
        ii. $c \leftarrow r w^{(j)} \|w^{(j)}\|^{-2}$
        iii. If $(x_j + c < 0)$ then $c \leftarrow -x_j$
        iv. $x_j \leftarrow x_j + c$
        v. $r \leftarrow r - c w^{(j)}$
        vi. If $\|r\| < \varepsilon$ then break
5. $x \leftarrow x/K$
6. return $x$
```

The algorithm begins with initializing the desired error ε (the minimum proportion of y that can be left in the residual r), the maximum number of iterations M (allowed per trial) and the number of trials K. Next, the vector of unknown coefficients x is initialized to zero. For every trial $k = 1,...,K$ one computes a solution of the problem. Each solution is calculated by initializing the residual with the right vector, y, and performing a number of maximum M random iterations in order to minimize this residual. First, a random direction j is selected with equal probability. Then one computes the projection c of the residual on this direction. Here, $w^{(j)}$ is the column vector j of the matrix system w. If the corresponding coefficient x_j becomes negative (after updating), then the projection is limited such that the constraint is satisfied. The corresponding coefficient x_j and the residual are updated with the resulted projection. If the residual becomes smaller than the desired error, then the current solution calculation stops. One can see that the algorithm has an intrinsic mechanism for calculating the average of all K solutions.

Let us analyze the residual projection in more detail. Assuming that r is the current residual and j is the randomly selected direction, we can decompose the current residual on two orthogonal directions, one corresponding to the direction $w^{(j)}$ and one corresponding to the future residual r':

$$r = \left(r \cdot \frac{w^{(j)}}{\|w^j\|} \right) \frac{w^{(j)}}{\|w^j\|} + r'$$

Because $w^{(j)}$ and r' are orthogonal, we have:

$$\|r\|^2 = \left(r \cdot \frac{w^{(j)}}{\|w^j\|} \right)^2 + \|r'\|^2 = \|r\|^2 \cos^2(r, w) + \|r'\|^2$$

The norm of the residual is therefore always decreasing:

$$\|r'\|^2 = \|r\|^2 [1 - \cos^2(r, w)] \le \|r\|^2,$$

and the algorithm converges. This is the intrinsic, built in mechanism for convergence. Thus, our Monte Carlo algorithm always converges to a solution.

Numerical results and discussion

We have applied the proposed deconvolution methods to several synthetic data cases. Both regularization methods, Tikhonov and Krylov, have performed well, showing very similar performances. However, the Monte Carlo method has shown superior performance in the presence of noise.

These artificially created cases were considered mostly for their mathematical and numerical interest. Their purpose is solely to test the performance of the proposed deconvolution methods. Here, we do not give any interpretation and identification of the flow regimes observed on the solution. Identifying flow regimes, which appear as characteristic patterns displayed by the pressure derivative data, is important because a regime is the geometry of the flow streamlines in the tested formation. There are eight such common subsurface flow regimes on log-log plots (Fig. 2). For each flow regime identified, a set of well or reservoir parameters can be computed using only the portion of the transient data that exhibits the characteristic pattern behavior. We have performed a large number of synthetic tests. Here, we give only four such cases with the purpose of illustrating our methods (Figs. 3-6). One can see that most of the flow regimes can be identified in these cases. Also, the fourth case is a long history production case.

In all analyzed cases, the data points are equally spaced in time and the initial pressure is 5000 Psi. First, we have considered the situation in which the noise level is zero. One can see that the pressure signal is accurately reconstructed. The logarithmic derivative (G), calculated using the deconvolution methods corresponds perfectly to the values obtained using the Bourdet derivative ($\tilde{G}$). The deconvolution methods have the advantage of obtaining the derivative for the whole test duration, while the Bourdet derivative gives only a part of these values. We have considered the same cases but we added 5% noise to the pressure values. One can see that the Bourdet derivative cannot provide any meaningful information in this case, while the deconvolution results are still very good. In the last situation, we have the same amount of noise as before, but the deconvolution was performed using only the build-up pressure points (the last build-up for the fourth case Fig. 6). In all described situations, the initial pressure was treated as an unknown. The calculated value was $p_i \approx 5000 \pm 3$ Psi (depending on the noise level), which is very close to the real value.

Our calculations have shown that, for all considered cases, a number of $J = 15 \div 20$ exponential basis functions with $\gamma = 3$ are enough to obtain good results. The range for the regularization parameters is: $\mu = 10^{-3} \div 10^{-10}$ for the Tikhonov method, respectively $k_{\max} = N$ and $\varepsilon = 10^{-4} \div 10^{-10}$ for the conjugate gradient method. The values of the regularization parameters (μ and ε) increase with the level of noise in the data. Therefore, the lower range values are for clean data (zero noise), while the higher range values are appropriate for noisy data. For the Monte Carlo method we have used the following parameters $\varepsilon = 10^{-3}$, $M = 10^5$ and $K = 10$.

We have applied the presented methods to two sets of real field data. The best results have been obtained with the Monte Carlo method and are shown in Figs. 7-8.

The first case (Fig. 7) is a well test in which we know the rate signal for the whole test period and we know the pressure signal only for the build-up. There are $N = 2802$ data points, however the pressure is known only for the last 2502 points. The initial pressure is 18600 Psi. One can see that the pressure signal is very well reconstructed

for the known and the unknown period. Also, the logarithmic derivative looks very good, showing the expected flowing regimes.
The second case is a long history production case, where the initial pressure is 1330 Psi and all the history data of rate and pressure is known. There are $N = 8655$ data points. One can see that the reconstruction is very good and the logarithmic derivative is very well defined on the whole time interval.

Conclusions

We have shown that the solution of the deconvolution problem can be successfully represented as a sum of exponential basis functions. This approach is justified by the fact that in well-test and production data analysis we are interested in the transient response, which for a stable linear system is a linear combination of exponential functions. Also, we have derived three deconvolution algorithms. The first two algorithms are based on regularization concepts borrowed from the well-known Tikhonov and Krylov methods, while the third one is a new stochastic Monte Carlo algorithm.
Tikhonov regularization is stable and feasible only for small to average number of data points, due to the large round-off error accumulations in the matrix inversion step (if the number of data points increases). The conjugate gradient method is recommended for small to large number of data points. There are other several advantages of the conjugate gradient method:
- there is no need to perform the inversion of the system's matrix;
- the system's matrix is involved only in matrix-vector product operations;
- requires very little storage;
- it has a fast convergence.

The best results have been obtained with the Monte Carlo method which has several advantages:
- it can handle easily large amount of data;
- the data is involved only in vector-vector computation;
- the requirements for storage are very low;
- there is no need for additional tuning parameters;
- it is very robust to noise and it does not get stuck in local minima like the deterministic methods.

However, the Monte Carlo method is slower than the other methods.
We have considered the case in which the initial pressure is not known, and therefore it must be treated as one of the variables in the problem. In this case, our analysis has shown again that the exponential basis functions decomposition method gives a robust, meaningful solution in the presence of moderate levels of noise (<5%).

Acknowledgements

The authors would like to thank L. Mattar, K. Zaoral and S. Farshidi from Fekete Associates Inc. (Calgary) for useful discussions and for providing the data used for testing the proposed deconvolution methods.

References

1. Thompson L G, Reynolds A C: **Analysis of variable rate well-test pressure data using Duhamel’s principle.** *SPEFE (October 1986) 453.*

2. Roumboutsos A, Stewart G: **A direct deconvolution or convolution algorithm for well test analysis**, *paper SPE 18157 presented at the 1988 SPE Annual Technical Conference and Exhibition, Houston, 2–5 October.*
3. Kuchuk, F J, Carter, R G , Ayestaran, L: **Deconvolution of wellbore pressure and flow rate**, *SPEFE (March 1990) 53.*
4. Fair P S, Simmons J F: **Novel well testing applications of Laplace transform deconvolution**, *paper SPE 24716 presented at the 1992 SPE Annual Technical Conference and Exhibition, Washington, DC, 4–7 October.*
5. Baygün B, Kuchuk F J, Arikan O: **Deconvolution under normalized autocorrelation constraints**, *SPEJ (September 1997) 246.*
6. von Schroeter T, Hollaender F, Gringarten A C: **Deconvolution of well test data as a nonlinear total least squares problem**, *SPEJ (December 2004) 375.*
7. Levitan M M, Crawford G E, Hardwick A: **Practical considerations for pressure-rate deconvolution of well-test data**, *SPEJ (March 2006) 35.*
8. Ilk D, Valkó P P, Blasingame T A, **Deconvolution of variable rate reservoir performance data using B-splines**, paper *SPE 95571 presented at the 2005 SPE Annual Technical Conference and Exhibition, Dallas, 9-12 October.*
9. Cinar M, Ilk D, Onur M, Valkó P P, Blasingame T A: **A comparative study of recent robust deconvolution algorithms for well-test and production-data analysis**, paper *SPE 102575, presented on the 2006 SPE Annual Technical Conference and Exhibition, San Antonio, 24-27 October.*
10. Bourdet D, Ayoub J A, Pirard Y M: **Use of pressure derivative in well test interpretation**, *paper SPE 12777, presented at the 1984 SPE California Regional Meeting, Long Beach, 11–13 April.*
11. Bourdet D: ***Well Test Analysis: The Use of Advanced Interpretation Models****, Elsevier, Amsterdam 2002.*
12. Tikhonov A N, Arsenin V A: ***Solution of Ill-Posed Problems****, Winston & Sons, Washington, 1977.*
13. Hestenes M R, Stiefel E: **Methods of conjugate gradients for solving linear systems**, *J. Research Nat. Bur. Standards 1952, 49: 409–436.*
14. van der Vorst H: ***Iterative Krylov Methods for Large Linear Systems****, Cambridge Monographs on Applied and Computational Mathematics, Cambridge, 2003.*
15. Bay J S: ***Fundamentals of Linear State Space Systems****, McGraw Hill, 1999.*
16. Antsaklis P J, Michel A N: ***Linear systems****, McGraw Hill, 1997.*
17. Press W H, Flannery B P, Teukolsky S A, Vetterling W T: ***Numerical Recipes in C: The Art of Scientific Computing****, Cambridge University Press, Cambridge, 1992.*
18. Tarantola A: ***Inverse Problem Theory****, SIAM, Philadelphia, 2005.*
19. Bjork A: ***Numerical Methods for Least Squares Problems****, SIAM, Philadelphia (1996).*
20. Golub G H, Van Loan C F: ***Matrix Computations****, The John Hopkins University Press: Baltimore, MD, 1996.*
21. Cheney W, Kinkaid D: ***Numerical Mathematics and Computing****, Thomson Brooks/Cole, Pacific Grove, CA, 2004.*
22. M. Andrecut, ***Introductory Numerical Analysis****, Univ. Publishers, Parkland, 2000; (second edition) Overseas Press India, New Delhi, 2006.*

Figures

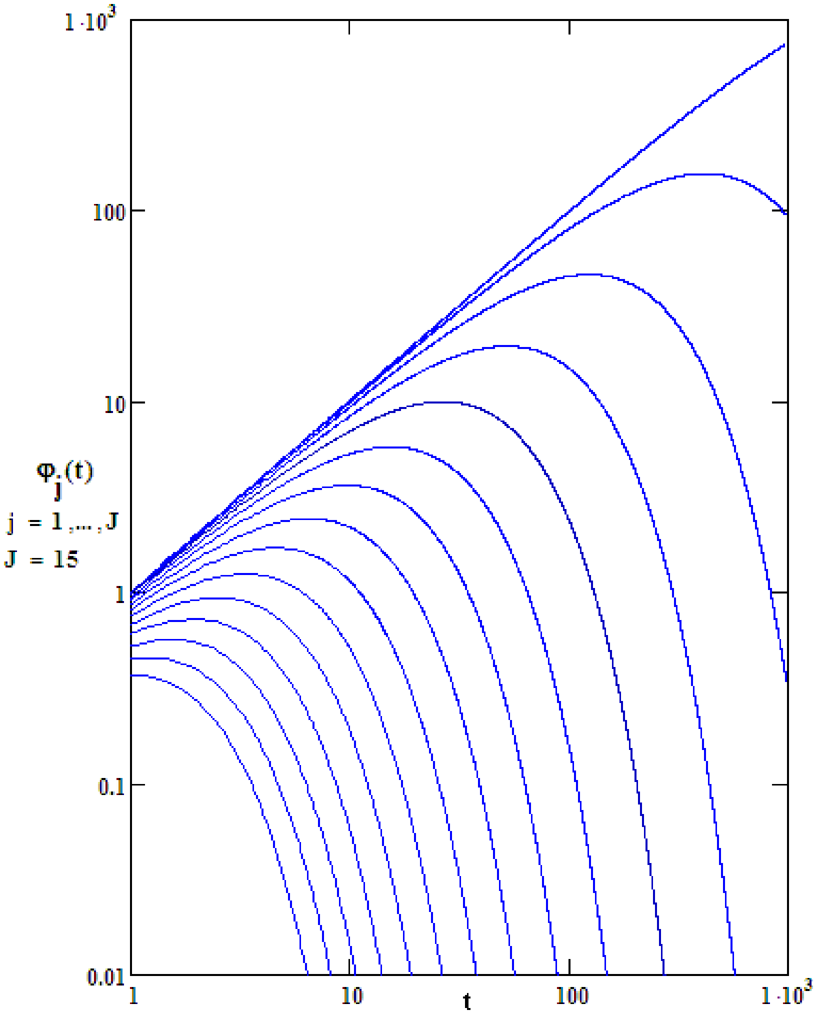

Figure 1 - Exponential basis functions on a log-log plot scale.

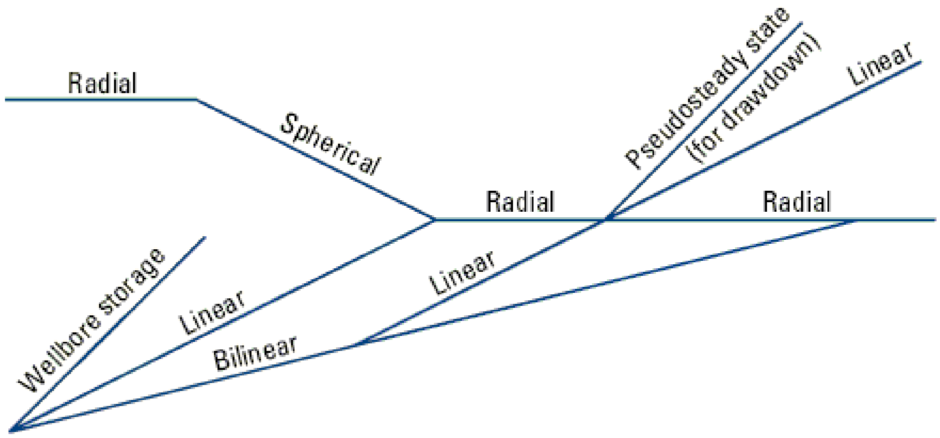

Figure 2 - Flow regime identification tool.

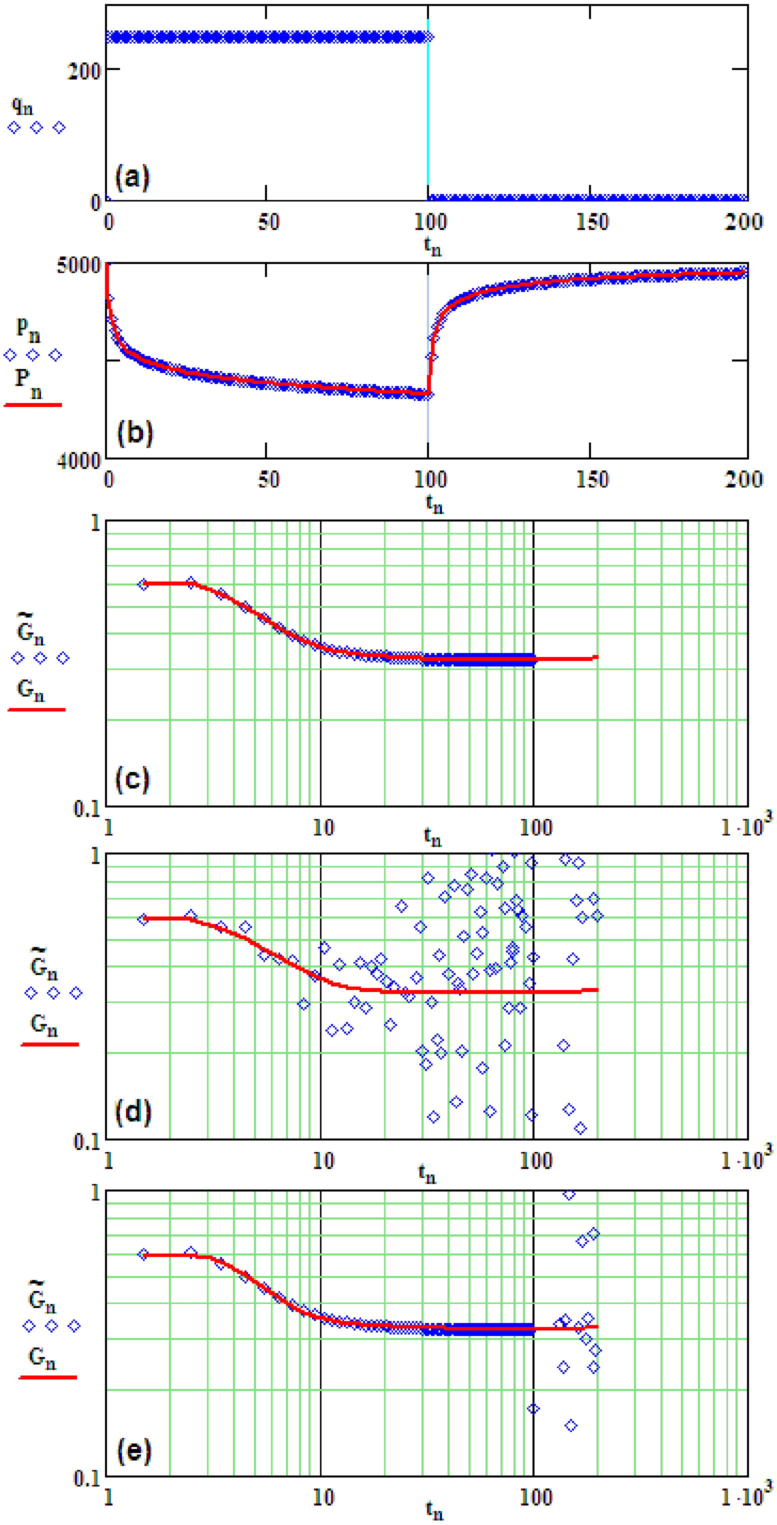

Figure 3 - Synthetic case I.

(a) q -flow rate; (b) p -pressure; P -reconstructed pressure; $\tilde{G}$ -Bourdet derivative; G -deconvolved derivative (c) zero noise, all pressure history (d) 5% noise, all pressure history; 5% noise, build-up pressure data only ($t \geq 100$).

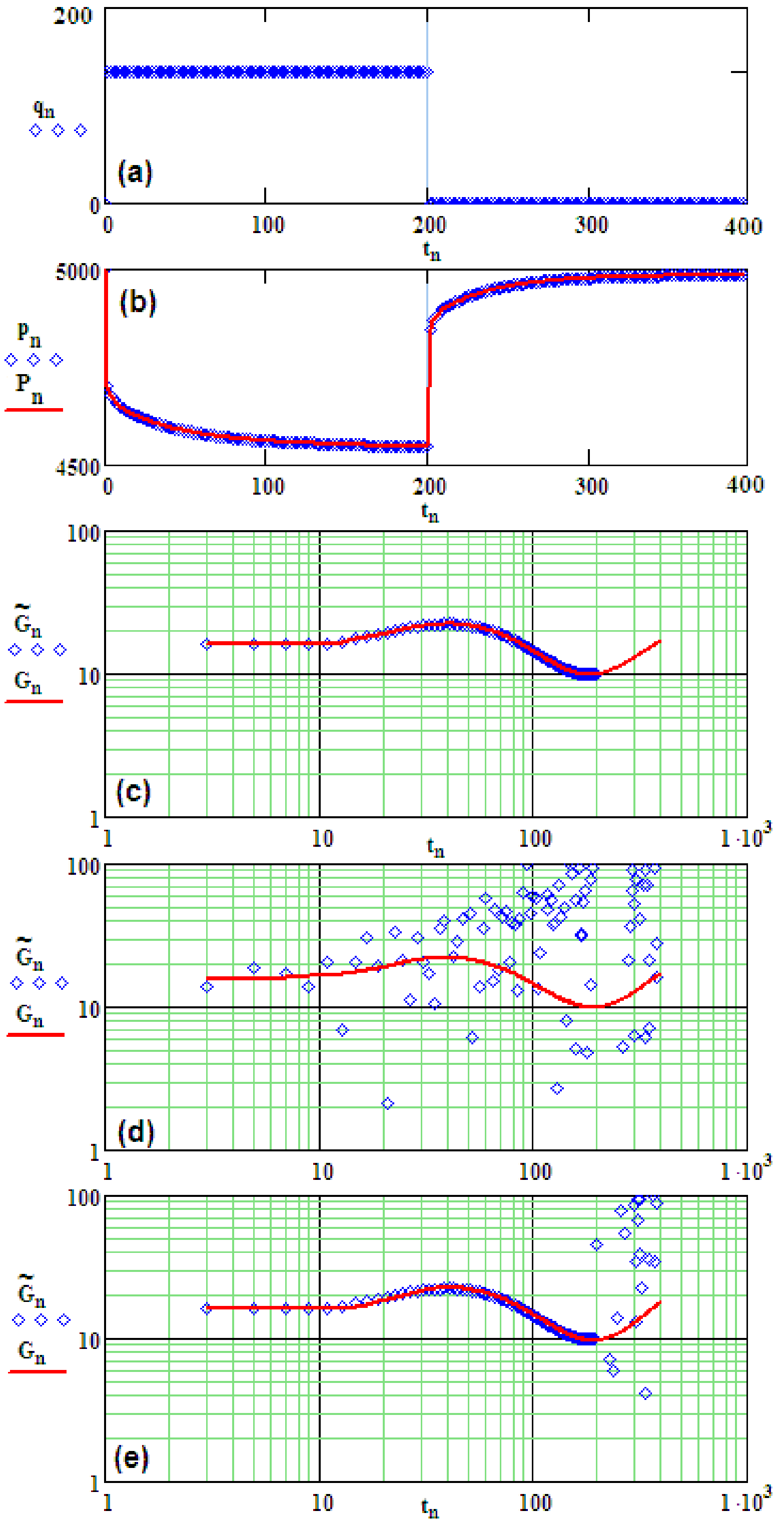

Figure 4 - Synthetic case II.

(a) q -flow rate; (b) p -pressure; P -reconstructed pressure; $\tilde{G}$ -Bourdet derivative; G -deconvolved derivative (c) zero noise, all pressure history (d) 5% noise, all pressure history; 5% noise, build-up pressure data only ($t \geq 200$).

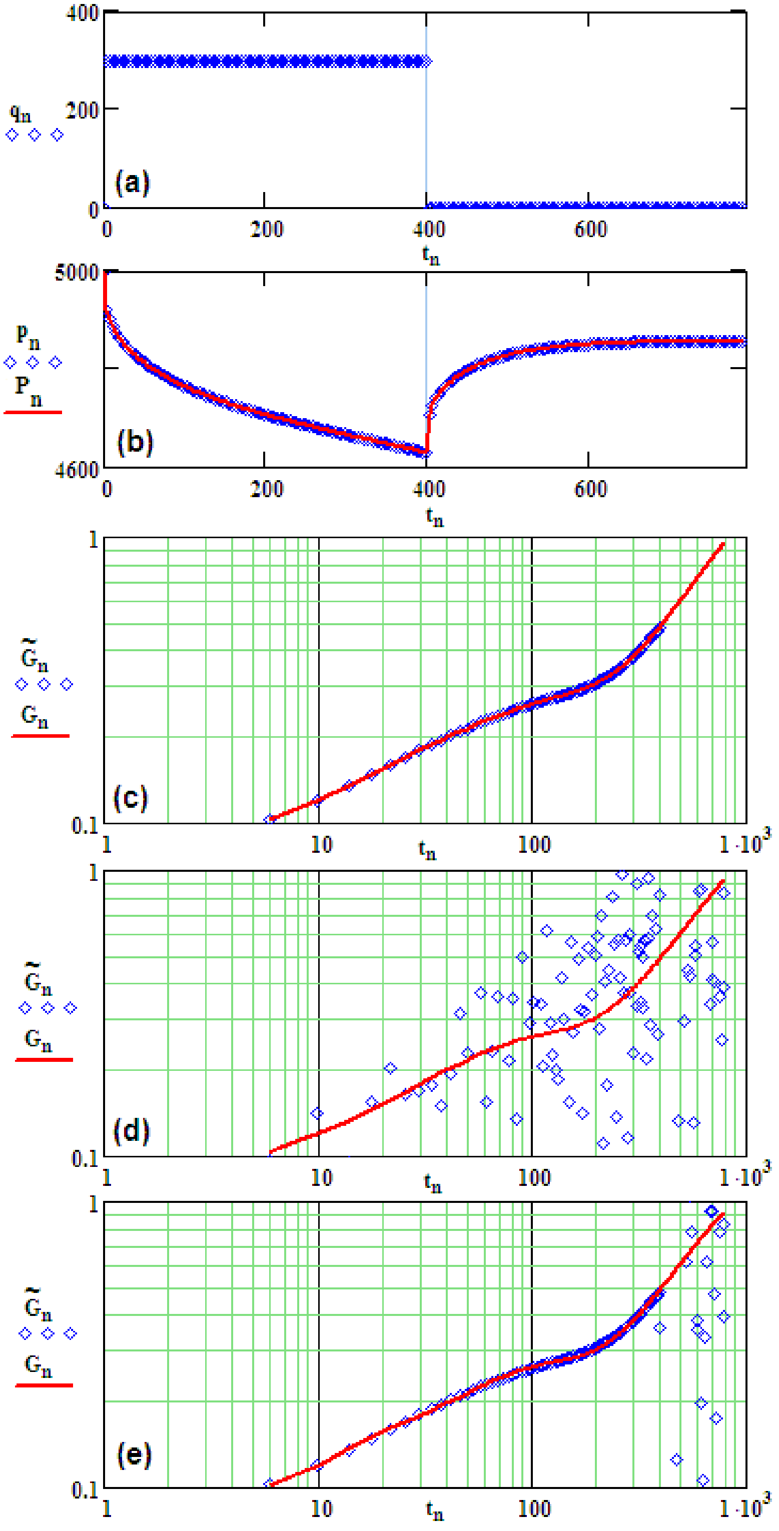

Figure 5 - Synthetic case III.

(a) q -flow rate; (b) p -pressure; P -reconstructed pressure; $\tilde{G}$ -Bourdet derivative; G -deconvolved derivative (c) zero noise, all pressure history (d) 5% noise, all pressure history; 5% noise, build-up pressure data only ($t \geq 400$).

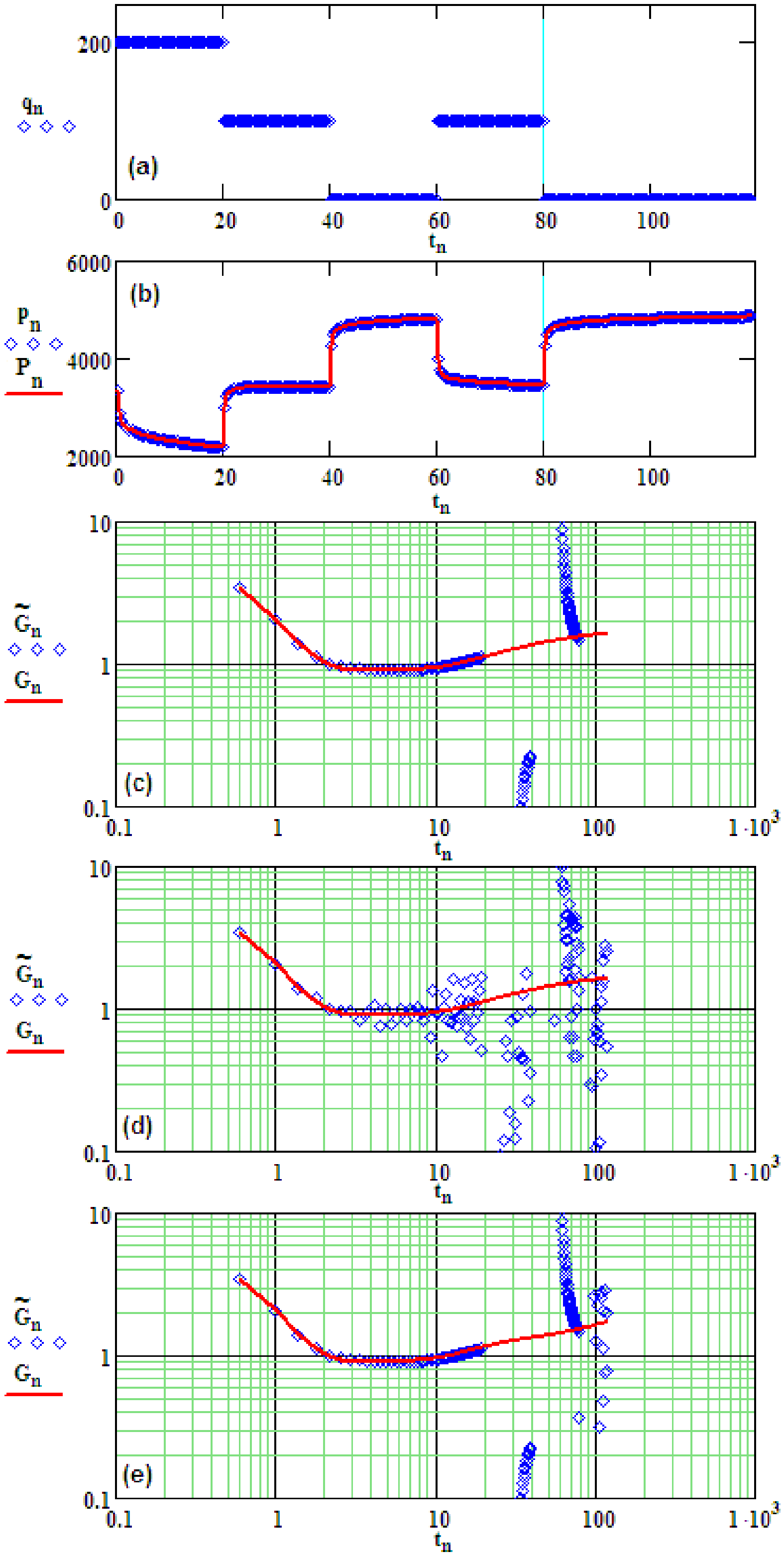

Figure 6 - Synthetic case IV.

(a) q-flow rate; (b) p-pressure; P-reconstructed pressure; $\tilde{G}$-Bourdet derivative; G-deconvolved derivative (c) zero noise, all pressure history (d) 5% noise, all pressure history; 5% noise, build-up pressure data only ($t \ge 80$).

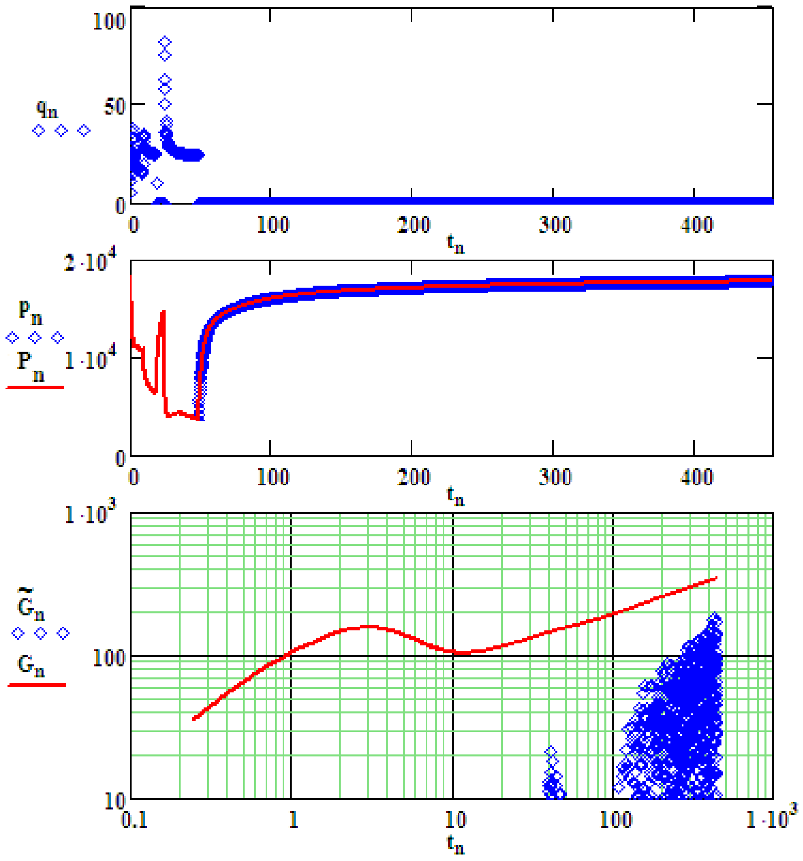

Figure 7 - Field case I.

(a) q -flow rate; (b) p -pressure; P -reconstructed pressure; (c) $\tilde{G}$ -Bourdet derivative; G -deconvolved derivative.

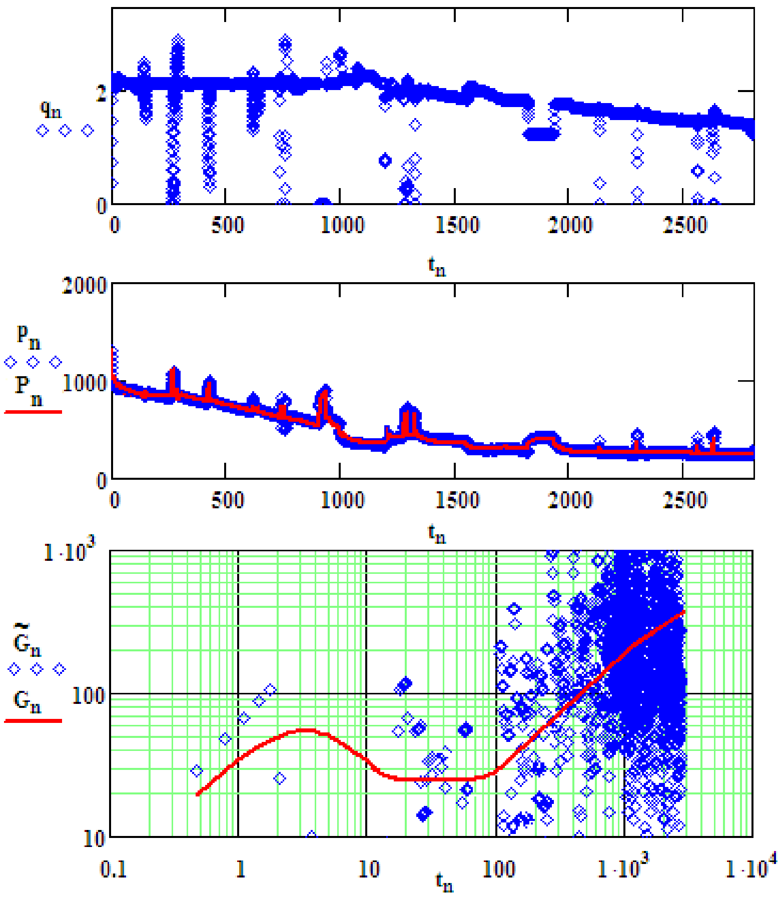

Figure 8 - Field case II.

(a) q -flow rate; (b) p -pressure; P -reconstructed pressure; (c) $\tilde{G}$ -Bourdet derivative; G -deconvolved derivative.