# A TIME-CONTINUOUS FINITE DIFFERENCE APPROACH FOR FLOW AND TRANSPORT SIMULATIONS

by

W.Z. Liu
J.J. Butler, Jr.

Kansas Geological Survey
Open-file Report 91-20

# A TIME-CONTINUOUS FINITE DIFFERENCE APPROACH FOR FLOW AND TRANSPORT SIMULATIONS

W. Z. Liu and J. J. Butler, Jr.
Kansas Geological Survey
1930 Constant Ave.
Lawrence, KS 66047

Prepared for presentation at
The American Geophysical Union
Spring Meeting in Baltimore, Maryland
May 28 -- June 1, 1991

KGS Open File Report #91-20

# A TIME-CONTINUOUS FINITE DIFFERENCE APPROACH FOR FLOW AND TRANSPORT SIMULATIONS

W.Z. Liu and J.J. Butler, Jr. (Both at: The Kansas Geological Survey, 1930 Constant Ave., Lawrence, KS 66047; 913-864-3965)

## I. INTRODUCTION

Approximate numerical methods such as the finite difference (FD) or finite element (FE) approaches have been widely used in groundwater studies for solving problems where analytical solutions are not feasible. These numerical approaches involve both spatial and temporal discretization of the governing equations, with the quality of the approximation dependent on the discretization strategy. In both groundwater flow and transport applications, problems may arise with conventional numerical approaches when a comparison between model output and measured data is desired for the specific times at which the measurements were made. This is especially true when working with data from hydraulic or tracer tests, where the density of the data in time may be quite large.

In order to avoid some of the problems associated with temporal discretization, semi-analytical methods, which combine spatial discretization with a Laplace transform in time, have been employed by a number of workers over the past two decades (e.g., Gurtin, 1965; Chen and

Chen, 1988). The most difficult problem associated with these methods has been the inversion of the Laplace-space solution back into real space. Various methods for inversion by numerical approximation have been developed (e.g., Stehfest, 1970; Crump, 1976; Talbot, 1979), all of which involve the evaluation and summation of the transform space function. Crump (1976) incorporates a summation transformation series into his inversion method in order to improve the rate of convergence of the summation series. Sudicky (1989) uses this method to demonstrate the potential of the approach for contaminant transport applications. However, the computationally intensive nature of the approach makes it of limited usefulness for many applications.

The research described here is directed at the development of a more efficient time-continuous finite difference method for groundwater flow and transport simulations. As noted by Sudicky (1989), De Hoog et al. [1982] proposed a quotient difference algorithm for increasing the rate of convergence of the summation series of Crump (1976). This quotient difference algorithm is employed here to significantly decrease the computations required for simulation of groundwater processes. The computational savings are such that this method may be of use in many practical applications.

## II. THEORY

The underlying idea of this time-continuous finite difference method is the elimination of temporal discretization by using a Laplace transform in time. The spatial discretization scheme is the same as that in a conventional finite difference model. The resultant spatially discretized system of algebraic equations in complex space is solved using complex arithmetic for the matrix inversion. The Laplace space solution is then inverted back into real space using an appropriate numerical inversion scheme. The above procedure will result in a solution that is continuous in time, with the only approximation in the temporal domain being that introduced by the numerical inversion scheme.

In order to employ this approach, Laplace transforms must exist for the governing equation together with all boundary conditions. A simple pulse test in a one dimensional radial flow configuration is employed here to demonstrate the theory. The governing equation is

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\,\frac{\partial s}{\partial r}\right) = \frac{S}{T}\frac{\partial s}{\partial t} \qquad (1)$$

where

s = drawdown, [L];
S = storage coefficient, [dimensionless];

T = transmissivity, [$L^2/T$];
t = time, [T];
r = radial distance [L].

with the initial and boundary conditions defined as

$$s(r, 0) = 0, \quad r < \infty \tag{2}$$

$$s(\infty, t) = 0, \quad t \geq 0 \tag{3}$$

$$2\pi T\left(r\frac{\partial s}{\partial r}\right)_{r \to r_w} = -\sum_{i=1}^{NP} q_i \square_i(t) \tag{4}$$

where

$\square_i(t)$ = box car function = 1, if $t_{1i} \leq t \leq t_{2i}$, where i=1, 2, ..., NP,
= 0, otherwise;
NP = number of pulses;
$t_{1i}$ = starting time for pulse i, [T];
$t_{2i}$ = ending time for pulse i, [T];
$r_w$ = radius of pumping well, [L];
$q_i$ = pumpage for pulse i, [$L^3/T$].

Application of the Laplace transform to equations (1) and (4) in conjunction with (2) results in:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\,\frac{\partial \bar{s}}{\partial r}\right) = \frac{S}{T}\,\bar{s}\,p \qquad (5)$$

$$2\pi T\left(r\,\frac{\partial \bar{s}}{\partial r}\right) = -\sum_{i=1}^{NP} q_i \frac{e^{-t_{1i}p} - e^{-t_{2i}p}}{p} \qquad (6)$$

where

$p$ = Laplace transform variable;
$\bar{s}$ = drawdown in Laplace space.

Application of a central difference scheme to (5) and then incorporation of the boundary conditions given in (6) and (3) produces the following system of equations, written in matrix form,

$$\left([A] + p[B]\right)[\bar{s}] = [C] + \sum_{i=1}^{NP} \frac{e^{-t_{1i}p} - e^{-t_{2i}p}}{p}\, q_i[D] \qquad (7)$$

where A, B, C, and D are matrices of constant coefficients and $\bar{s}$ is the vector of unknown drawdowns. For the sake of conciseness, equation (7) is rewritten in the following form:

$$[G][\bar{s}] = [W] \qquad (8)$$

Equation (8) includes the Laplace transform variable p, for which a value must be given before a solution in Laplace space can be obtained. The effort required to solve (8) can be best explained by comparison with a conventional finite difference model, which employs both spatial and temporal discretization. An equation similar to (8),

$$[G'][s]=[W'] \qquad (9)$$

can be constructed using standard temporal discretization techniques and the same spatial discretization as in (8). The matrices G' and W' are of the same size as W and G. The time required for finding $[\bar{s}]$ in (8) is equivalent to finding [s] in (9) only if p is a real number. When p is a complex number, the amount of computations required for matrix inversion is usually much greater because all the calculations are performed in complex space. Note that, unlike in (9), a number of matrix inversions for different p values ($p = p_k$, k=0, ..., 2N+1) are required in order to compute a solution at a new point in time. Therefore, for the solution of (8), considerable attention must be given both to matrix inversion in complex space and to careful selection of values for $p_k$ and other parameters. Inappropriate parameter selection may result in near singular matrices, which are much more difficult to invert and require large amounts of computer time.

Numerous methods for the approximate inversion of Laplace space solutions have been employed in the groundwater literature. The most common methods are those of Stehfest [1970], Crump [1976], and Talbot [1979]. Both the Stehfest and Talbot methods perform the inversion for one specific time, using a different set of $p_k$ values for different times. The Crump method differs from these approaches in that a single set of $p_k$ solutions can be employed to perform the inversion for a range of times. Since our work requires that solutions be obtained for a range of times, the method of Crump [1976] is chosen to be the basis of the inversion scheme used here.

The Crump method approximates the inversion of a Laplace space function by means of a Fourier series that involves both sine and cosine functions. If the value of s at node j is desired, $s_j$ is found using the following equation:

$$s_j(t) \approx \frac{e^{p_0 t}}{Tmax}\left\{\frac{\bar{s}(p_0)}{2} + \sum_{k=1}^{2N+1}\left[RE(\bar{s}_j(p_k))\cos\left(\frac{k\pi t}{Tmax}\right) - IM(\bar{s}_j(p_k))\sin\left(\frac{k\pi t}{Tmax}\right)\right]\right\} \qquad (10)$$

where

$\bar{s}_j(p_k)$ = solution from (8) at node j for $p = p_k$;

2Tmax = the period of the Fourier series approximating the inverse function on the interval [0, 2Tmax];

RE = real part of $\bar{s}$;

IM = imaginary part of $\bar{s}$;

$p_k = p_0 + ik\pi/Tmax$;

$p_0 = \mu - \ln(Er/2Tmax)$, the real part of $p_k$;

Er = relative error;

$\mu$ = maximum real value of all the singularity points of the function in Laplace space;

$i = (-1)^{1/2}$.

Equation (10) shows that the time variable t appears only in the sine, cosine and exponential functions. Since $p_k$ is independent of time, we can perform the inversion over a range of times based on one set of solutions for one specific Tmax. The solution is thus time continuous because once a set of $\bar{s}_j$ values is calculated, (10) will give the desired result at any time within the time range of [0, 2Tmax].

## III TRANSFORMATION OF THE SUMMATION SERIES

If the summation is performed as in (10), hundreds of terms may be needed to obtain a solution that satisfies a given error criterion. As discussed earlier, the computational effort required for the calculation of each p-space solution is at least equal to that required for one time step in a conventional numerical model. Often a total of 30 to 50 time steps are required to generate a time-drawdown curve using a conventional numerical model. If the number of terms in the summation of (10) is larger than 30, the advantage of using a time-continuous approach for practical applications will be greatly diminished. Therefore, it is necessary to find a way to increase the rate of convergence for the series summation of (10) without decreasing the accuracy of the solution.

Crump [1976] found that a series transformation may be incorporated into (10) to speed up the rate of convergence and, at the same time, reduce the truncation error. The required number of terms in the summation becomes on the order of tens instead of hundreds. Essentially, this approach, known as the epsilon algorithm (EPAL), involves the approximation of the summation series inside the bracket of (10) by using 2(N+1) partial sums ($S_n$) given by

$$S_n = \sum_{k=0}^{n} a_k, \qquad n = 0, 1, 2, \ldots\ldots, 2N+1 \qquad (11)$$

A recursive equation is defined that allows a sequence of successive approximations to the series summation to be obtained. This sequence provides a much better approximation than the untransformed sequence of partial sums ($S_0$, $S_1$, $S_2$, ..., $S_{2N+1}$). Unfortunately, the recursive equation has a tendency to magnify small errors, which may result in large errors being introduced into the final solution.

An alternative method that eliminates disadvantages inherent in EPAL was presented by De Hoog et al. [1982]. The De Hoog et al. algorithm (HGAL) is a quotient difference algorithm, which has been shown to be superior to EPAL in terms of both accuracy and speed of convergence.

In HGAL, the summation series inside the brackets of (10) may be rewritten as

$$S_{2N} = \sum_{k=0}^{2N} a_k z^k, \qquad \text{where } z = e^{i\pi t/Tmax} \tag{12}$$

This summation can be approximated by

$$S_{2N} = v(z, 2N) = d_0/(1+d_1 z/(1+\ldots+d_{2N} z)) \tag{13}$$

where $d_j$, j=0,..., 2N are called the continued fraction coefficients and are given by

$$d_0=a_0,\ d_{2n-1}=-q_n^{(0)},\ d_{2n}=e_n^{(0)}, \qquad n=1,\ldots,N. \tag{14}$$

With $e_0^{(i)}=0$, for i=0, ....., 2N, and $q_1^{(i)}=\frac{a_{i+1}}{a_i}$ for i=0, ....., 2N-1, the q and e coefficients are calculated by the following relations:

For r=1, 2, ..., N: $e_r^{(i)}=q_r^{(i+1)}-q_r^{(i)}+e_{r-1}^{(i+1)}$, i=0, 1, ..., 2N-2r (15)

and for r=2, 3,..., N: $q_r^{(i)}=\frac{q_{r-1}^{(i+1)}\,e_{r-1}^{(i+1)}}{e_{r-1}^{(i)}}$, i=0, 1, ..., 2N-2r-1 (16)

The following recursive equations can be used to calculate (13):

$$\begin{aligned} A_n &= A_{n-1} + d_n z A_{n-2} \\ B_n &= B_{n-1} + d_n z B_{n-2} \end{aligned} \qquad n=1,2,\ldots,2N \tag{17}$$

with $A_{-1}=0$, $B_{-1}=1$, $A_0=d_0$ and $B_0=1$. An approximation of the summation series (11) can now be written as $S_{2N}=A_{2N}/B_{2N}$. A solution for drawdown at node j therefore becomes:

$$s_j(t) \approx \frac{1}{Tmax}\, e^{p_0 t} RE\left(\frac{A_{2N}}{B_{2N}}\right) \tag{18}$$

The superiority of HGAL over EPAL can be clearly demonstrated by comparing these two algorithms with respect to speed of convergence and accuracy. The unit function, 1/p in Laplace space, is used here as an example for such a comparison.

The number of terms required for convergence of EPAL and HGAL for the case of 1/p are displayed in Figure 1. Throughout the time range displayed in Figure 1, HGAL needs fewer terms to converge than does EPAL. At times much smaller than Tmax, Nterm for EPAL is twice or more that of HGAL. Also note that for HGAL there exists a period of time during which the number of terms required for convergence is close to a constant. This behavior is very important for obtaining solutions at different times from one suite of $\bar{s}_j$ solutions.

As shown in Figure 1, when the time at which a solution is required is much smaller than or very close to 2Tmax, the number of terms needed for convergence with HGAL is much less than that for EPAL. Table 1 quantifies the errors that arise for the example of Figure 1 when the number of terms is restricted to 20. The large errors produced by EPAL for times close to 2Tmax is simply due to the fact that not enough terms are being used in the summation to obtain a reasonable solution. Another feature shown in Table 1 is that within the period of t = 5 to 14, the errors produced by both algorithms are about the same. From Figure 1, it can be

**seen that the number of terms needed for convergence is close to or less than 20 for both methods in this interval.**

**The Crump inversion method is based on the acceptance of a relative error (Er). Any solution that has converged to within this relative error will not improve by adding more terms. Convergence must therefore be checked after every new term is added to the summation series in order to avoid unnecessary calculations.**

## IV PARAMETER SELECTION AND THEIR INFLUENCE:

Three parameters, namely $\mu$, Er, and Tmax, need to be selected prior to use of the Crump approach. The definitions following equation (10) define how the real valued parameter $p_0$ relates to $\mu$ and Er. The value of Tmax does not play any role in the selection of $\mu$ and Er as Tmax is used solely for determining the time range applicable for the Crump inversion. Since only $p_0$ is present in the final equation, $\mu$ and Er can be treated as one parameter. Generally, $\mu$ and Er are chosen independently with $\mu$ being set to the maximum real value of all the singularity points of the Laplace space function and Er being set to the maximum acceptable relative error. Note that Er must be small enough to generate a reasonable solution. For most practical groundwater applications, Er $\leq 10^{-5}$ should be adequate. Since the outlined procedure involves matrix inversion, careful attention should be paid to the selection of $p_k$ values. One should avoid selecting $p_k$ values for which the ratio of $p_0$ over ($k\pi$/Tmax) is small, so that the matrix [G] will not be near-singular. If a near-singular matrix is encountered, the value of $\mu$ should be increased.

The selection of Tmax is often the most critical step in determining the amount of computations required for a particular application. Figure 1 clearly shows how the number of terms for convergence is dependent on Tmax. Further numerical experiments indicate that for the range of t = 0.1 to 1.7 Tmax, HGAL should only require around 20 terms. In this case, the computational efficiency may be superior to a conventional numerical

method for applications with a high data density. Note that the inversion can only be done for $t \leq 2Tmax$ because the Fourier transform used to numerically invert the Laplace transform function is approximated over the period of [0, 2Tmax]. Even in the case when all times are less than 2Tmax, it may not be efficient to use a single set of $\bar{s}$ values based on one Tmax if results are desired over a time range of two orders of magnitude or greater. In this case, recalculating the solution $\bar{s}$ based on a smaller Tmax may be a more efficient procedure. For example, if results are required at t=0.01 and 10 for the Laplace space function 1/p, more than one hundred terms must be calculated as shown on Figure 1. Only a total of about 30 terms are required to obtain the same results if two sets of solutions are calculated using two different Tmax's (Tmax=0.01 and Tmax=10). It is therefore recommended that if results are to be calculated over a long range of times, one should divide the time range into different intervals and select different Tmax values for each interval. Consideration should be given to choosing the Tmax values such that the total number of summation terms is kept to a minimum. One can not overemphasize the importance of minimizing the number of summation terms required for convergence, since, for most applications, a matrix inversion in complex space will be required.

# V EXAMPLES

## Example 1: Comparison with Theis solution

The semi-analytical one-dimensional pulse test in section II (equation (7)) can be compared against the analytical solution of Theis [1935] by setting $t_{11}=0.0$ and $t_{21}=\infty$ for a single pulse. The outer radial boundary is arbitrarily placed at 37,364 meters. A total of 40 nodes is employed for the radial discretization scheme, with an observation well being placed at r=19.4 meters. The parameter values used in this example are $\mu=0.0$ and $Er=10^{-6}$. The resulting time (t) versus drawdown (s) curve is shown in Figure 2. Since the time range displayed in Figure 2 is over five orders of magnitude, the drawdown was generated using a series of Tmax values (0.03, 0.5, 8, and 80) in order to minimize the total number of summation terms. The total number of summation terms for this case is 60. The error caused by radial discretization is the major contributor to the differences between solutions. The '*' on Figure 2 denote the solution calculated by a finite difference model with temporal discretization and the same spatial discretization scheme as the HGAL solution. It is clear that the differences between the temporally discrete solution and the HGAL solution are nearly negligible.

## Example 2: Simulation of a Three-Pulse Pumping Test

This example uses Eq. (7) for a three-pulse pumping test with the same radial discretization, observation well location, and aquifer properties as in Example 1. The duration of each pulse is 5 minutes, starting at 0.0, 0.1, and 0.2 days. This example is used to demonstrate the influence of functional form on the number of terms needed for convergence. The bottom two curves depict drawdown calculated from an analytical solution and the HGAL algorithm. The top two curves are Nterm versus time plots for Tmax=0.1 and Tmax= 0.17. By comparing these two Nterm versus time curves with those in Figure 1, it is clear that Nterm will increase if there is a sharp change of direction in the drawdown curve. The net increase in Nterm caused by such a change can be reduced by better selection of Tmax. In this cases, selection of a Tmax in the vicinity of a peak is recommended. Note that as the distance between the observation and pumping wells increases, Nterm will decrease as the peaks of the pulse-induced drawdown flatten and broaden.

# IV SUMMARY AND CONCLUSION

A time-continuous finite difference method for the solution of groundwater flow and transport problems has been presented. This method is based on the use of a Laplace transform in time in conjunction with a finite-difference scheme in space. The resultant Laplace space solutions are inverted to real space using the numerical approximation of Crump (1976). A quotient difference series transformation, developed by De Hoog et al. (1982), is incorporated into the Fourier series method of Crump to significantly increase the rate of convergence. This numerical inversion scheme is efficient and accurate, requiring only a limited number of Laplace-space evaluations. An important advantage of this inversion approach is that solutions in real space can be determined for a range of times based on one set of Laplace-space solutions. The approach outlined here is especially useful for cases when a comparison between simulated and measured data is required, a considerable number of measurements exist, and the forcing function is not continuous in time. Although not emphasized here, Sudicky (1989) outlines further advantages with respect to transport applications and to parallel processing algorithms.

## REFERENCES


Chen, H. T., and C. K. Chen, Hybrid Laplace transform/ finite-element method for two-dimensional heat conduction, *J. Thermophys.*, 2(1), pp. 31-36, 1988.

Crump, K. S., Numerical inversion of Laplace transforms using a Fourier Series approximation, *J. Assoc. Comput. Mach.*, 23(1), pp. 89-96, 1976.

De Hoog, F. R., J. H. Knight, and A.N. Stokes, An improved method for numerical inversion of Laplace transforms, *SIAM J. Sci. Stat. Comput.*, 3(3), pp. 357-366, 1982.

Gurtin, M. E., Variational principles for initial value problems, *Q. App. Math.*, 22, pp. 252-256, 1965.

Stehfest, H., Numerical inversion of Laplace transforms-algorithm 368, *Commun. Assoc. Comput. Mach.*, 13(1), pp. 47-49, 1970.

Sudicky, E. A., The Laplace transform Galerkin technique: A time-continuous finite element theory and application to mass transport in groundwater, *Water Resour. Res.*, 25(8), pp. 1833-1846, 1989.

Talbot, A., The accurate numerical inversion of Laplace transforms, *J. Inst. Math. Appl.*, 23, pp. 97-120, 1979.

## Table 1

**Errors in estimates of f(t)=1, double precision Tmax=10, $\mu$=0.0, Er=$10^{-6}$, and Nterm=20**

| t | HGAL | EPAL |
|---|---|---|
| .13 | .21643E-02 | .27601E+00 |
| .25 | .19513E-03 | .51555E-01 |
| .50 | .31678E-04 | -.12962E+00 |
| 1.00 | -.10125E-05 | .96252E-01 |
| 2.00 | -.99581E-06 | -.50650E-03 |
| 3.00 | -.99992E-06 | -.26321E-04 |
| 4.00 | -.99999E-06 | .22665E-04 |
| 5.00 | -.10000E-05 | .83547E-06 |
| 6.00 | -.10000E-05 | -.11470E-05 |
| 8.00 | -.10000E-05 | -.10046E-05 |
| 10.00 | -.10000E-05 | -.10000E-05 |
| 12.00 | -.10000E-05 | -.10015E-05 |
| 14.00 | -.99999E-06 | .24413E-07 |
| 16.00 | -.10033E-05 | -.14338E-01 |
| 17.00 | -.97323E-06 | -.20154E+04 |
| 18.00 | .13069E-04 | -.61197E+04 |
| 18.50 | -.22659E-03 | .11450E+05 |
| 19.00 | -.65480E-02 | -.23622E+05 |

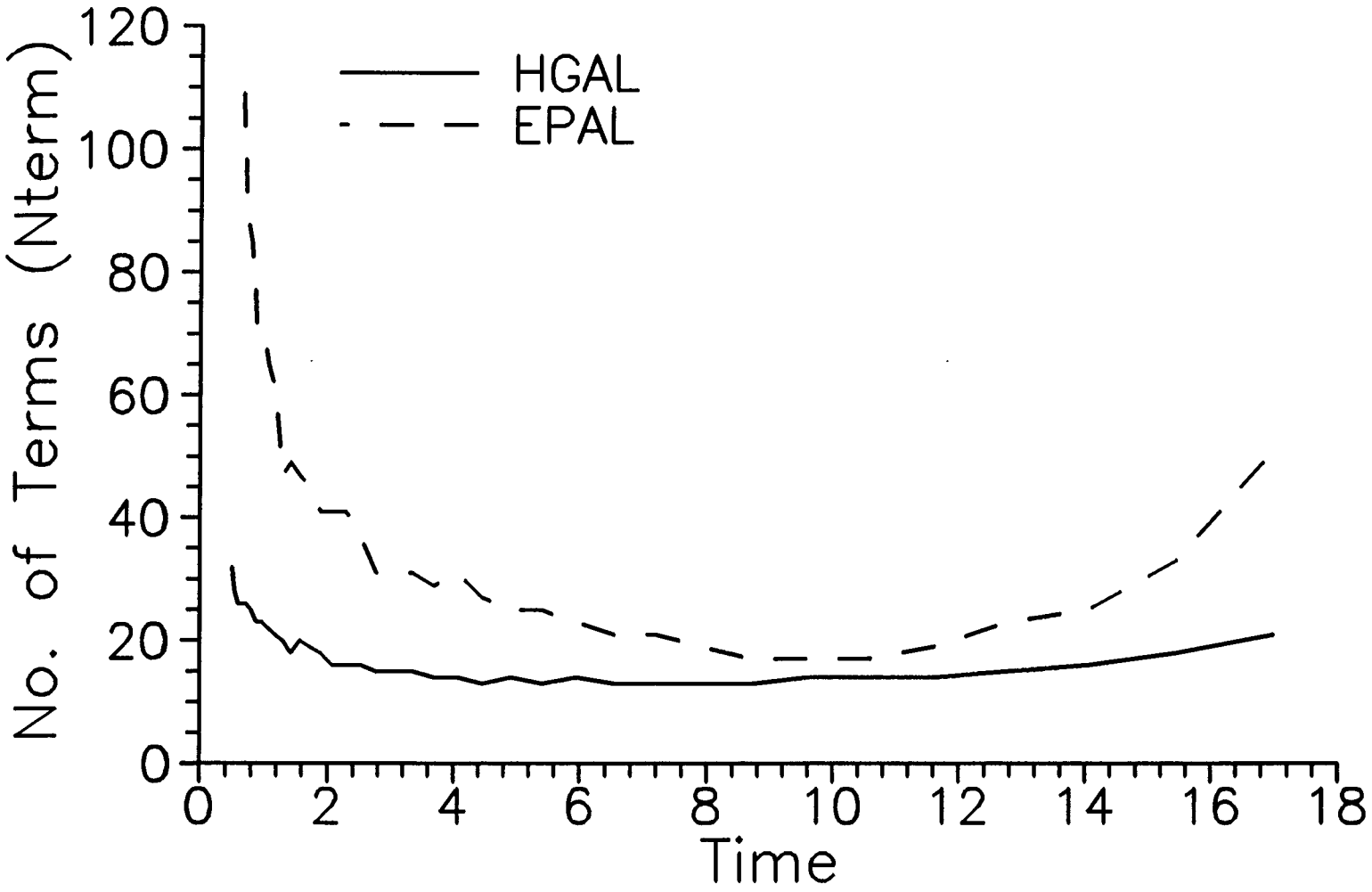


FIGURE 1. Dependence of Nterm on time
Unit function 1/p: Er=$10^{-6}$, $\mu$=0, Tmax=10

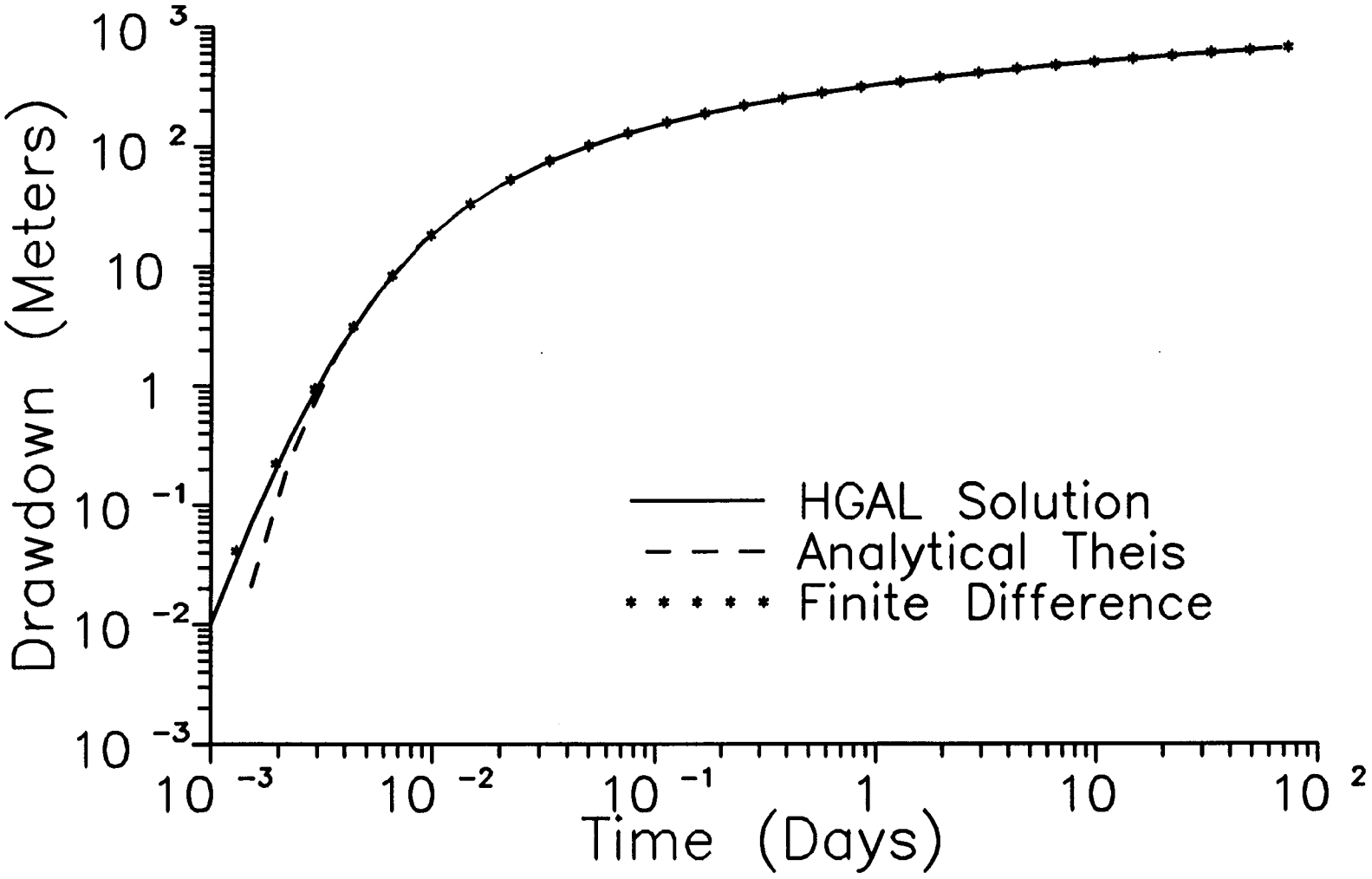


FIGURE 2: Drawdown versus time plot
$Q=1000 m^3/d$, $r=19.4m$, $T=1 m^2/d$, $S=10^{-4}$
$Er=10^{-6}$, $\mu=0$, Tmax=(0.03, 0.5, 8, and 80)

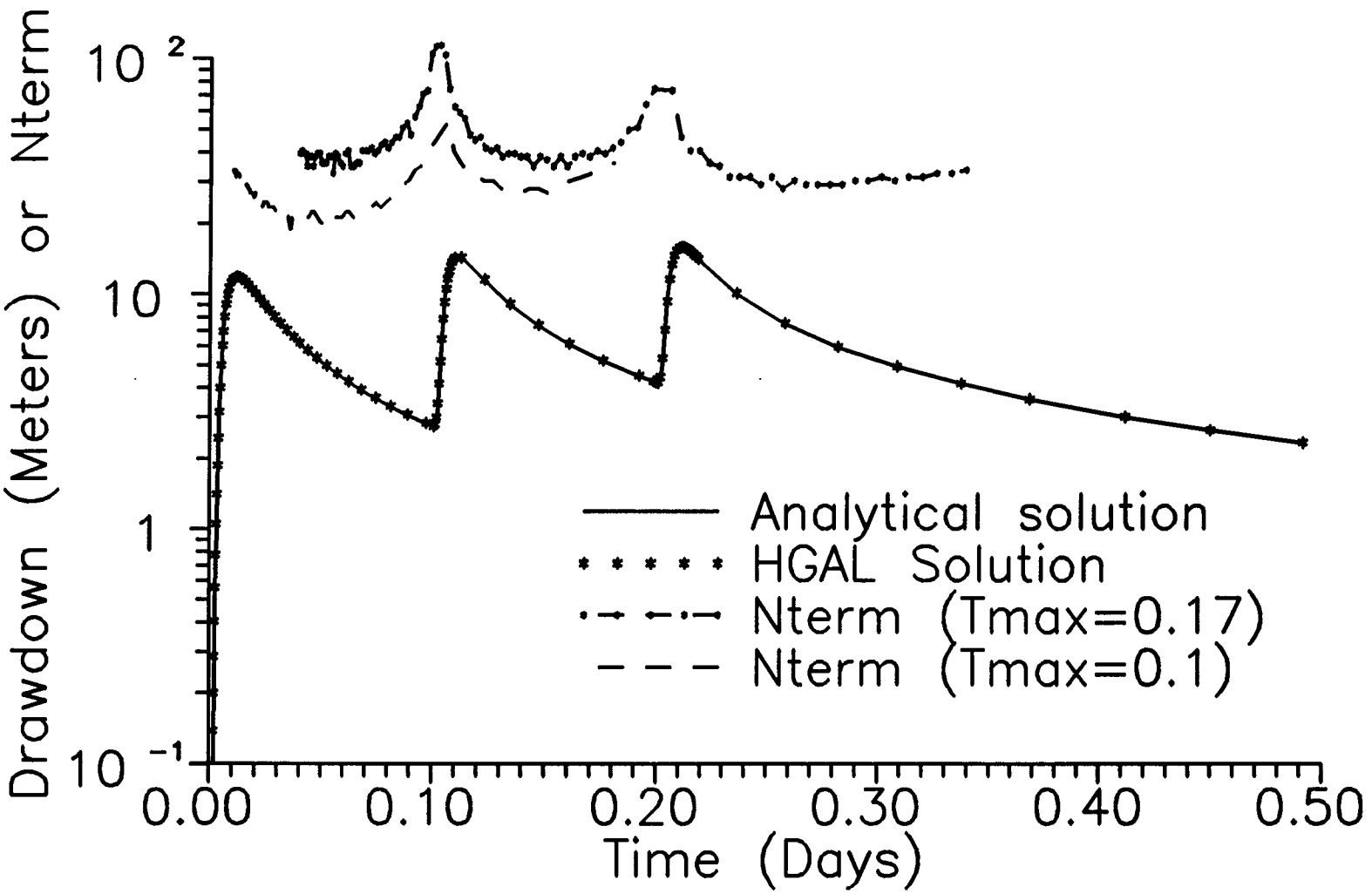


FIGURE 3: Drawdown or Nterm versus time plot Three Pulses (duration of 5 minutes each) starting at 0.0, 0.1, and 0.2 days, respectively; $Er=10^{-6}$, $\mu=0$.