Determination of the Three-Dimensional Anisotropy Tensor in an Unconfined Aquifer Using Depth-Specific Drawdown Data

by

Christopher Holmes

submitted in partial fulfillment of the requirements

for the degree of

Master of Science in Hydrology

New Mexico Institute of Mining and Technology

Summer 1993

DEDICATION

Dedicated to the cannon fodder of science (i.e. graduate students)

"Everything should be made as simple as possible, but no simpler."

--Albert Einstein

ACKNOWLEDGEMENTS

I thank my advisor, Chia-Shyun Chen, for his advice and perseverance during M.S. program at New Mexico Tech. The work reported here was supported financially by USGS(14-08-001-G1744) and WRRI(1423640 and 4-23652). I also thank the graduate students, David Chace, Jingfang He, Weidong Li, and Jinzhong Liu, who worked with me on this project for their help. The work of Irl Downes; our driller, Miles Olinger who helped bring the data-logger on line, and Ted Stans; Manager of the Sevilleta National Wildlife Refuge, is appreciated.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

During the fall of 1992, an aquifer pumping test was conducted in an unconfined alluvial aquifer to determine the aquifer's three-dimensional anisotropy tensor. A partially penetrating well was pumped and depth-specific drawdown data were measured at three different depths of three different locations. A procedure based on the Laplace-Hankel domain counterpart of an appropriate three-dimensional unconfined well hydraulics solution was developed for analyzing the three-dimensional depth-specific drawdown data and estimating the three-dimensional anisotropy tensor. In the Laplace-Hankel domain, the three-dimensional unconfined well hydraulics solution separates into three terms; the baseline condition representing the Theis solution, the water table effect, and the partial penetration effect. It is shown that at large-times the water table effect is constant for all depths and distances in the vicinity of the pumping well. The partial penetration effect reaches steady state at large-times but its magnitude varies with depth and distance from the pumping well. Thus, for an unconfined aquifer subject to a partially penetrating pumping well, the large-time drawdown histories for different depths of the same planar location are parallel Theis curves. The depth-specific drawdown data recorded during the pumping test supports this theoretical conclusion. Using this large-time property, the horizontal hydraulic conductivity of the aquifer was estimated from the slope of the parallel drawdown histories. A system of three simultaneous equations was solved to give the planar anisotropy tensor. The vertical hydraulic conductivity of the aquifer was estimated from the vertical distance between the parallel drawdown histories measured at different depths of the same planar location.

INTRODUCTION

During the fall of 1992, an aquifer pumping test was conducted in an unconfined alluvial aquifer at a research site north of Socorro, New Mexico. Depth-specific drawdown data were collected at different depths of different locations to estimate the aquifer's three-dimensional anisotropy tensor for the study of a three-dimensional solute transport problem. The three-dimensional solute problem is not elaborated upon here; its details can be found in Chen et. al. (1993). It should be noted that the storage coefficient and specific yield are not of interest in studying the solute problem and thus are not estimated. The purpose of this paper is to demonstrate the method developed for determining the three-dimensional anisotropy tensor using depth-specific drawdown data.

The site where the pumping test was conducted is located on the flood plain of the Rio Salado within the boundaries of the Sevilleta National Wildlife Refuge approximately 32 kilometers north of Socorro, New Mexico as shown in Figure 1. In the region of the research site the Rio Salado, an ephemeral tributary of the Rio Grande, is a dry channel on the average of 320 days per year. The Sevilleta aquifer is unconfined with a shallow water table located approximately three meters below the ground surface. Split-spoon soil samples obtained during the coarse of this research suggest that the aquifer is relatively homogeneous. The aquifer consists of Holocene Rio Salado alluvium overlying Pleistocene axial stream deposits of the Sierra Ladrones formation. The Rio Salado alluvium consists of interbedded sand, gravel, and silt. The axial stream deposits also consist of interbedded sand and silt with occasional gravel and clay layers. Although the exact thickness of the aquifer is unknown, the drilling experiences suggest that the

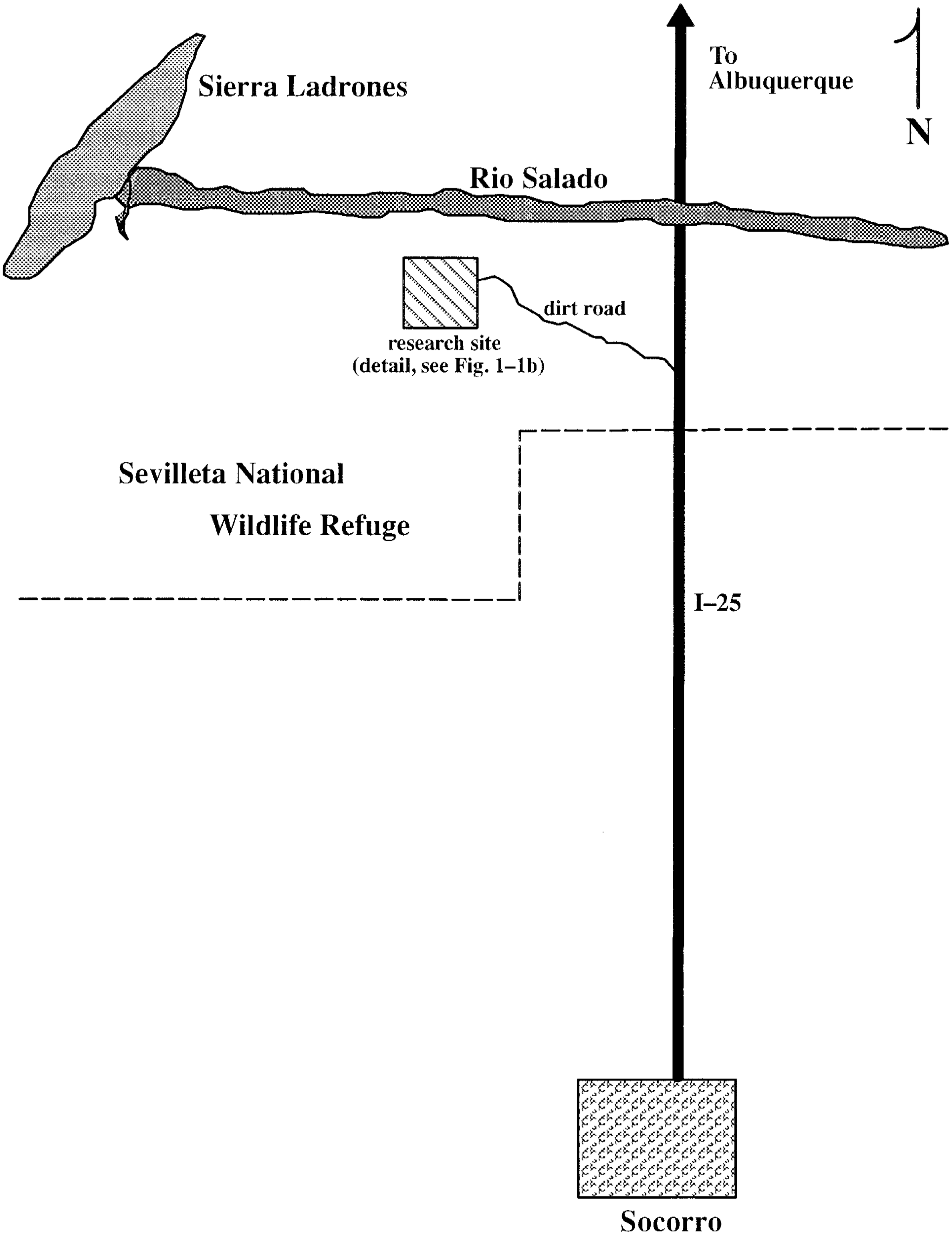
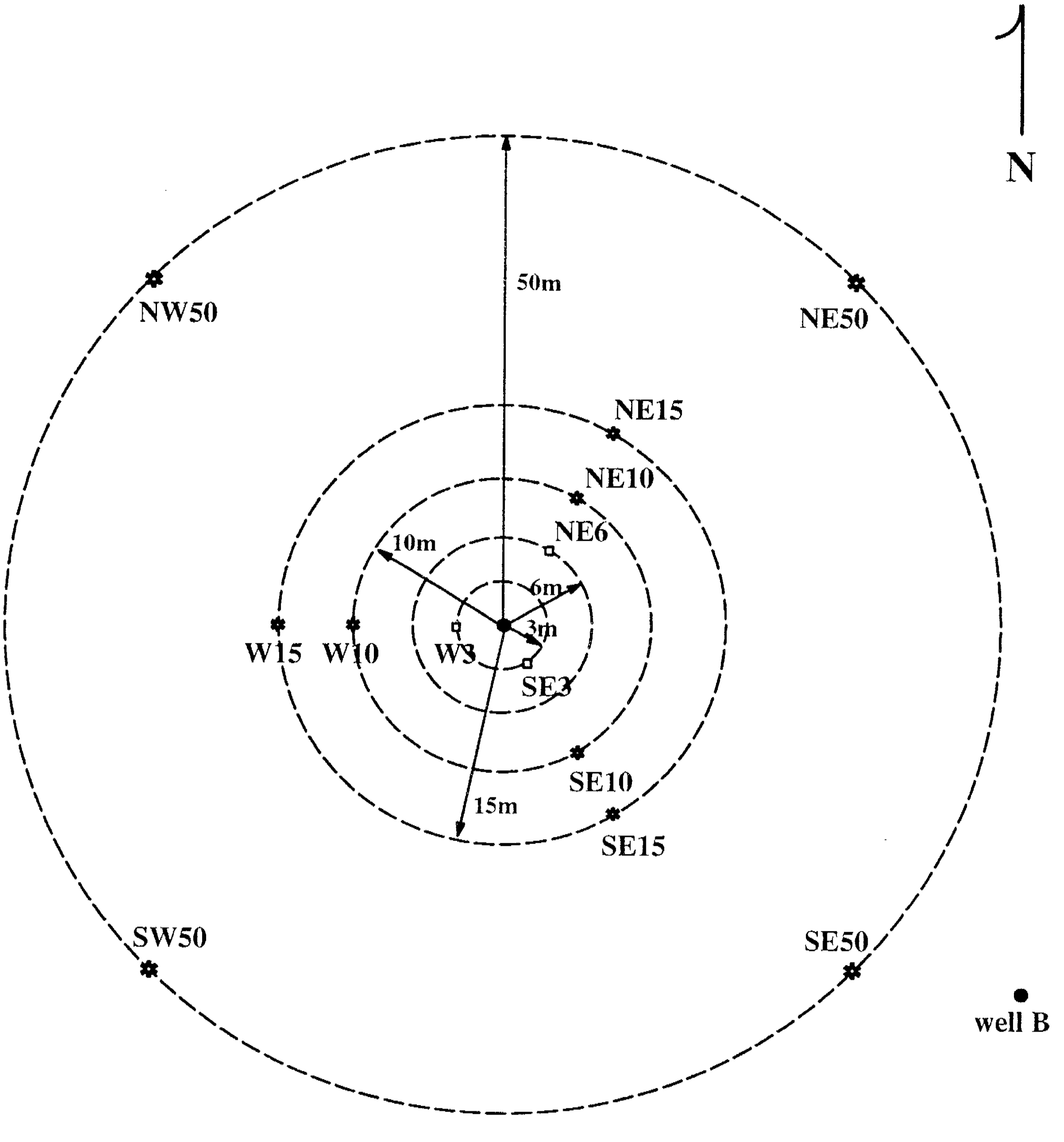

To Albuquerque
N
Sierra Ladrones
Rio Salado
dirt road
research site (detail, see Fig. 1–1b)
Sevilleta National Wildlife Refuge
I–25
Socorro

Figure 1a. Research Site Inside the Sevilleta National Wildlife Refuge North of Socorro, New Mexico.



● – 6–inch well A, approximately 60 meters from 6–inch well B

✱ – 2–inch observation well

□ – functional multilevel samplers/piezometers (MLSP)

Figure 1b. Locations of Observation Wells and MLSP's at the Sevilleta Research Site.

thickness is more than 24.38 meters.

A total of two 15.24 centimeter wells, three multilevel samplers/piezometers (MLSP), and ten 5.08 centimeter observation wells exists at the research site. The locations and configuration of these wells are shown in Figure 1b. Both Well A and Well B are constructed from 15.24 centimeter PVC pipe with 0.8 millimeter machine cut slots in the screened interval. Well A is cased from the ground surface to 6.1 meters and screened from 6.1 to 25.91 meters. Well B is of similar design but is only 12.19 meters in depth. Well A is at the center of the well field consisting of three MLSP's and ten 5.08 centimeter observation wells. The MLSP's and observation wells were installed on three rays in the directions of west (W), northeast (NE), and southeast (SE) on four concentric circles which surround Well A with radii of 3, 6, 10, and 15 meters. Each of the MLSP's and observation wells is designated by its direction and distance measured with respect to Well A. For example, SE3 represents MLSP located 3 meters from Well A in the SE direction. The MLSP's provide depth-specific drawdown data for three-dimensional analysis. As shown in Figure 2, one MLSP includes a cluster of eleven 0.64 centimeter outside diameter polyethylene tubes for groundwater sampling and seven 1.27 centimeter outside diameter PVC standing pipes connected to 0.95 centimeter outside diameter flexible tubing for drawdown measurement. Each standing pipe is 6.1 meters long and connected to a 0.95 centimeter outside diameter flexible tube of which the length depends on the water intake depth. The water intake was made of a 1.27 centimeter outside diameter PVC pipe of length 7.62 centimeters. Holes were drilled and a nylon mesh put on this section. One MLSP allows for the collection of groundwater

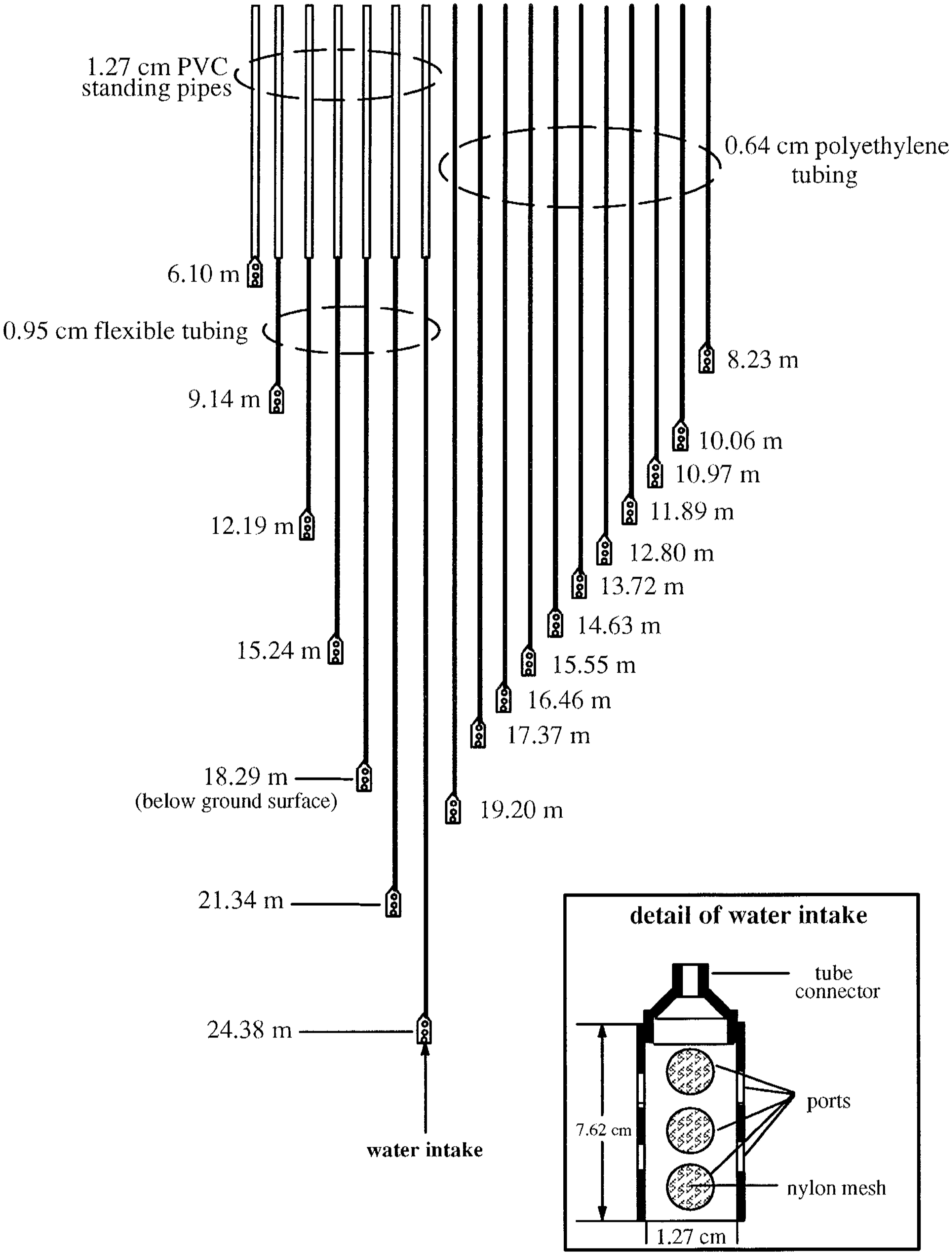


Figure 2. Design of the Multilevel Samplers/Piezometers (MLSP).

samples from eleven different depths and drawdown data at seven different depths. Little vertical averaging can occur inside the small water intakes, and data yielded by the MLSP are depth-specific. The ten observation wells are constructed of 5.08 centimeter with PVC pipe, with 18.29 meters of 0.8 millimeter machine slotted screened interval. The depth to the screened interval varies from one well to the next. These observation wells provide vertically averaged drawdown data which are not used for the current study. Observation wells NW50, NE50, SE50, and SW50 were primarily used to monitor the regional groundwater flow field, which was relatively uniform from NW to SE with a small hydraulic gradient of approximately 10^{-3} meters per meter, provided no significant rainfall events occur.

Drawdown data collected showed an interesting condition. When packers were placed inside Well A at about 13 to 14 meters below the ground surface, drawdown distributions measured at depths below 14 meters were negligibly small relative to those measured above 14 meters, provided that the pumping took place from above the packers. If pumping took place below the packers placed at this depth inside Well A, drawdown distributions measured at depths above 14 meters were negligibly small relative to those measured below 14 meters. This observation suggests that a low-permeability layer exists at approximately 13 to 14 meters below the ground surface in the vicinity of Well A. The existence and thickness of this low-permeability layer were not discernable from the soil samples, possibly due to the fact that the soil sampling missed the layer.

In order to simplify the groundwater flow field in the vicinity of Well A it was decided that a set of inflatable packers would be placed in Well A between 12.19 and 14.63 meters below the ground surface during any pumping tests. Therefore, the Sevilleta aquifer is effectively separated into an upper and lower stratum by a proposed low permeability layer located in between 12 and 14 meters below the ground surface. The conceptualization of the hydrogeologic conditions in the vicinity of Well A is illustrated in Figure 3. Before pumping starts, the water table which was 2.57 meters below the ground surface is assumed to be horizontal. Because the top of the well screen is always below the water table before and during any pumping tests, Well A is partially penetrating. It is assumed that drawdown due to pumping is small compared to the initial saturated thickness, b, between the water table and the low-permeability layer. Depending on the actual depth where the low-permeability layer is located, b has a range from 9.43 to 11.43 meters. An approximation of 10 meters was used for data analysis. Measured from the initial position of the water table, to the top of the well screen, the solid well casing, d, has a length of 3.53 meters. The vertical distance from the initial position of the water table to the bottom of the pumping interval (i.e. the top of the packers in this case), *l*, is 9.63 meters. The vertical distance from the initial position of the water table to the observation point where drawdown is measured (i.e. the water intakes of the MLSP's) is z. As explained earlier, drawdown is negligible in the lower stratum and is only measurable at the upper three water intakes of each MLSP.

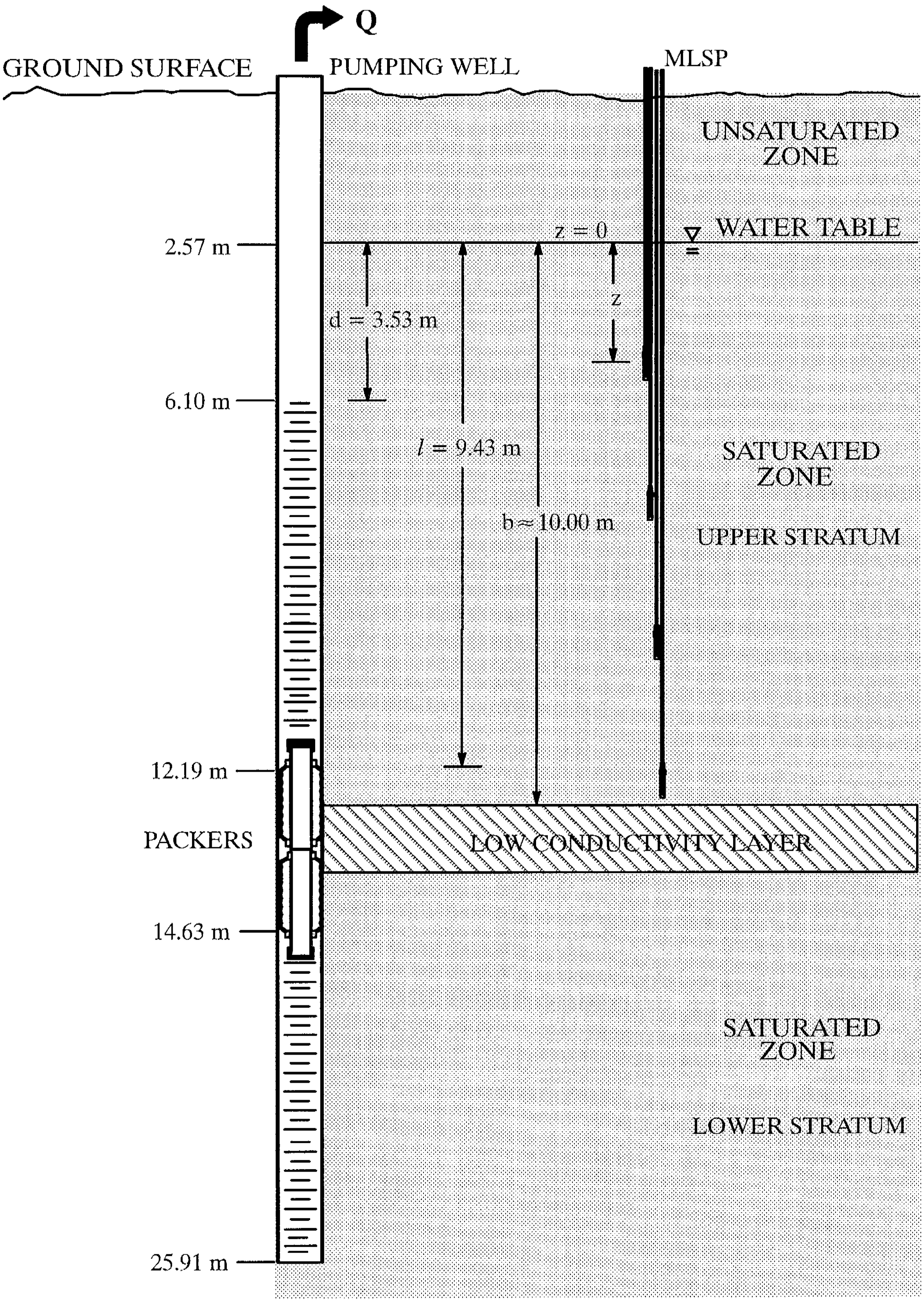


Figure 3. Conceptualization of Hydrogeologic Conditions in the Vicintiy of Well A.

The pumping test was run for about 32.5 hours. Depth-specific drawdown data were taken from three different depths of SE3, W3, and NE6. The depth-specific drawdown data taken at SE3, W3, and NE6 are shown in Figures 4, 5, and 6, respectively. The drawdowns started to drop at around $7\text{x}10^4$ seconds. This drop was likely caused by a pumping rate change. It can be seen very clearly in Figures 4 through 6 that the large-time drawdown data at different depths for a fixed MLSP become parallel straight lines on the semilog plots. This feature forms the backbone of the method developed to estimate the planar anisotropy for the saturated thickness above 14 m at the Sevilleta research site.

Well hydraulics theories are needed to analyze the drawdown data. Unconfined well hydraulics theories have been developed by various investigators. Boulton (1954, 1963) developed analytical solutions for unconfined well hydraulics where an empirical constant called the delay index was used to account for the delayed yield due to water table decline. Neuman (1972, 1974), however, emphasized the three-dimensional nature of unconfined flow and gave three-dimensional analytical solutions. Recently, Akindunni and Gillham (1992) and Nwankwor et al. (1992) have asserted that the yield due to water table decline is time-dependent as influenced by the slow drainage occurring in the vadose zone and significant vertical flow exists in the unsaturated zone. Several graphical curve-matching methods are available for estimating the storage coefficient, S, the specific yield, S_y, the horizontal hydraulic conductivity, K_r, and the vertical hydraulic conductivity, K_z, (or the delay index) of an unconfined horizontally isotropic aquifer. Prickett (1965), Dagan (1967), Neuman (1975), and Lakshminarayana and Rajagopalan

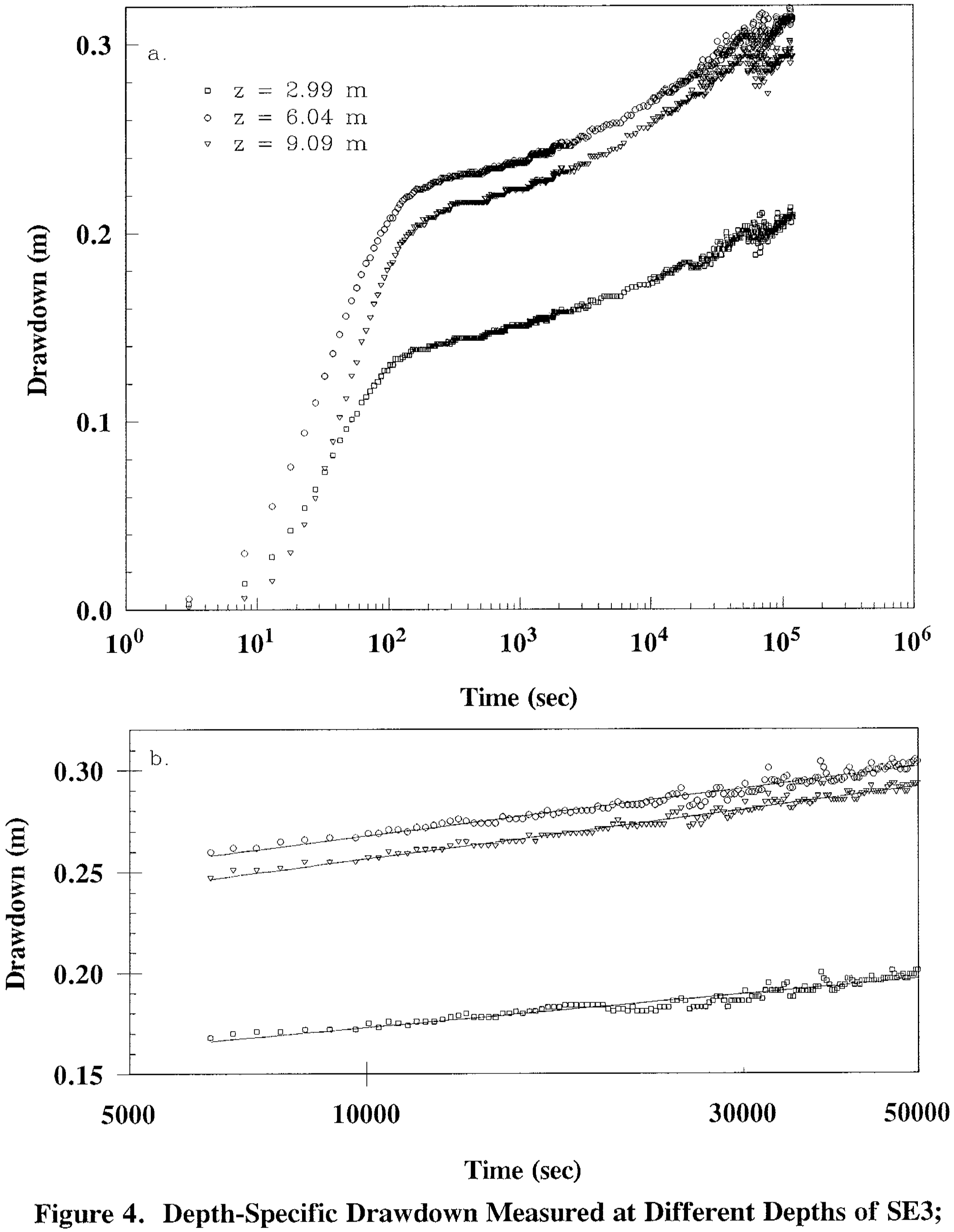


Figure 4. Depth-Specific Drawdown Measured at Different Depths of SE3; $x = 1.60$ m, $y = -2.41$ m, $m_1 = 0.49$ m/s.

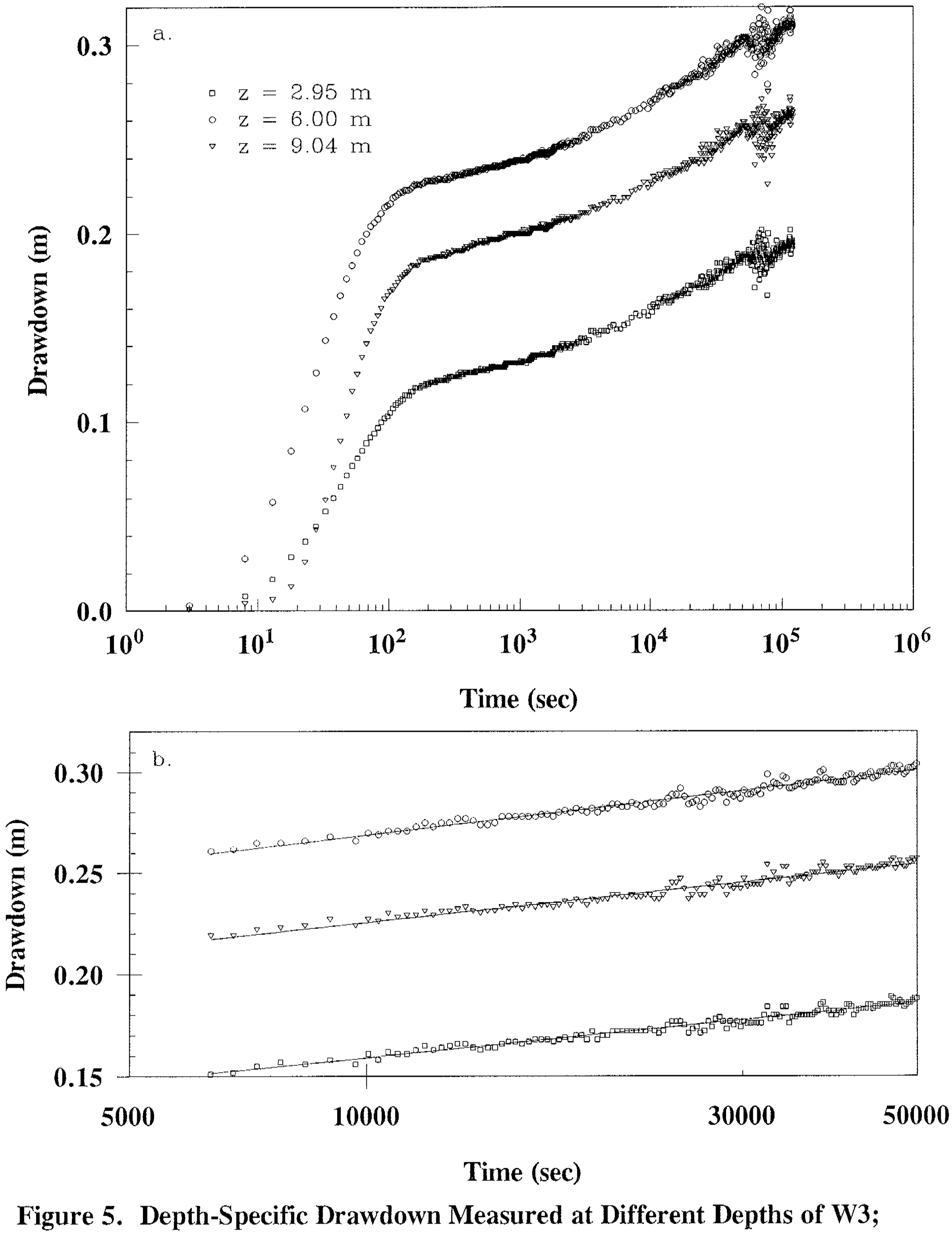


Figure 5. Depth-Specific Drawdown Measured at Different Depths of W3; $x = -3.12$ m, $y = 0.08$ m, $m_2 = 0.042$ m/s.

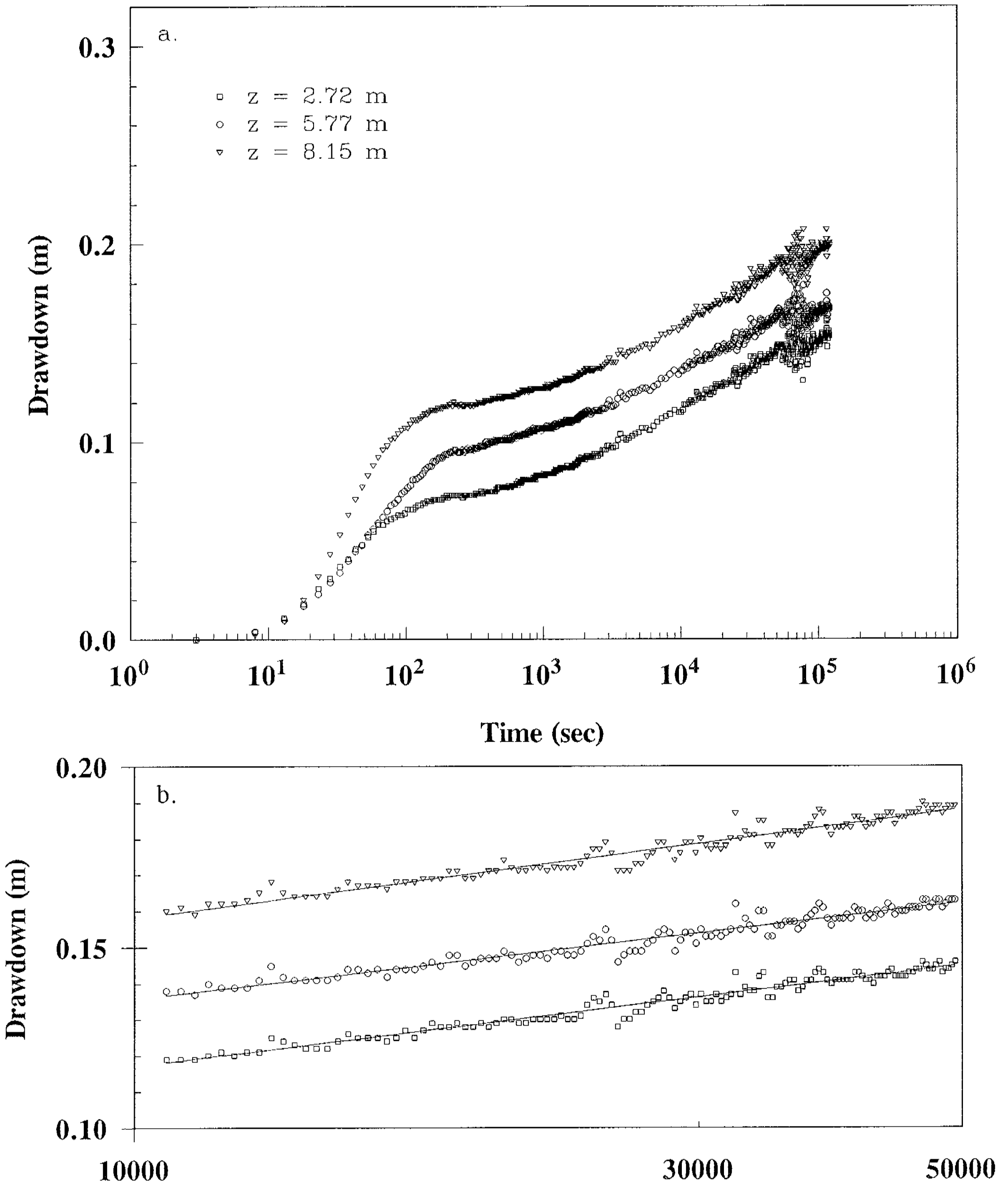


Figure 6. Depth-SpecificDrawdown Measured at Different Depths of NE6;
$x = 3.29$ m, $y = 5.12$ m, $m_3 = 0.041$ m/s.

(1978) developed graphical curve-matching methods based on different unconfined well hydraulics solutions. Because it is difficult to generate appropriate type curves for the full range of the parameters, these authors employed asymptotic solutions which are only valid at small- and large-time to prepare types curves. Various investigators have studied the determination of aquifer anisotropy for other aquifer conditions. Hsieh and Neuman (1985a,b) developed theories and methods to estimate the three-dimensional anisotropy tensor for a fractured formation, where any of the three principal directions is unknown as a priori. Way and McKee (1982) presented a method to estimate the three-dimensional anisotropy of a leaky aquifer. Other aquifer anisotropy studies can be found in Hantush (1964, 1966a,b), Papadopulos (1965), Neuman et al. (1984), Stoner (1981), Miller [1984], Hibsch and Kreft (1980) and others. The linear or nonlinear least-squares fitting method, which does not use graphical type curves, has been frequently used to estimate aquifer parameters (e.g., Johns et al., 1992; Chandler et al., 1981, and others). This method becomes ineffective when the number of unknown parameters is greater than three or four. These aforementioned methods are not applicable to the current study because it deals with three-dimensional anisotropy tensor having four unknowns, the pumping well is partially penetrating, and the aquifer is unconfined.

METHOD DEVELOPMENT

The three-dimensional unconfined well hydraulics solution given by Neuman (1974) is suitable for the situation shown in Figure 3. Although the original solution was given for horizontally isotropic aquifers, it has been adapted for horizontally anisotropic aquifers by using the coordinated transformation given by Papadopulos (1965). Since the vertical direction normally is a principal direction for alluvial aquifers, the anisotropy tensor for this study can be written as

$$\underline{\underline{K}} = \begin{bmatrix} K_{xx} & K_{xy} & 0 \\ K_{xy} & K_{yy} & 0 \\ 0 & 0 & K_z \end{bmatrix} \tag{1}$$

where K_{xx} K_{xy}, and K_{yy} are elements of horizontal anisotropy. In the coordinate transformation, the terms r and K_r are defined as

$$r^2 = \frac{x^2 K_{yy} + y^2 K_{xx} - 2xyK_{xy}^2}{K_r} \tag{2}$$

and

$$K_r = \sqrt{K_{xx}K_{yy} - K_{xy}^2} \tag{3}$$

where x and y are the spacial coordinates of the observation well. When the aquifer is horizontally isotropic (i.e. $K_{xx} = K_{yy}$, $K_{xy} = 0$), r and K_r reduce to

$$r^2 = x^2+y^2$$

and

$$K_r = K_{xx} = K_{yy}$$

The original analytical solution with modification for horizontal anisotropy is in the form of a complicated integral and is not easy to evaluate. Here, the mathematically simpler Laplace-Hankel domain version of the solution is used. Both the Laplace and Hankel transforms are integral transform techniques for solving partial differential equations. In this problem, the Laplace transform is applied to time and p is the transform parameter of t. The Hankel transform is applied to the radial distance and a is the transform parameter of r. Detailed discussions of the Laplace and Hankel transforms can be found in Sneddon (1972). In the Laplace-Hankel domain, the three-dimensional unconfined well hydraulics solution separates into three terms; the baseline condition representing the Theis solution, the water table effect, and the partial penetration effect.

Based on equation A14 of Neuman (1974) and presenting the partially penetrating effects in appropriate form, the Laplace-Hankel domain solution for the depth-specific drawdown for the current study can be written as

$$g(a,z,p) = g_1 - g_2 + g_3 \tag{4}$$

where

$$g_1 = \frac{Q}{2\pi b K_r p} \left(a^2 + \frac{pS}{bK_r}\right)^{-1} \tag{5}$$

$$g_2 = \frac{Q}{2\pi b K_z} \left(F_p F_w\right) \tag{6}$$

$$g_3 = \frac{Q}{\pi^2 K_r (l-d)} \sum_{n=1}^{\infty} F_n \left(a^2 + p\frac{S}{bK_r} + \frac{n^2 \pi^2}{b^2}\,\frac{K_z}{K_r}\right)^{-1} \tag{7}$$

$$F_p = \frac{b}{l-d}\;\frac{\sinh[\eta(b-d)] - \sinh[\eta(b-l)]}{\sinh(\eta b)} \tag{8}$$

$$F_w = \frac{\cosh[\eta(b-z)]}{[\alpha_y\, \eta\, \sinh(\eta b) + p\, \cosh\,(\eta b)]\eta^2} \tag{9}$$

$$F_n = \frac{1}{n} \cos\left(\frac{n\pi z}{b}\right) \left[\sin\left(\frac{n\pi l}{b}\right) - \sin\left(\frac{n\pi d}{b}\right)\right] \tag{10}$$

and

$$\alpha_y = K_z/S_y \quad , \quad \eta^2 = \frac{K_r}{K_z}\ (a^2 + \frac{pS}{bK_r})$$

The Laplace and Hankel inverse of (4) gives the results of interest in the r and t domain. By equation 6.532.4 of Gradshteyn and Ryzhik (1980; p.678), the Hankel inverse of g_1 is found to be

$$H^{-1}\left[\frac{Q}{2\pi bK_r p}(a^2 + \frac{pS}{bK_r})^{-1}\right] = \frac{Q}{2\pi bK_r p}\int_0^\infty \frac{aJ_0(ar)}{a^2 + \frac{pS}{bK_r}}da$$

$$= \frac{Q}{2\pi bK_r p}K_o\ (r\sqrt{\frac{pS}{bK_r}}) \qquad \textbf{(11)}$$

The Laplace inverse of (11), by equation 13.41 of Oberhettinger and Badii (1973; p. 338), gives the Theis solution, or

$$h_1(r,t) = L^{-1}H^{-1}(g_1) = \frac{Q}{4\pi bK_r}W(u) \qquad \textbf{(12)}$$

where W(u) is the exponential integral which can be evaluated with the formulae given by Abramowitz and Stegun (1970; p. 231). The function, g_2, contributes the water table effect to the drawdown because the specific yield, accounting for drainage of water table decline, is only involved in g_2. The Laplace and Hankel inverse of g_2

gives the water table effect on drawdown, which is

$$h_2(r,z,t) = \frac{Q}{2\pi bK_z} L^{-1}H^{-1}(F_pF_w) \qquad \textbf{(13)}$$

where the Hankel inverse is handled numerically through the fast Hankel transform (FHT) technique and the Laplace inverse is calculated with the Stehfest (1970) numerical method. The Stehfest method is easy to use and has been frequently employed to numerically invert many Laplace-domain solutions of groundwater problems (e.g., Moench, 1984b; Chen 1985 and others). The FHT method, however, has been commonly used in geophysics but is not recognized in hydrogeology. Anderson (1979) gave detailed discussions on the FHT method. The FORTRAN subroutine, DHANKL, developed by Anderson (1982) is employed to handle the Hankel inverse. As given in (7), the Hankel transform part of g_3 (i.e., $a^2 + Ps/T_r + n^2\pi^2K_z/(b^2K_r)$) can be exactly inverted using (11). Completion of this leaves only the Laplace transform for the remainder of g_3, therefore, the effects of a partially penetrating pumping well can be determined by

$$h_3(r,z,t) = L^{-1}H^{-1}(g_3) = \frac{Q}{\pi^2K_r(l-d)} \sum_{n=1}^{\infty} F_n \, L^{-1}\left[\frac{K_o(r^2p \; \frac{S}{bK_r} + n^2\pi^2\beta)^{1/2}}{p}\right] \qquad \textbf{(14)}$$

where $\beta = (K_zr^2)/(K_rb^2)$. As shown in Hantush (1961), the analytical inverse of the Laplace transform in (14) is an integral function that is difficult to evaluate. Thus, the Stehfest method is used to numerically calculate the Laplace inverse in (13). The modified Bessel function, $K_0(x)$, is evaluated using the function K0 from the SPECFUN library by Cody (1987). As a result, the depth-specific drawdown in an unconfined aquifer subject to a partially penetrating pumping well condition (i.e., $d > 0$ and/or $l < b$) can be determined by

$$h(r,z,t) = h_1 - h_2 + h_3 \tag{15}$$

where h_1, h_2, and h_3 are determined using (12), (13), and (16), respectively.

In Table 1, dimensionless depth-specific drawdowns computed using the Laplace-Hankel domain calculation are compared to results from DELAY2, the computer program described by Neuman (1975). The maximum difference between drawdowns computed using Laplace-Hankel domain calculation and drawdowns computed using Delay2 is about 0.8 percent. This difference occurs when $(4u)^{-1}$ is 10^{-1} and β is 10^{-2}. If $(4u)^{-1}$ is increased to 10^{1} and β remains 10^{-2}, the maximum difference between drawdowns computed by the two methods is only about 0.2 percent. When $(4u)^{-1}$ remains constant but β is increased to 1, the maximum difference in drawdowns computed by the different methods decreases to about 0.3 percent. In general, the Laplace-Hankel domain calculation method produces relatively accurate results.

Table 1. Comparison of dimensionless drawdown calculated by the Laplace–Hankel method and by DELAY2; the pumping well is partially penetrating , (d/b = 0.35 and l/b = 1.)and drawdown is depth specific.

S/Sy = 10^{-9}						
$(4u)^{-1}$	z/b = 0.3		z/b = 0.6		z/b = 0.9	
	current	DELAY2	current	DELAY2	current	DELAY2
$\beta = 10^{-2}$						
10^{-1}	0.3919 X 10^{-2}	0.3912 X 10^{-2}	0.3836 X 10^{-1}	0.3809 X 10^{-1}	0.3835 X 10^{-1}	0.3808 X 10^{-1}
10^{1}	0.4607	0.4607	0.1591 X 10	0.1589 X 10	0.1607 X 10	0.1604 X 10
10^{2}	0.2522 X 10	0.2521 X 10	0.5639 X 10	0.5643 X 10	0.6375 X 10	0.6378 X 10
10^{5}	0.2542 X 10	0.2541 X 10	0.5674 X 10	0.5678 X 10	0.6417 X 10	0.6421X 10
$\beta = 10^{-1}$						
10^{-1}	0.1317 X 10^{-1}	0.1306 X 10^{-1}	0.3740 X 10^{-1}	0.3757 X 10^{-1}	0.3833 X 10^{-1}	0.3843 X 10^{-1}
10^{1}	0.6491	0.6500	0.1288 X 10	0.1289 X 10	0.1517 X 10	0.1516 X 10
10^{2}	0.1356 X 10	0.1357	0.2549 X 10	0.2553 X 10	0.3059 X 10	0.3063 X 10
10^{5}	0.1356 X 10	0.1357	0.2549 X 10	0.2553 X 10	0.3059 X 10	0.3063 X 10
$\beta = 1$						
10^{-1}	0.1523 X 10^{-1}	0.1522 X 10^{-1}	0.2773 X 10^{-1}	0.2780 X 10^{-1}	0.3394 X 10^{-1}	0.3391 X 10^{-1}
10^{1}	0.2791	0.2795 X 10^{-1}	0.4985	0.4988	0.6089	0.6084
10^{2}	0.2954	0.2957 X 10^{-1}	0.5273	0.5276	0.6442	0.6436
10^{5}	0.2955	0.2957 X 10^{-1}	0.5274	0.5277	0.6443	0.6437

LARGE-TIME APPROXIMATION

According to the actual field data shown in Figures 4b, 5b, and 6b, the large time drawdown histories recorded at different depths of the same planar location are parallel Theis curves. Since the Laplace transform parameter, p, is inversely related to t, the asymptotic nature of drawdown at large time can be understood by evaluating the Laplace-domain counterpart at small p's. For p = 0 as for t approaching infinity, both g_2 and g_3 become independent of p and still remain finite. This indicates that for large times, the water table effects and the partially penetrating well effects can reach the steady state. The Theis solution or h_1, however, never reaches the steady state. Thus at large times, depth-specific drawdown for unconfined aquifers subject to a partially penetrating well is described by

$$h(r,z,t) = \frac{Q}{4\pi b K_r}[W(u)-h_w(r,z)+h_p(r,z)] \tag{16}$$

where $h_w(r,z)$ and $h_p(r,z)$ represent the steady-state water table effects and partially penetrating effects, respectively. Specifically, h_w and h_p are

$$h_w = H^{-1}(\frac{2K_r}{K_z}\acute{F}_p\acute{F}_w) \tag{17}$$

and

$$h_p = \sum_{n=1}^{\infty} \frac{4b}{\pi(l-d)}F_n K_o(n^2\pi^2\beta)^{1/2} \tag{18}$$

where F_P' and F_W' are derived from F_P and F_W by setting p equal to zero. Note that no Laplace inverse exists in (17) and (18) because p has been dropped for large time approximation. According to (16), drawdown at a specific r and z for large times is the Theis solution offset by a constant composed of h_w and h_p. Although h_w is mathematically a function of z, the numerical results indicate that at large times the water table effect reaches a stable condition for all r and z in the vicinity of the pumping well. This means that h_W in (16) is actually constant for a given aquifer condition, as shown in Figure 7a. When z changes, h_p varies as well as shown in Figure 7b, indicating drawdown curves for different z at a fixed r are parallel Theis curves. Each of these curves can be fitted by the Theis solution using a constant K_r and a value different from S for the storage coefficient. This "fitted storage coefficient" actually represents the sum of S, S_y, and the partially penetrating effects, and thus it is more appropriately termed as an effective storage coefficient. This effective storage coefficient changes with depths at a fixed r, since the offset of h_p varies with z. As r increases, h_p diminishes, indicating that the partially penetrating effects vanish at large distances.

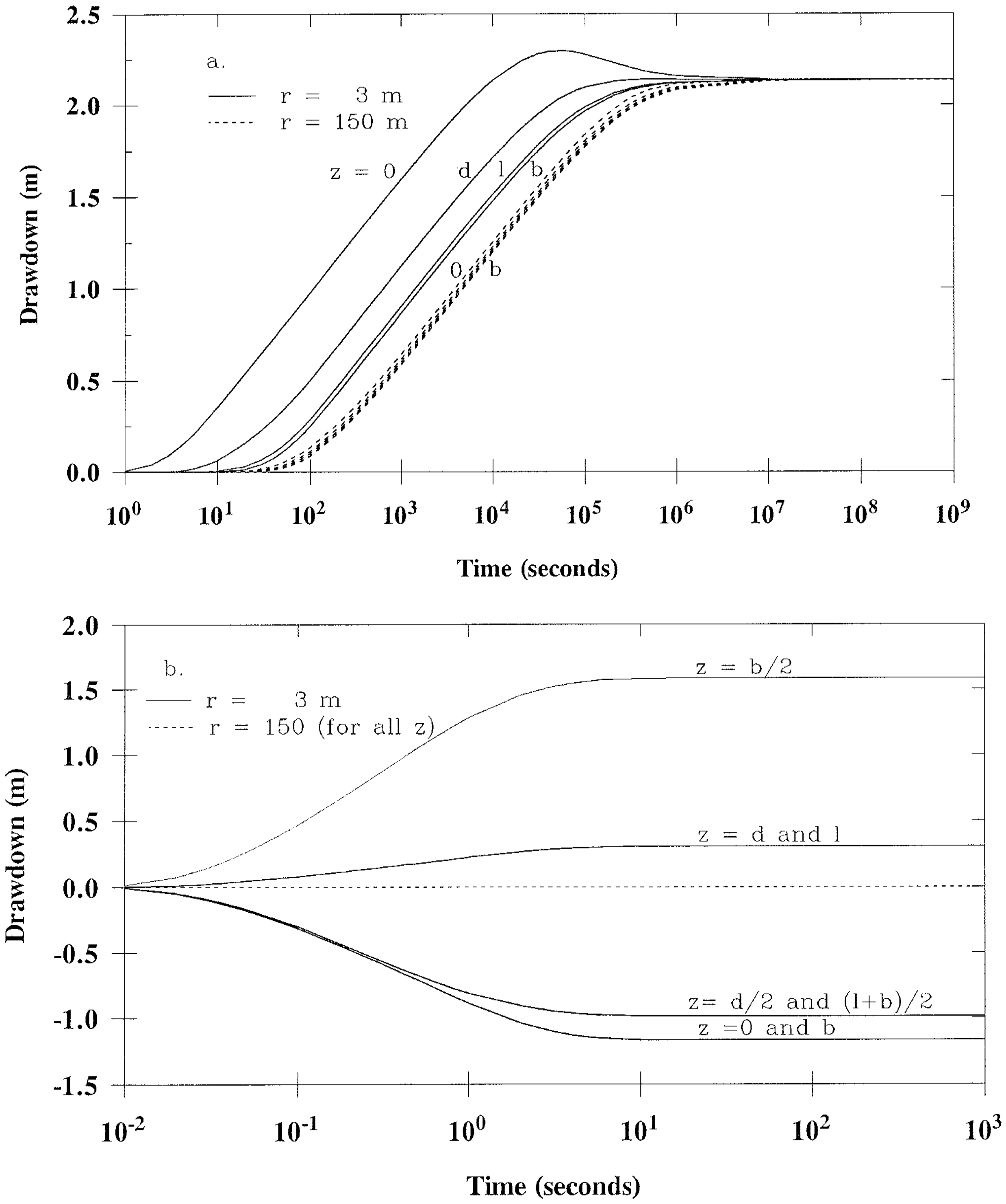


Figure 7. Drawdown at Different Depths due to (a) the Water Table Effect and (b) the Partially Penetration Effect.

ESTIMATION OF AQUIFER ANISOTROPY AND RESULTS

Using the cylindrical coordinates, the three-dimensional groundwater flow field due to pumping can be expressed as

$$\begin{Bmatrix} v_r \\ v_\theta \\ v_z \end{Bmatrix} = - \begin{bmatrix} K_{rr} & K_{r\theta} & 0 \\ K_{r\theta} & K_{\theta\theta} & 0 \\ 0 & 0 & K_z \end{bmatrix} \begin{Bmatrix} \partial h/\partial r \\ \frac{1}{r}\ \partial h/\partial \theta \\ \partial h/\partial z \end{Bmatrix} \tag{19}$$

which can be reduced to

$$v_z = - K_z\ \partial h/\partial z \tag{20}$$

and

$$\begin{Bmatrix} v_r \\ v_\theta \end{Bmatrix} = -\underline{\underline{K}}' \begin{Bmatrix} \partial h/\partial r \\ \frac{1}{r}\partial h/\partial \theta \end{Bmatrix} \tag{21}$$

where

$$\underline{\underline{K}}' = \begin{bmatrix} K_{rr} & K_{r\theta} \\ K_{r\theta} & K_{\theta\theta} \end{bmatrix}$$

$$K_{rr} = K_{xx}\cos^2\theta + K_{xy}\sin 2\theta + K_{yy}\sin^2\theta$$

$$K_{\theta\theta} = K_{xx}\sin^2\theta - K_{xy}\sin 2\theta + K_{yy}\cos^2\theta$$

$$K_{r\theta} = \frac{(K_{yy} - K_{xx})}{2}\sin 2\theta + K_{xy}\cos 2\theta$$

and

$$\theta = \tan^{-1}(y/x)$$

In light of (20) and (21), the horizontal and vertical flow components are separately related to the planar anisotropy characterized by $\underline{\underline{K}}'$ and to the vertical principal hydraulic conductivity of K_z, respectively. This means that $\underline{\underline{K}}'$ and K_z can be determined separately.

As shown earlier, the large-time depth-specific drawdowns measured at SE3, W3, and NE6 are parallel straight lines on the semilog plots. Using the logarithm approximation, each of the straight lines of large-time drawdown can be expressed as

$$h(x,y,z,t) = 0.183\frac{Q}{bK_r}\left[\log(2.25\frac{K_r bt}{r^2 S}) + h_p(r,z) - h_w\right] \tag{22}$$

It should be reemphasized that the water table effect, h_W, is constant in (22) as revealed by Figure 7a. Thus, the source of vertical dependence in (22) is h_p, which can be calculated with (18), provided $\underline{\underline{K}}$ is known. The constant h_W can be determined with (17) if S, S_y, and $\underline{\underline{K}}$ are known. These parameters are to be determined, and h_p and h_w cannot be known as a priori. However, this does not create trouble in estimating $\underline{\underline{K}}'$ using (22) because h_p and h_w are constant for a specific r and a specific z when the large-time conditions are concerned. These two constants can be easily incorporated into the logarithm function appearing in (22) by using the relationship of x = log (10^x). As a result, (22) is rewritten as

$$h(x,y,z,t) = 0.183\frac{Q}{K_r b}\log(2.25\frac{K_r bt}{r^2 S_e}) \tag{23}$$

where S_e is the product of S and the appropriate logarithmic conversion of h_p and h_w. As a lumped parameter combining the effects of storage, water table, and partial penetration, S_e is termed as the effective storage coefficient, which is dependent on r and z. At a fixed r, S_e changes with z and each straight line of large-time drawdown thus, has a different S_e. The "directional horizontal conductivity" evaluated at a fixed r of each MLSP is constant due to the fact that the straight lines of large-time drawdown are parallel and subject to a constant slope.

The method of estimating $\underline{\underline{K}}'$ using the large-time drawdown data is:

(1) Determine the directional horizontal conductivity for each set of the large-time data collected from SE3, W3, and NE6 using the relationship

$$K_i = 0.183\frac{Q}{bm_i} \quad , \quad i = 1,2,3$$

where m_i is the constant slope associated with the parallel straight lines of large-time drawdown at the ith MLSP. Here, the subindex, i, equal to 1,2, and 3 represent SE3, W3, and NE6, respectively. As indicated in Figures 4a, 5a, and 6a, m_1 is 0.049 m/s, m_2 is 0.042 m/s, and m_3 is 0.041 m/s. Using 3.43 x 10^{-3} cubic meters per second for Q, and 10 meters for b, K_1 is determined to be $1.27\text{x}10^{-3}$ meters per second, K_2 is $1.49\text{x}10^{-3}$ meters per second, and K_3 is $1.54\text{x}10^{-3}$ meters per second. These three directional horizontal conductivities are not significantly different in magnitude, implying the aquifer is not significantly anisotropic in the horizontal plane.

(2) Denote t_0 as the time at which the extrapolated straight line of large-time drawdown intercepts the horizontal axis of h being zero. The different values of S_e can be estimated with the associated t_0, at a specific depth. However, this calculation is not necessary since the ratio of t_0/S_e at different depths subject to a constant K_i is constant. Recalling that t_0 is associated with h equal zero, the argument of the logarithm function in (23) must be equal to

unity for t being t_0, that is;

$$2.25\frac{K_i b}{r_i^2}\left(\frac{t_0}{S_e}\right)_z = 1$$

which defines the constant ratio of $(t_0/S_e)_z$ as

$$\left(\frac{t_0}{S_e}\right)_z = \frac{r_i^2}{2.25K_i b} \tag{24}$$

where $(t_0/S_e)_z$ is the ratio of t_0 to S_e for a specific z, and r_i is the actual distance from the ith MLSP to Well A.

(3) Since (23) should be applicable to any of the straight lines of large-time drawdowns, the argument of its logarithm function under the true anisotropic conditions where K_r and r are defined by (2) and (3) must be unity for h being zero. Therefore, the following relationship must be true

$$2.25\frac{K_r b}{r^2}\left(\frac{t_0}{S_e}\right)_z = 1$$

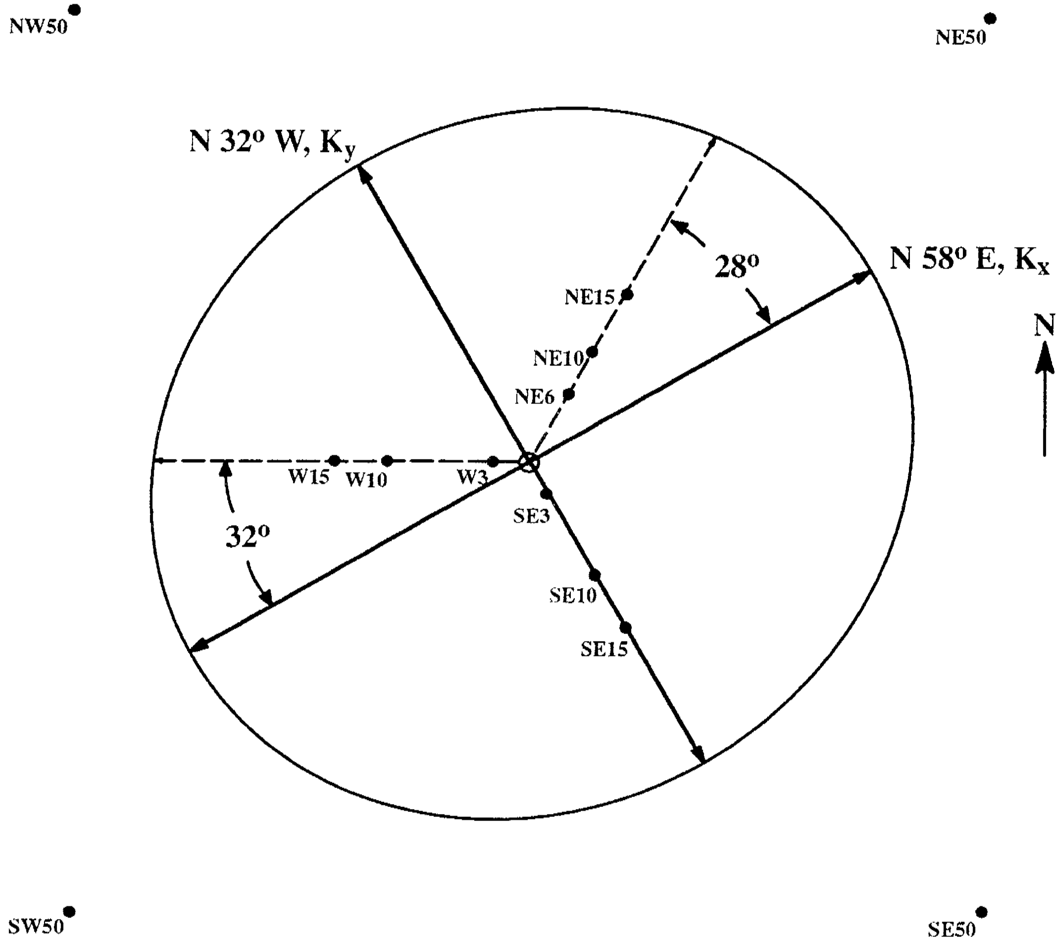
to which introducing (24) yields

$$\frac{K_r}{r^2} = \frac{K_i}{r_i^2} \quad , \quad i = 1,2,3, \tag{25}$$

(4) Three simultaneous equations of three unknowns, K_{xx}, K_{xy}, and K_{yy}, can be derived from (25) after replacing K_r and r by their definitions given in (2) and (3) as

$$\frac{K_{xx}y_i^2 + K_{yy}x_i^2 - 2K_{xy}x_iy_i}{K_{xx}K_{yy} - K_{xy}^2} = \frac{r_i^2}{K_i} \quad , \; i = 1,2,3, \qquad \mathbf{(26)}$$

where (x_i, y_i) refers to the coordinates of the ith MLSP, and $r_i^2 = x_i^2 + y_i^2$. The solution of the three simultaneous equations can be determined using either the linear approximation method or nonlinear methods. The linear approximation method linearizes (26) by using $(K_1 + K_2 + K_3)/3$ for $(K_{xx}K_{yy} - K_{xy}^2)^{1/2}$. The three unknowns of K_{xx}, K_{xy}, and K_{yy} left in the three linearized algebraic equations can be determined without difficulty. The nonlinear method deals with the original form of (26) without replacing the term $(K_{xx} K_{yy} - K_{xy}^2)^{1/2}$ by the arithmetic mean of K_1, K_2, and K_3. The solution of the three simultaneous nonlinear equations can be determined using minimization functions/subroutines available in software packages (e.g., the function "fmins" in MATLAB). Surprisingly, the solutions determined with the linear and nonlinear methods for the current case are almost identical, namely, K_{xx} is 1.52×10^{-3} meters per second, K_{yy} is 1.37×10^{-3} meters per second, and K_{xy} is 1.57×10^{-4} meters per second. Accordingly, the principal hydraulic conductivity of K_x is 1.62×10^{-3} meters per second in the direction of N58°E, and the principal hydraulic conductivity of K_y is 1.27×10^{-3} meters per second in the direction of N32°W. The planar anisotropy ellipse is demonstrated in Figure 8, where the eccentricity is 1.13.



$$K_x = 1.62 \text{ X } 10^{-3} \text{ m/s}$$
$$K_y = 1.27 \text{ X } 10^{-3} \text{ m/s}$$

Figure 8. The Planar Anisotropy Ellipse Estimated From Large–Time Drawdown Data.

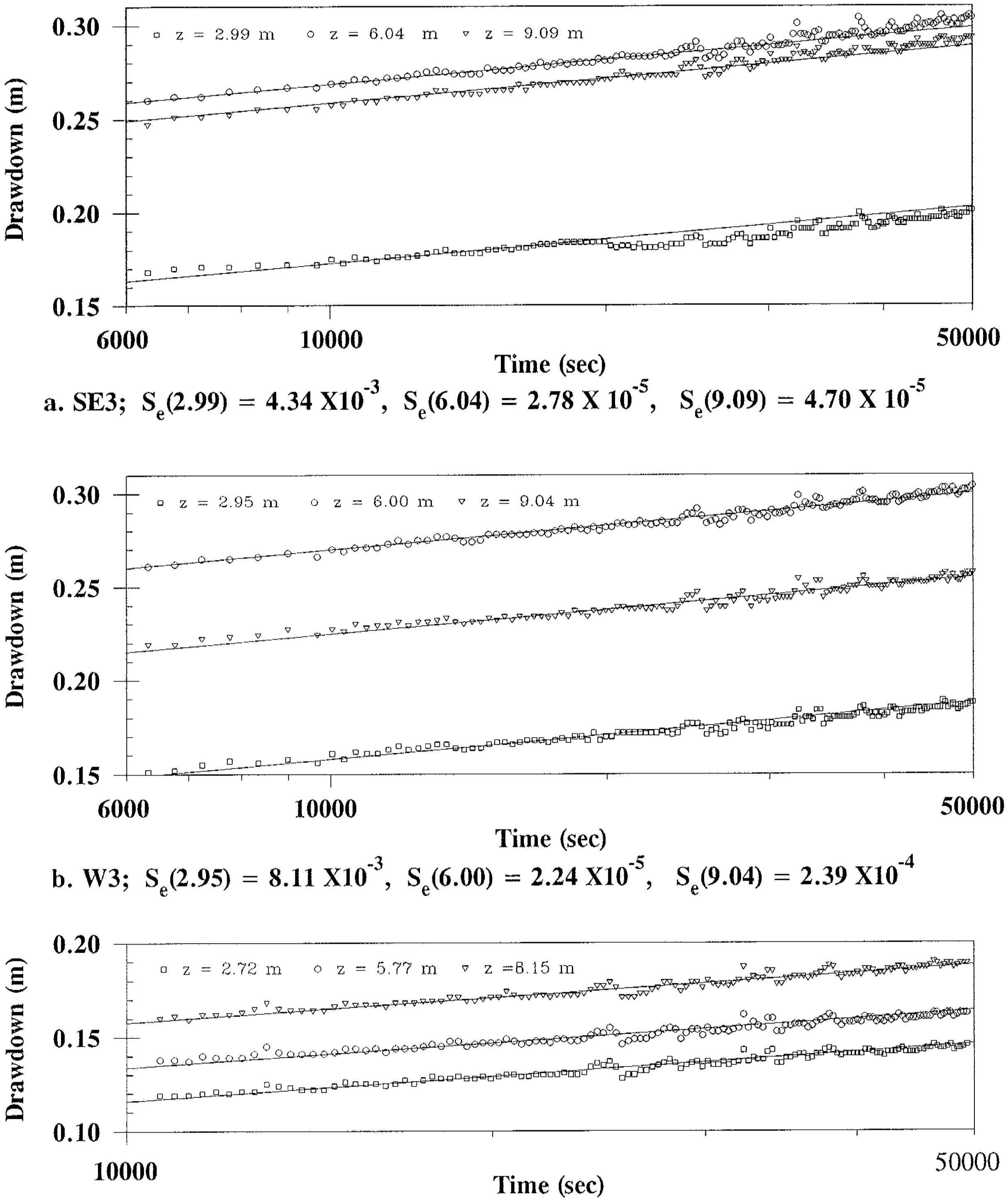
The eccentricity is defined by $(K_x/K_y)^{1/2}$. These values serve as the upper and lower bound for any directional hydraulic conductivity at a specific direction in between the two principal directions. In general, the hydraulic conductivities are of the magnitude of about 10^{-3} meters per second, which is representative of the hydraulic conductivity for sand and gravel [Table 2.2, Freeze and Cherry, 1979].

To demonstrate that this anisotropy condition is accurate, it is used in (23) to calculate the large-time drawdowns at different depths of SE3, W3, and NE6. The comparison of the measured and calculated large-time drawdowns is shown in Figure 9. It is seen that the field data indeed can be reproduced by (23) using the anisotropy condition obtained, supporting the validity of the method and its results developed here for estimating the planar anisotropy.

After the planar anisotropy is determined, the vertical hydraulic conductivity, K_z, can be estimated based on the fact that at large times the vertical drawdown variation is mainly caused by the partial penetration effect, h_p. For a fixed r, the drawdown difference at z_1 and z_2 thus, can be calculated by

$$\Delta h = hp(r,z_1) - hp(r,z_2) \tag{27}$$

where Δh can be estimated by measuring the vertical difference between any two parallel large-time drawdown lines associated with z_1 and z_2, respectively. The large-time partial penetration effect, h_p, is given in (18). The only unknown in (27) is



a. SE3; $S_e(2.99) = 4.34\ X10^{-3}$, $S_e(6.04) = 2.78\ X\ 10^{-5}$, $S_e(9.09) = 4.70\ X\ 10^{-5}$

b. W3; $S_e(2.95) = 8.11\ X10^{-3}$, $S_e(6.00) = 2.24\ X10^{-5}$, $S_e(9.04) = 2.39\ X10^{-4}$

c. NE6; $S_e(2.72) = 1.92\ x\ 10^{-2}$, $S_e(5.77) = 7.45\ X10^{-3}$, $S_e(8.15) = 2.11\ X10^{-3}$

Figure 9. Comparison of Measured (Symbols) with Calculated (Lines) Drawdown at Large-Time.

the parameter, β, which is defined as

$$\beta = \frac{K_z}{K_r} \frac{r^2}{b^2} \tag{28}$$

Therefore, β can be uniquely determined from (27) for a known Δh. Then, K_z can be determined from (28) for the β value obtained. As shown in Figure 10, the large-time partial penetration effect, h_p, at the three depths of SE3 is plotted against different β's. Reading from Figure 4a, Δh between z = 2.99 meters and 9.09 meters is about 0.1 meter, which is related to about 0.0038 for β as indicated by Figure 10. The large-time drawdown difference, Δh, between z = 6.04 meters and 9.09 meters measured from Figure 4a is about 0.01 meter, which refers to a β value approximately equal to 0.0032. These two β's determined are close and their average, 0.0035, is chosen to estimate K_z by means of (28). Now, K_r and r in (28) have to be calculated with the appropriate anisotropic conditions determined already using (2) and (3). As a result, K_z is calculated to be 5.74x10^{-5} meters per second.

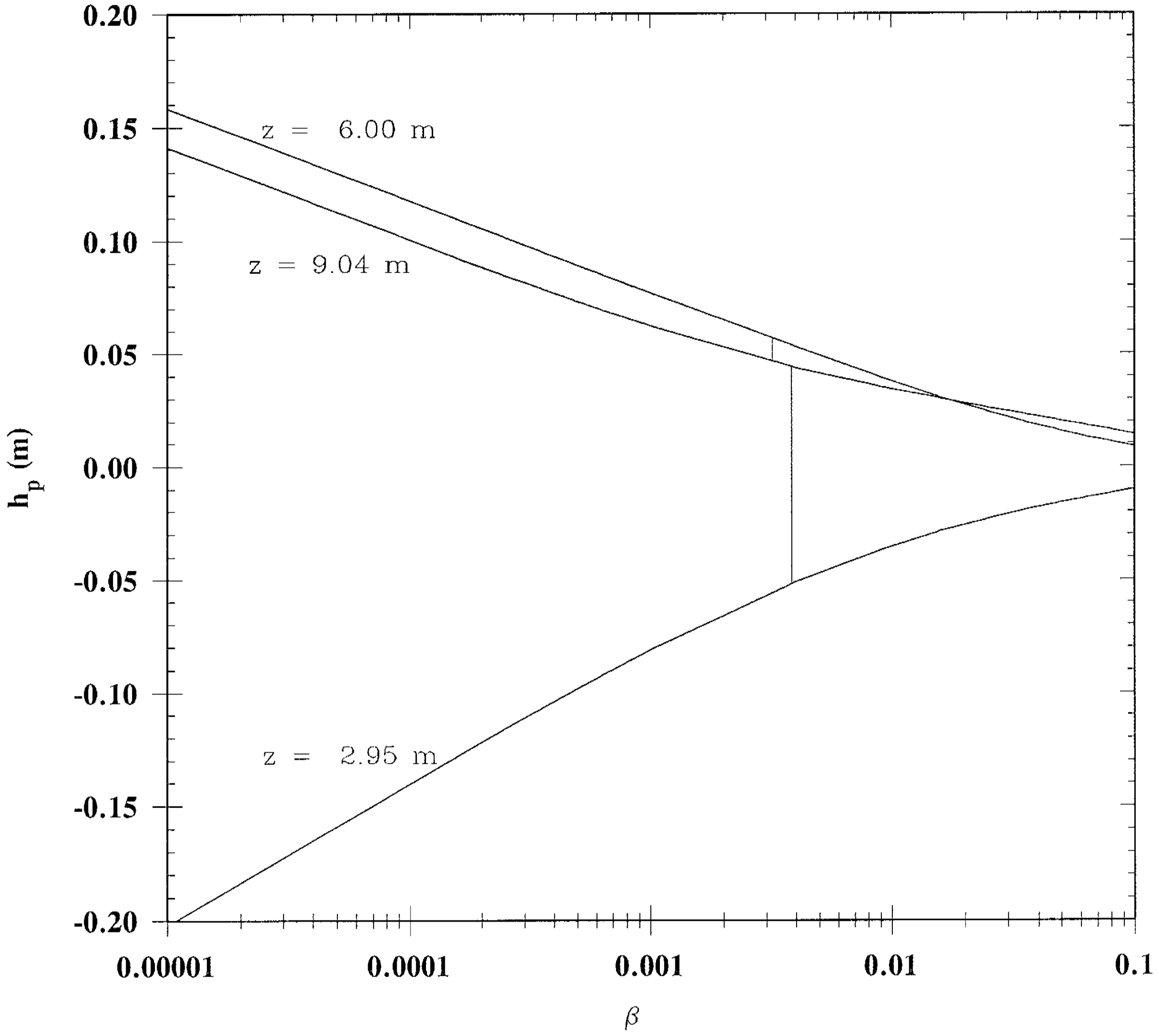


Figure 10. Determination of β from Vertical Variation of the Partial Penetration Effect at SE3.

CONCLUSIONS

In the Laplace-Hankel domain, the three-dimensional unconfined well hydraulics solution given by Neuman (1974) separates into three terms; the baseline condition or Theis solution, the water table effect, and the partial penetration effect. Both the water table effect and partial penetration effect can reach steady state. At large-time, the water table effect reaches a stable condition for all r and z in the vicinity of the pumping well, but the partial penetration effect varies with respect to r and z. For an unconfined aquifer subject to a partially penetrating pumping well, the large-time drawdown histories for different depths of the same planar location are parallel Theis curves. The horizontal hydraulic conductivity was estimated from the slope of the large-time drawdown curves. A system of three simultaneous equations was set up and solved to determine the planar anisotropy tensor. The principal horizontal conductivities for the Sevilleta aquifer are $1.62 \text{ X } 10^{-3}$ in the direction N58°E meters per second and $1.27 \text{ X } 10^{-3}$ meters per second in the direction N32°W. The eccentricity of the anisotropy ellipse is 1.13. The vertical hydraulic conductivity was estimated from the vertical distance between the parallel drawdown histories measured at different depths of the planar location. It is estimated to be $5.74\text{x}10^{-5}$ meters per second.

NOTATION

a	Hankel transform parameter.
b	initial saturated thickness of aquifer (L).
d	vertical distance from initial position of water table to top of screened interval in the pumping well (L).
h	drawdown (L).
h_1	drawdown give by the Theis solution (L).
h_2	depth specific water table effect (L).
h_3	depth specific partial penetration effect (L).
$J_0(x)$	zero order Bessel function of the first kind.
$K_0(x)$	zero order modified Bessel function of the second kind.
K_r	horizonal hydraulic conductivity (L/T).
K_{xx}, K_{xy}, K_{xy}, K_z	elements of anisotropy tensor (L/T).
l	vertical distance from initial position of water table to bottom of screened interval in the pumping well (L).
p	Laplace transform parameter.
Q	pumping rate (L^3/T).
r	radial distance from pumping well (L).
S	elastic storage coefficient.
S_y	specific yield.
t	elapsed time (T).
u	argument of well function, $r^2S/4bK_rt$.

$W(u)$	well function.
x, y, z	spacial coordinates of observation point where drawdown is measured (L).
α_y	K_z/S_y (L/T)
β	$(K_z r^2)/(K_r b^2)$

REFERENCES

Abramowitz, M., and I.A. Stegun, 1970. Handbook of Mathematical Functions Dover Publication, 1046 pp.

Akindunni, F.F., and R.W. Gillham, 1992. Unsaturated and saturated flow in response to pumping of an unconfined aquifer: Numerical investigation of delayed drainage, Ground Water, 30(6), pp. 873-884.

Anderson, W.L., 1979. Numerical integration of related Hankel transforms of orders 0 and 1 by adaptive digital filtering. Geophysics, 44(7), pp. 1287-1305.

Anderson, W.L., 1982. Fast Hankel transforms using related and lagged convolutions, ACM Transactions on Mathematical Software, 8(4), pp. 344-368.

Boulton, N.S., 1954. The drawdown of the water table under non-steady conditions near a pumped well in an unconfined formation. Proc. Inst. Civil Engr., Part 3, pp. 564-579.

Boulton, N.S., 1963. Analysis of data from non-equilibrium pumping tests allowing for the delayed yield from storage. Proc. Inst. Civil Engrs., 26, pp. 469-482.

Chandler, S.P., N. Kapoor, and S.K. Goyal, 1981. Analysis of pumping test data using Marquardt algorithm, Ground Water, 19(3), pp. 225-227.

Chen, C.S., 1985. Analytical and approximate solutions to radial dispersion from an injection well to a geological unit with simultaneous diffusion into adjacent strata, Water Resour. Res., 21(8), pp. 1069-1076.

Chen, C.-S., Christopher Holmes, Weidong Li, David Chace, Mike Fort, Jingfang He, and Jinzhong Liu, 1993. Multidimensional field and theoretical investigation of solute transport from an injection well into a homogeneous and anisotropic aquifer. Technical Completion Report. USGS Report No.14-08-0001-G1744.

Cody, W. J., 1987. SPECFUN - a Portable Special Function Package in New Computing Environments: Microcomputers in Large Scale Scientific Computing, ed. A. Wouk, SIAM, pp. 1-12

Dagan, G., 1967. A method of determining the permeability and effective porosity of unconfined anisotropic aquifers, Water Resour. Res., 3(4), pp. 1059-1071.

Freeze, R.A. and J.A. Cherry, 1979. Groundwater, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Gradshteyn, I.S. and I.M. Ryzhik, 1980. Tables of Integrals, Series, and Products, Academic Press, 1160 pp.

Hantush, M.S., 1961. Drawdown around a partially penetrating well, Proc. Am. Soc. Civil Engrs., 87(HY4), pp. 83-98.

Hantush, M.S., 1964. Hydraulics of Wells in Advances in Hydrosci., ed. V.T. Chow, 1, pp 281-432.

Hantush, M.S., 1966a. Wells in homogeneous anisotropic aquifers, Water Resour. Res. 2(2), pp 273-279.

Hantush, M.S., 1966b. Analysis of data from pumping tests in anisotropic aquifers, J. Geophys. Res. 71(2), pp. 421-426.

Hibsch, G. and A. Kreft, 1980. Determination of aquifer characteristics by the multi-well method, Nuclear Techniques in Groundwater Pollution Research, Rep. STI/PUB/518 197-208, Int. At. Energy Agency, Vienna, Austria.

Hsieh, P.A. and S.P. Neuman, 1985a. Field determination of the three-dimensional hydraulic conductivity tensor of anisotropic media, 1. Theory, Water Resour. Res. 21(11) pp. 1655-1665.

Hsieh, P.A. and S.P. Neuman, 1985b. Field determination of the three-dimensional hydraulic conductivity tensor of anisotropic media, 1. Methodology and application to fractured rocks, Water Resour. Res. 21(11) pp. 1667-1676.

Johns, R.A., L. Semprin, and P.A. Roberts, 1992. Estimating aquifer properties by nonlinear least-squares analysis of pump test response, Ground Water, 30(1), pp. 68-77.

Lakshminarayana, V., and S.P. Rajagopalan, 1978. Type-curve analysis of time-drawdown data for partially penetrating wells in unconfined anisotropic aquifers, Ground Water, 16(5), pp 328-333.

Miller, R.T., 1984. Anisotropy in the Irontan and Galesville sandstones near a thermal-energy-storage well, St. Paul, Minnesota, Ground Water 22(5) pp. 532-537.

Moench, A.F., 1984. Analysis of constant discharge wells by numerical inversion of Laplace transform solutions, in Groundwater Hydraulics (J.S. Rosenshein and G.D. Bennett, eds.). Water Resources Monograph Series 9, AGU, Washington, D.C., pp. 146-170.

Neuman, S.P., 1973. Supplementary comments on theory of flow in unconfined aquifers considering delayed response of the watertable, Water Resources Res., 9(4), pp. 1102-1103.

Neuman, S.P., 1972. Theory of flow in unconfined aquifers considering delayed response of the water table. Water Resour. Res., 8(4), pp. 1031-1044.

Neuman, S.P., 1974. A.S.P. Neuman effect of partial penetration on flow in unconfined aquifers considering delayed gravity response. Water Resour. Res., 10(2), pp. 303-312.

Neuman, S.P., 1975. Analysis of pumping test data from anisotropic unconfined aquifers considering delayed gravity response, Water Resources Res., 11(2), pp. 329-342.

Neuman, S.P., G.R. Walter, H.W. Bentley, J.J.Ward, and D.D. Gonzales, 1984. Determination of horizontal aquifer anisotropy with three wells, Ground Water 22(1) pp. 66-72.

Neuman. S.P., C.L. Winter, C.M. Newman, 1987. Stochastic theory of field-scale Fickian dispersion in anisotropic porous media, Water Resour. Res. 23(3) pp. 453-466.

Nwankwor, G.I., R.W. Gillham, G. van der Kamp, and F.F. Akindunni, 1992. Unsaturated and saturated flow in response to pumping of an unconfined aquifer: Field evidence of delayed drainage. Ground Water, 30(5), pp. 690-700.

Oberhettinger, F. and L. Badii, 1973. Tables of Laplace Transforms, Springer-Verlag, 428 pp.

Papadopulos, J.S., 1965. Nonsteady flow to a well in an infinite anisotropic aquifer, in Proc. of the Dubrovnik Symposium on Hydrology of Fractured Rocks, 21-31, Inter. Assoc. Sci. Hydrology.

Prickett, T.A., 1965. Type curve solution to aquifer tests under water table conditions, Ground Water., 3(3); pp. 5-14.

Sneddon, I.N., 1972. The Use of Integral Transforms, McGraw-Hill.

Stehfest, H., 1970. Algorithm 368 numerical inversion of Laplace transforms, Commun. ACM, 13(1), pp. 47-48.

Stoner, J.D., 1981. l Horizontal anisotropy determined by pumping in Two Powder River basin coal aquifers, Montana, Ground Water 19(1) pp. 34-40.

Theis, C.V., 1935. The relation between the lowering of the piezometric surface and the rate and duration of discharge of a well using groundwater storage, Trans. Amer. Geophys. Union, Vol. 16, pp. 519-524.

Way, S.C. and C.R. McKee, 1982. In-situ determination of three-dimensional aquifer permeability, Ground Water 20(5) pp. 594-603.

APPENDIX

FORTRAN code for computing drawdown using
the Laplace-Hankel domain calculation.

```
************************************************************************
      program drawdn
************************************************************************
*                                                                      *
* Purpose:                                                             *
* This program computes drawdown due to pumping under a wide variety   *
* of pumping and aquifer conditions.                                   *
*                                                                      *
* The aquifer can be:                                                  *
*   a) confined or unconfined                                          *
*   b) isotropic or anisotropic                                        *
*                                                                      *
* The pumping well can be:                                             *
*   a) fully or partially penetrating.                                 *
*                                                                      *
* The observation well can be:                                         *
*   a) fully penetrating, partially penetration, or treated as a       *
*      point.                                                          *
*                                                                      *
* Authors:                                                             *
* christopher holmes                                                   *
* holmes@jupiter.nmt.edu                                               *
*                                                                      *
* Chia-Shyun Chen                                                      *
* Associate Professor of Hydrology                                     *
* New Mexico Institute of Mining and Technology                        *
* Socorro, NM 87801                                                    *
*                                                                      *
*  References on Equations Used in the Computations                    *
*                                                                      *
*  (1) Anderson W. L., 1979, Numerical Integration of Related Hankel   *
*       Transforms of Orders 0 and 1 by Adaptive Digital Filtering,    *
*       Geophysics, Vol. 44 No. 7, pp 1287-1305                        *
*  (2) Anderson W. L., 1982, Fast Hankel Transforms Using Related and  *
*       Lagged Convolutions, ACM Transactions on Mathematical          *
*       Software, Vol. 8, No. 4 pp 344-368                             *
*  (3) Hantush, M.S., 1961, Aquifer Tests on Partially Penetrating     *
*       Wells Proc. Am. Soc. Civl Engrs., vol. 87 No. HY5, pp 171-195  *
*  (4) Hantush, M. S., 1964, Hydraulics of Wells in Advances in        *
*       Hydroscience, vol. 1, Academic, New York                       *
*  (5) Neuman, Shlomo P., 1972, Theory of flow in                          *
*       unconfined aquifers considering delayed response                   *
*       of the water table:  Water Resources Res., Vol.8,                  *
*       No.4, pp.1031-1045                                                 *
*  (6) Neuman, Shlomo P., 1974, Effect of partial                          *
*       penetration on flow in unconfined aquifers                         *
*       considering delayed gravity response:  Water                       *
*       Resources Res., Vol.10, No.9, pp.303-312                           *
*  (7) Neuman, Shlomo P., 1975, Analysis of pumping test data from         *
*       anisotropic unconfined aquifers considering delayed gravity        *
*       response: Water Resources Res., Vol.11, No.2, pp.329-342           *
*  (8) Streltsova, T.D., 1976, Comments on 'Analysis of pumping test       *
*       data from anisotropic unconfined aquifers considering delayed      *
*       gravity response' by Shlomo P. Neuman: Water Resources Res.,       *
*       Vol.12, No.1, pp.113-114                                           *
*  (9) Neuman, Shlomo P., 1979, Perspective on 'Delayed                    *
*       Yield':  Water Resources Res., Vol.15, No.4, pp.899-908            *
* (10) Gambolati, Giuseppe, 1981, Comment on 'Perspective                  *
*       on "Delayed Yield" by Shlomo P. Neuman:  Water Resources           *
*       Res., Vol.17, No.2, pp.443-444                                     *
* (11) Neuman, Shlomo P., 1981, Reply: Water Resources Res.,               *
*       Vol. 17, No. 2, p.445                                              *
* (12) Neuman, Shlomo P., 1981, Delayed drainage in a                      *
*       stream-aquifer system: Journal of the Irrigation and               *
*       Drainage Division, proceedings of the A.S.C.E., Vol.               *
*       107, No.IR4, pp.407-410                                            *
* (14) Way, Shao-Chih and Chester R. McKee, 1982 In-Situ                   *
*       Determination of Three-Dimensional Aquifer Permeabilities          *
*       Groundwater, Vol. 20, No 5, pp 594-603                             *
* (15) Stehfest Harald, 1970, Algorithm 368; Numerical Inversion of        *
*       Laplace Transforms, Communications of the ACM, Vol. 13, No. 1      *
*       pg 47                                                              *
****************************************************************************
*                                                                          *
* Subroutines and external Functions uses include                          *
*                                                                          *
*   BESK0 from SPECFUN by                                                  *
*   W. J. Cody and Laura Stoltz                                            *
C   Mathematics and Computer Science Division                              *
C   Argonne National Laboratory                                            *
C   Argonne, IL 60439                                                      *
*                                                                          *
*   DHANKL by                                                              *
*   W. L. Anderson                                                          *
*   U.S. Geological survey                                                  *
*   Denver, colorado                                                        *
****************************************************************************
*                                                                           *
*  The input file must have the following format                            *
*                                                                           *
* tittle  (a72)                                                             *
* kx      (d20.4)                                                           *
* ky      (d20.4)                                                           *
* ss      (d20.4)                                                           *
* sy      (d20.4)                                                           *
* b       (d20.4)                                                           *
* d       (d20.4)                                                           *
* l       (d20.4)                                                           *
* q       (d20.)                                                            *
* nplaces (i11)                                                             *
* x       (d20.4)                                                           *
* y       (d20.4)                                                           *
* do      (d20.4)                                                           *
* lo      (d20.)                                                            *
* ntimes  (i11)                                                             *
* t       (d20.4)  the number of times listed must equal ntimes             *
* x       (d20.4)  give the next location                                   *
* y       (d20.4)                                                           *
* do      (d20.4)                                                           *
* lo      (d20.4)                                                           *
* t       (d20.4)  the number of times listed must equal ntimes             *
****************************************************************************
*                                                                           *
*  Input variables                                                          *
*                                                                           *
* b       = initial saturated thickness of aquifer, meters                  *
* d       = vertical distance between the top of the screen in the          *
*              pumping well and initial position of water table,            *
*              meters                                                       *
* do      = vertical distance between the top of the screen in the          *
*              observation well and initial position of water table,        *
*              meters                                                       *
* Kx      = horizontal hydraulic conductivity in the X direction,           *
*              meters/second                                                *
* Ky      = horizontal hydraulic conductivity in the Y direction,           *
*              meters/second                                                *
*  kz      = vertical hydraulic conductivity, meters/second                  *
*  l       = vertical distance between the bottom of the screen in the       *
*               pumping well and the initial position of water table,        *
*               meters                                                       *
*  lo      = vertical distance between the bottom of the screen in the       *
*               observation well and the initial position of water           *
*               table, meters                                                *
*  nplace  = number of places to read from input file                        *
*  ntimes  = number of times to read from input file                         *
*  q       = pumping rate, cubic meters / second                             *
*  ss      = specific storage, 1/meters                                      *
*  sy      = specific yield, unitless                                        *
*  t       = elapsed time, seconds                                           *
*  title   = title of simulation                                             *
*  x       = x coordinate of the observation well, meters                    *
*  y       = y coordinate of the observation well, meters                    *
*                                                                            *
******************************************************************************
*                                                                            *
*   Output variables                                                         *
*                                                                            *
*  h     = drawdown                                                          *
*  h1    = drawdown given by the Thies solution                              *
*  h2    = unconfined (water table) effect                                   *
*  h3    = partial penetration effect                                        *
*  sigma = Ssb/Sy                                                            *
*  ts    = dimensionless time with respect to S (Ss*b)                       *
*  ty    = dimensionless time with respect to Sy                             *
******************************************************************************
*                                                                            *
*   Intermedaiate variables used in computations                             *
*                                                                            *
*  Kd = anisotropy ration, Kz/Kr                                             *
*  Kr = effective horizontal hydraulic conductivity, see reference 11        *
*  r  = effective radius, see reference 11                                   *
*  u  = argement of well function                                            *
*  wu = well function                                                        *
*  zl = vertical distance from base of aquifer to bottom of screened         *
*          interval in observation well, (i.e. Neuman's z1)                  *
*  zu = vertical distance from base of aquifer to top of screened            *
*             interval in observation well, (i.e. Neuman's z2)               *
******************************************************************************
      double precision  kx, ky, kz, ss, sy,
     +    b, d, do, l, lo, q, r, t,
     +    h, h1, h2, h2d, h3, h3a, h3d,
     +    b2, kr, kr1, log2, p, pi, q4pit,
     +    qh2pt, qh2va, qh3pt, qh3va, dh2pt, dh2va, dh3pt, dh3va,
     +    r, r1, u, u1, u2, uy, uy1, uy2, wu,
     +    zl, zu,
     +    f3pt, f3va, f3fp, ppf3pt, ppf3va, ppf3fp, w,
     +    beta, kd, sigma, ts, ty,
     +    wt(18)
*
      integer i, iprint, j, k, nplace, ntimes
*
      character infile*10, outfile*10, outdls*12, outdms*12, title*72
*
      logical steady
*
      external f3fp, f3pt, f3va, ppf3pt, ppf3va, ppf3fp
************************************************************************
**  The following data statement is used to set the parameters used   *
**  in calling the subroutine dhankl.                                   *
************************************************************************
      double precision tol, dwork(801,1), dans(1,1), arg(1)
      integer nord(1)
      data tol/1.0d-4/,ntol/1/,nb/1/,nrel/1/,nord/0/
************************************************************************
** The following common statements are used to pass variables to the  *
** the functions call by dhankl.                                        *
************************************************************************
      common /c1/ kr, ss, kz, sy
      common /c2/ b, p, t, zl, zu
      common /ld/ l, d
      common /cpi/ pi
************************************************************************
** The common block weight contains the weighting factors for the      *
** Stehfest inversion.                                                  *
************************************************************************
      common /weight/ wt
*
      log2 = dlog(2.d0)
      pi   = 4.d0 * datan(1.d0)
************************************************************************
**   Specify type of output; dimensional, dimensionless, or both.      *
************************************************************************
```

```
 10   write(6,*) 'Enter 1 for dimensional, 2 for dimensionless or ',
     +     '3 for both'
      read(5,*) iprint
      if(iprint.ne.1 .and. iprint.ne.2 .and. iprint.ne.3) goto 10
************************************************************************
**  Enter names of input and output files interactively, open files.   *
************************************************************************
      write(6,*) 'Enter name of input file'
      read(5,*) infile
*
      write(6,*) 'Enter name of output file'
      read(5,*) outfile
*
      open(unit = 7, file = infile, status = 'old')
      lfn      = index(outfile, ' ')
      if(iprint .ne. 2) then
         outdms(1:lfn-1)   = outfile
         outdms(lfn:lfn+3) = '.dms'
         open(unit = 8, file = outdms)
         write(6,*) 'Dimensional Output is in File ', outdms
      endif
      if(iprint .ne. 1) then
         outdls(1:lfn-1)   = outfile
         outdls(lfn:lfn+3) = '.dls'
         open(unit = 9, file = outdls)
         write(6,*) 'Dimensionless Output is in file ', outdls
      endif
************************************************************************
**          Read aquifer and pumping well parameters                   *
************************************************************************
      read(7,5000) title
      if(iprint .ne. 2) then
         write(8,5010) title
         write(8,5020)
      endif
      if(iprint .ne. 1) then
         write(9,5010) title
         write(9,5020)
      endif
      read(7,5030) kx
      if(iprint .ne. 2) write(8, 5040) kx
      if(iprint .ne. 1) write(9, 5040) kx
      read(7,5030) ky
      if(iprint .ne. 2) write(8, 5050) ky
      if(iprint .ne. 1) write(9, 5050) ky
      read(7,5030) kz
      if(iprint .ne. 2) write(8, 5060) kz
      if(iprint .ne. 1) write(9, 5060) kz
      read(7,5030) ss
      if(iprint .ne. 2) write(8, 5070) ss
      if(iprint .ne. 1) write(9, 5070) ss
      read(7,5030) sy
      if(iprint .ne. 2) write(8, 5080) sy
      if(iprint .ne. 1) write(9, 5080) sy
      read(7,5030) b
      if(iprint .ne. 2) write(8, 5100) b
      if(iprint .ne. 1) write(9, 5100) b
      read(7,5030) d
      if(iprint .ne. 2) write(8, 5110) d
      if(iprint .ne. 1) write(9, 5110) d
      read(7,5030) l
      if(iprint .ne. 2) write(8, 5120) l
      if(iprint .ne. 1) write(9, 5120) l
      if(iprint .ne. 1) then
        write(9, 5115) d/b
        write(9, 5125) l/b
      endif
      read(7,5030) q
      if(iprint .ne. 2) write(8, 5130) q
      if(iprint .ne. 1) write(9, 5130) q
      if(iprint .ne. 1) write(9, 5131)
*********************************************************************************
**        Check aquifer and pumping parameters for errors                       *
*********************************************************************************
      if(kx .le. 0. ) then
        write(6,5400)
        if(iprint .ne. 2) write(8,5400)
        if(iprint .ne. 1) write(9,5400)
        stop
      endif
      if(ky .le. 0. ) then
        write(6,5420)
        if(iprint .ne. 2) write(8,5420)
        if(iprint .ne. 1) write(9,5420)
        stop
      endif
```

```
if(kz .lt. 0.d0) then
   write(6,5440)
   if(iprint .ne. 2) write(8,5440)
   if(iprint .ne. 1) write(9,5440)
   stop
endif
if(ss .lt. 0.d0) then
   write(6,5460)
   if(iprint .ne. 2)write(8,5460)
   if(iprint .ne. 1) write(9,5460)
   stop
elseif(ss .ge. 1.d0) then
   write(6,5470)
   if(iprint .ne. 2) write(8,5470)
   if(iprint .ne. 1) write(9,5470)
   stop
endif
if(sy .lt. 0.d0) then
   write(6,5480)
   if(iprint .ne. 2) write(8,5480)
   if(iprint .ne. 1) write(9,5480)
   stop
elseif(sy .gt. 1.d0) then
      write(6,5490)
      if(iprint .ne. 2) write(8,5490)
      if(iprint .ne. 1) write(9,5490)
      stop
endif
if(sy .gt. 0.d0 .and. kz .eq. 0.d0) then
   write(6,5500)
   if(iprint .ne. 2) write(8,5500)
   if(iprint .ne. 1) write(9,5500)
   stop
endif
if(b .lt. 1) then
   write(6,5520)
   if(iprint .ne. 2) write(8,5520)
   if(iprint .ne. 1) write(9,5520)
   stop
endif
if(b .le. 0.d0) then
   write(6,5530)
   if(iprint .ne. 2) write(8,5530)
```

```
        if(iprint .ne. 1) write(9,5530)
        stop
      endif
      if(q .eq. 0 .and. iprint .ne. 2) then
        write(6,5540)
        write(8,5540)
        stop
      endif
**********************************************************************
**   Write aquifer and pumping well characteristics to output file  *
**********************************************************************
      if(iprint .ne. 2) write(8,5135)
      if(iprint .ne. 1) write(9,5135)
*
      if(sy .eq. 0.d0) then
        if(iprint .ne. 2) write(8,5140)
        if(iprint .ne. 1) write(9,5140)
      elseif(sy .gt. 0.d0) then
        if(iprint .ne. 2) write(8,5142)
        if(iprint .ne. 1) write(9,5142)
      endif
*
      if(kx .eq. ky) then
        if(iprint .ne. 2) write(8,5150)
        if(iprint .ne. 1) write(9,5150)
      elseif(kx .ne. ky) then
        if(iprint .ne. 2) write(8,5151)
        if(iprint .ne. 1) write(9,5151)
      endif
      if(d .eq. 0.d0 .and. l .eq. b) then
        if(iprint .ne. 2) write(8,5160)
        if(iprint .ne. 1) write(9,5160)
      else
          if(iprint .ne. 2) write(8,5161)
          if(iprint .ne. 1) write(9,5161)
      endif
**********************************************************************
** Compute drawdown                                                 *
**********************************************************************
      if(iprint .ne. 2) write(8,5190)
      if(iprint .ne. 1) write(9,5191)
      if(sy .gt. 0.d0) then
        sigma =  ss*b/sy
```

```
          if(iprint .ne. 1) write(9,5260) sigma
        endif
*
        b2 = b*b
        dl = 1-d
        kr1 = kx*ky
        kr  = dsqrt(kr1)
        kd  = kz / kr
        dh2pt = -2.d0*b / kd /dl
        dh3pt =  4.d0*b/pi/dl
        q4pit = q / 4.d0 /pi / b/ kr
        qh2pt = -q/2.d0/pi/kz/dl
        qh3pt = q / pi / pi / kr /dl
        rtol  = 2.d0*b/dsqrt(kd)
        tptol = b2*ss/kz/2.d0
        u1    = ss/4.0d0 / kr
        uy1   = (ss*b + sy)/4.d0/kr/b
*
        read(7,5180) nplace
        do 3000 i = 1, nplace
          h   = 0.d0
          h1  = 0.d0
          h2  = 0.d0
          h2d = 0.d0
          h3  = 0.d0
          h3d = 0.d0
          wu  = 0.d0
          steady = .false.
          told   = 0.0
          read(7,5030) x
          read(7,5030) y
          read(7,5030) do
          read(7,5030) lo
          if(lo .lt. do) then
            write(6,5550)
            if(iprint .ne. 2) write(8,5550)
            if(iprint .ne. 1) write(9,5550)
            stop
          endif
          r1    = (x*x*ky + y*y*kx) / kr
          r     = dsqrt(r1)
          ttol  = 10.d0*r1*sy/kr/b
          u2    = u1  * r1
```

```
      uy2   = uy1 * r1
      zl    = b - lo
      zu    = b - do
      beta  = r1*kd/b2
      qh2va = qh2pt/(zu -zl)
      dh2va = dh2pt/(zu -zl)
      qh3va = q * b / pi/pi/pi/ kr / (l-d) / (zu-zl)
      dh3va = 4.d0*b*b/pi/pi/(l-d)/(zu-zl)
      read(7,5180) ntimes
      if(sy .eq. 0.d0) then
        if(d .eq. 0.d0 .and. 1 .eq. b) then
***********************************************************************
**  Case 1:                                                           *
**  Confined aquifer with fully penetrating pumping well              *
***********************************************************************
          if(iprint .ne. 2) then
            write(8,5170)
            write(8,5211) x, y
            write(8,5600)
          endif
          if(iprint .ne. 1) then
            write(9,5170)
            write(9,5211) x, y
            write(9,5320)
          endif
          do 100 k = 1, ntimes
            read(7,5030) t
            u  = u2 / t
            wu = w(u)
            h1 = q4pit * wu
            h  = h1
            if(iprint .ne. 2) write(8,5300) t, h, h1, 0., 0.
            if(iprint .ne. 1) write(9,5311) t, 1.d0/u, u, wu
 100        continue
        elseif(d .ne. 0.d0 .or. lo .ne. b) then
***********************************************************************
**  Cases 2 and 3                                                     *
**  Confined aquifer with partially penetrating pumping well          *
**  Case 2 drawdown is depth specific                                 *
**  Case 3 drawdown is vertically averaged                            *
***********************************************************************
          if(iprint .ne. 1) then
            if(do .eq. lo) write(9,5171)
        if(do .ne. lo) write(9,5172)
        if(r .gt. rtol) write(9,5220)
        write(9,5212)
        write(9,5213) x, y, do, lo
        write(9,5214) beta, do/b, lo/b
        write(9, 5610)
      endif
      if(iprint .ne. 2) then
        if(do .eq. lo) write(8,5171)
        if(do .ne. lo) write(8,5172)
*         if(r .gt. rtol) write(8,5220)
        write(8,5210) x, y, do, lo
        write(8,5600)
      endif
      do 400 j = 1, ntimes
        read(7,5030) t
        if(t .lt. told) then
          write(6,5560)
          if(iprint .ne. 2) write(8,5560)
          if(iprint .ne. 1) write(9,5560)
          stop
        endif
        p  = log2 / t
        u  = u2 / t
        wu = w(u)
        h1 = q4pit * wu
        if(r .le. rtol .and. .not. steady) then
          if(do .eq.lo) then
            call pppt(h3a, 0.d0, b, d, l, p, r, zl, zu)
            h3  = qh3pt * h3a
            h3d = dh3pt * h3a
          else
            call ppva(h3a, 0.d0, b, d, l, p, r, zl, zu)
            h3  = qh3va * h3a
            h3d = dh3va * h3a
          endif
          if(t .gt. tptol) steady = .true.
        endif
        h  = h1 + h3
        hd = wu + h3d
        if(h .lt. 0.d0) h = 0.d0
        if(hd .lt. 0.d0) hd = 0.d0
        if(iprint.ne.2) write(8,5300) t,h,h1,0.d0,h3
```

```
            if(iprint.ne.1) write(9,5312) t,1/u,hd,wu,0.d0,h3d
            told = t
400        continue
        endif
************************************************************************
**   Unconfined aquifer with fully penetrating pumping well.           *
************************************************************************
      elseif(sy .gt. 0.d0) then
        if(d .eq. 0.d0 .and. l .eq. b) then
          h3     = 0.d0
          h3d    = 0.d0
          if(iprint .ne. 2) then
            if(do .eq. lo) write(8,5176)
            if(do .ne. lo) write(8,5177)
            write(8,5210)x, y, do, lo
            write(8,5600)
          endif
          if(iprint .ne. 1) then
            write(9,5212)
            if(do .eq. lo) write(9,5176)
            if(do .ne. lo) write(9,5177)
            write(9,5213) x, y, do, lo
            write(9,5214) beta, do/b, lo/b
            write(9,5620)
          endif
          do 1300 j = 1, ntimes
            read(7,5030) t
            u  = u2 / t
            wu = w(u)
            h1 = q4pit * wu
            if(t .gt. ttol .and. r .ge. rtol) then
              uy  = uy2/t
              wuy = w(uy)
              h   = q4pit*wuy
              hd  = wuy
              h2  = h - h1
              h2d = h2 / q4pit
            else
              p  = log2 / t
              if(do .eq. lo) then
                call dhankl(r, nb, 1, tol, ntol, nord, f3pt,
    +                    dwork,dans,arg, nofun1,ierr)
                h2  = qh2pt*dans(1,1)
```

```
                  h2d = dh2pt * dans(1,1)
               elseif(do .eq. 0.d0 .and. lo .eq. b) then
                  call dhankl(r, nb, 1, tol, ntol, nord, f3fp,
     +                    dwork,dans,arg, nofun1,ierr)
                  h2  = qh2va * dans(1,1)
                  h2d = dh2va * dans(1,1)
               else
                  call dhankl(r, nb, 1, tol, ntol, nord, f3va,
     +                    dwork,dans,arg, nofun1,ierr)
                  h2  = qh2va * dans(1,1)
                  h2d = dh2va * dans(1,1)
               endif

               h  = h1 + h2
               hd = wu + h2d
            endif
            ts = 0.25d0 / u
            ty = ts *sigma
            if (h2 .gt. 0.d0) h2 = 0.d0
            if (h2d .gt. 0.d0) h2d = 0.d0
            if(iprint .ne. 2) write(8,5300) t, h, h1, -h2, 0.d0
            if(iprint .ne. 1) write(9,5313)t,ts,ty,hd,wu,-h2d,0.d0
 1300      continue
      elseif(d .ne. 0.d0 .or. l .ne. b) then
************************************************************************
**  Unconfined aquifer with partially penetrating pumping well.        *
************************************************************************
         if(iprint .ne. 2) then
            write(8,5210)x, y, do, lo
            write(8,5600)
         endif
         if(iprint .ne. 1) then
            write(9,5212)
            if(do .eq. lo) write(9,5178)
            if(do .ne. lo) write(9,5179)
            write(9,5213) x, y, do, lo
            write(9,5214) beta, do/b, lo/b
            write(9,5620)
         endif
         do 1700 j = 1, ntimes
            read(7,5030) t
            u  = u2 / t
            wu =  w(u)
```

```
           h1 = q4pit * wu
           if(t .gt. ttol .and. r .ge. rtol) then
              uy  = uy2/t
              wuy = w(uy)
              h   = q4pit*wuy
              hd  = wuy
              h3  = 0.d0
              h3d = 0.d0
              h2  = h - h1
              h2d = h2 / q4pit
           else
              p  = log2/t
              if(do .eq. lo) then
                 call dhankl(r, nb, 1, tol, ntol, nord, ppf3pt,
+                      dwork, dans, arg, nofun1,ierr)
                 h2  = qh2pt * dans(1,1)
                 h2d = dh2pt * dans(1,1)
                 if(.not. steady) then
                    call pppt(h3a, 0.d0, b, d, l, p, r, zl, zu)
                    h3  = qh3pt * h3a
                    h3d = dh3pt * h3a
                 endif
              elseif(do .eq. 0.d0 .and. lo .eq. b) then
                 call dhankl(r, nb, 1, tol, ntol, nord, ppf3fp,
+                      dwork, dans,arg, nofun1,ierr)
                 h2  = qh2va * dans(1,1)
                 h2d = dh2va * dans(1,1)
              elseif(lo .gt. do) then
                 if(.not. steady) then
                    call ppva(h3a, 0.d0, b, d, l, p, r, zl, zu)
                    h3  = qh3va * h3a
                    h3d = dh3va * h3a
                 endif
                 call dhankl(r, nb, 1, tol, ntol, nord, ppf3va,
+                      dwork, dans,arg, nofun1,ierr)
                 h2  = qh2va * dans(1,1)
                 h2d = dh2va * dans(1,1)
              endif
              if(t .gt. tptol) steady = .true.
              h  = h1 + h2 + h3
              hd = wu + h2d + h3d
              if(h .lt. 0.d0) h = 0.d0
              if(hd .lt. 0.d0) hd = 0.d0
              if(h2 .gt. 0.d0) h2 = 0.d0
              if(h2d .gt. 0.d0) h2d = 0.d0
            endif
            ts = 0.25d0 / u
            ty = ts *sigma
            if(iprint .ne. 2)write(8,5300) t, h, h1, -h2, h3
            if(iprint .ne. 1)write(9,5313)t,ts,ty,hd,wu,-h2d,h3d
 1700       continue
          endif
        endif
 3000 continue
************************************************************************
************************************************************************
 5000 format(a72)
 5010 format(’TITLE: ’, a72, //)
 5020 format(7x, ’X, Y, Z, ARE THE THREE PRINCIPAL DIRECTIONS.  ’,
     + ’Z IS VERTICAL.’,/,
     +     7x,’THE DEPTHS TO THE TOP AND BOTTOM OF THE SCREENED ’,
     +     ’INTERVAL IN THE’, /,
     +     7x,’PUMPING AND OBSERVATION WELLS ARE ’,
     +     ’MEASURED FROM THE TOP OF THE AQUIFER.’, ///,
     +     ’INPUT PARAMETERS:’,/ /,
     +     7x,’UNITS ARE METERS AND SECONDS’,/)
 5030 format(d20.4)
 5040 format(7x,’HYDRAULIC CONDUCTIVITY IN X DIRECTION, Kx, (m/s): ’,
     + d9.4)
 5050 format(7x,’HYDRAULIC CONDUCTIVITY IN Y DIRECTION, Ky, (m/s): ’,
     + d9.4)
 5060 format(7x,’HYDRAULIC CONDUCTIVITY IN Z DIRECTION, Kz, (m/s): ’,
     + d9.4,/)
 5070 format(7x,’SPECIFIC STORAGE, Ss, (1/m):’, d9.4)
 5080 format(7x,’SPECIFIC YIELD, Sy, (dimsionless):  ’, d9.4,/)
 5100 format(7x, ’INITIAL SATURATED THICKNESS OF AQUIFER, b:’,15X,f15.3)
 5110 format(7x,’VERTICAL DISTANCE TO TOP OF SCREEN IN PUMPING ’,
     +      ’WELL,  d:  ’,f15.3)
 5115 format(/, 7x,’NORMALIZED DISTACNCE TO TOP OF SCREEN IN PUMPING’,
     +     ’ WELL, d/b: ’,f7.4)
 5120 format(7x, ’VERTICAL DISTANCE TO BOTTOM OF SCREEN IN PUMPING ’,
     +      ’WELL, l:’,f15.3)
 5125 format(7x,’NORMALIZED DISTANCE TO BOTTOM OF SCREEN IN PUMPING’,
     +     ’ WELL l/b:’, f7.4)
 5130 format(/,7x, ’PUMPING RATE, Q (cubic meters/second): ’, d11.4,/)
 5131 format(7x, ’DIMENSIONLESS CALCULATIONS DO NOT INVOLVE Q’)
```

```
5135 format(///, 'AQUIFER AND PUMPING WELL CHARACTERISTICS, ',
   +     'AS DETERMINED FROM INPUT PARAMETERS:', //)
5140 format(7x, 'THE AQUIFER IS CONFINED.', /)
5142 format(7x, 'THE AQUIFER IS UNCONFINED.', /)
5150 format(7x, 'THE AQUIFER IS HORIZONTALLY ISOTROPIC, Kx = Ky.',/)
5151 format(7x, 'THE AQUIFER IS ANISOTROPIC, Kx not equal Ky.',/)
5160 format(7x, 'THE PUMPING WELL IS FULLY PENETRATING.')
5161 format(7x, 'THE PUMPING WELL IS PARTIALLY PENETRATING.',/)
5170 format(//,7x, 'SIMULATION CONDITIONS CORRESPONDE TO CASE 1 OF',
   +     ' THE TABLE.  DRAWDOWN IS GIVEN', /, 7x,
   +     'BY THE THEIS EQUATION.  THE PARAMETERS do, lo, and '
   +      'Kz ARE NOT USED IN THE', /, 7x, 'CALCULATIONS.')
5171 format(//, 7x, 'THE SIMULATION CONDITIONS CORRESPONDE ',
   +     'TO CASE 2 OF THE TABLE.  DEPTH SPECIFIC ', /, 7x
   +     'DRAWDOWN IN CONFINED AQUIFER WITH PARTIALLY ',
   +     'PENETRATING PUMPING WELL.')
5172 format(//, 7x, 'THE SIMULATION CONDITIONS CORRESPONDE ',
   +     'TO CASE 3 OF THE TABLE.  VERTICALLY',  /, 7x,
   +     'AVERAGED DRAWDOWN IN CONFINED AQUIFER WITH PARTIALLY ',
   +     'PENETRATING PUMPING WELL.')
5176 format(/,7x, 'SIMULATION CONDITIONS CORRESPONDE TO CASE 4 OF',
   +     ' THE TABLE.  DEPTH SPECIFIC ', /, 7x,
   +     'DRAWNDOWN IN AN UNCONFINED AQUIFER WITH FULLY ',
   +     'PENETRATING', /, 7x, 'PUMPING WELL.',/)
5177 format(/,7x, 'SIMULATION CONDITIONS CORRESPONDE TO CASE 5 OF',
   +     ' THE TABLE.  VERTICALLY', /, 7x,
   +     'AVERAGED DRAWNDOWN IN AN UNCONFINED AQUIFER WITH',
   +     ' FULLY PENETRATING', /, 7x, 'PUMPING WELL.',/)
5178 format(/,7x, 'SIMULATION CONDITIONS CORRESPONDE TO CASE 6 OF',
   +     ' THE TABLE.  DEPTH SPECIFIC ', /, 7x,
   +     'DRAWNDOWN IN AN UNCONFINED AQUIFER WITH PARTIALLY ',
   +     'PENETRATING PUMPING WELL.', /)
5179 format(/,7x, 'SIMULATION CONDITIONS CORRESPONDE TO CASE 7 OF',
   +     ' THE TABLE.  VERTICALLY ', /, 7X,
   +     'AVERAGED DRAWNDOWN IN AN UNCONFINED AQUIFER WITH ',
   +     'PARTIALLY PENETRATING', /, 7x,'PUMPING WELL.', /)
5180 format(i11)
5190 format(///, 'RESULTS OF CALUCATIONS:',/)
5191 format(///, 'DIMENSIONLESS RESULTS OF CALCULATIONS:',/)
5210 format(/,'DRAWDOWN, h at x =', f10.3, 2x, 'y =',
   +     f10.3, 2x, 'do =', f10.3, 2x, 'lo =', f10.3, / )
5211 format(/,'DRAWDOWN, h at x =', f15.3, 15x, 'y =', 2x, f15.3,/)
5212 format(/, 4x, 'DIMENSIONLESS DRAWDOWN AT:',/)
```

```
5213 format(4x, ’x =’, f10.3, 4x, ’y =’, 4x, f10.3, 4x, ’do =’,
   +     f10.3, 4x, ’lo =’, f10.3,/)
5214 format(4x, ’BETA, Kd r^2 / b^2 =’, d10.4, 6x, ’do/b =’,
   +     f7.4, 6x, ’lo/b = ’, f7.4, /)
5215  format(4x, "r/B, r*sqrt(K/[Tb’]) =", d10.4, /)
5216 format(4x, "r/B, r*sqrt(K/[Tb’]) =", d10.4, 6x, ’do/b =’,
   +     f7.4, 6x, ’lo/b = ’, f7.4, /)
5220 format(/, 7x, ’r > 1.5 b sqrt(Kr/Kz); PARTIAL PENETRATION ’,
   +     ’EFFECT IS NEGLIGIBLE’)
5250 format(7x, ’Kd, Kz/Kr:   ’, d9.4)
5260 format(5x, ’SIGMA, S/Sy: ’, d9.4/)
5300 format(5f15.3)
5310 format(d10.4, 4x, d10.4, 4d15.4)
5311 format(4d20.4)
5312 format(6d14.4)
5313 format(7d12.4)
5320 format(12x, ’time’, 17x, ’1/u’, 18x, ’u’, 18x, ’W(u)’)
5321 format(14x, ’t’, 18x, ’1/u’, 18x, ’u’, 16x, ’W(u,r/B)’)
5400 format(7x, ’Error; Kx must be greater than zero!’)
5420 format(7x, ’Error; Ky must be greater than zero!’)
5440 format(7x, ’Error; Kz must be greater than or equal to zero!’)
5460 format(7x, ’Error; Ss must be greater than or equal to zero!’)
5470 format(7x, ’Error; Ss must be less than one!’)
5480 format(7x, ’Error; Sy must be greater than or equal to zero!’)
5490 format(7x, ’Error; Sy must be less than one!’)
5500 format(7x, "Error; when Kz = 0 the aquifer is confined !", /,
   +     ’reset Sy = 0, or Kz greater than zero.’)
5520 format(7x, ’Error; l must be less than b’)
5530 format(7x, ’Error; b must be greater than zero’)
5540 format(7x, ’Error; When dimensional output is chosen, ’,/
   +     ’Q must be greater than zero!’)
5550 format(7x, ’Error; do must be less than or equal to lo!’)
5560 format(7x, ’Error; times must be entered in ascending order’)
5600 format(68x, ’PARTIAL’, /,
   +     37x, ’CONFINED’, 5x, ’UNCONFINED’, 4x ’PENETRATION’, /
   +     10x, ’TIME’, 8x, ’DRAWDOWN’, 7x, ’DRAWDOWN’, 9x, ’EFFECT’,
   +     9x, ’EFFECT’, /,
   +     28x, ’h’, 13x, ’h1’, 12x, ’h2’, 14x, ’h3’)
5610 format(72x, ’DIMENSIONLESS’, /,
   +     44x, ’DIMENSIONLESS’, 2x, ’DIMENSIONLESS’, 4x, ’PARTIAL’ /,
   +     30x,’DIMENSIONLESS’, 4x, ’CONFINED’, 4x, ’UNCONFINED’, 4x,
   +     ’PENETRATION’, /,
   +     8x,’t’, 12x, ’1/u’, 9x, ’DRAWDOWN’, 6x, ’DRAWDOWN’, 7x,
     +     'EFFECT', 8x, 'EFFECT', /,
     +     36x, 'h', 13x, 'h1', 12x, 'h2', 12x, 'h3')
 5620 format('Ts IS DIMESIONLESS TIME WITH RESPECT TO Ss,',
     +     '(t*Kr)/(Ss*r^2)', /,
     +     'Ty IS DIMESIONLESS TIME WITH RESPECT TO Sy,',
     +     '(t*Kr)/(Sy*r^2)', //,
     +     72x, 'DIMENSIONLESS', /,
     +     47x, 'DIMENSIONLESS', 1X,'DIMENSIONLESS',2x,
     +      'PARTIAL', /,
     +     35x, 'DIMENSIONLESS', 3x,'CONFINED',3x, 'UNCONFINED',
     +      2x, 'PENETRATION', /,
     +     6x, 't', 11x, 'Ts', 10x, 'Ty', 6x 'DRAWDOWN', 5x,
     +     'DRAWDOWN', 5x, 'EFFECT', 6x, 'EFFECT', /,
     +     42x, 'h', 11x, 'h1', 10x, 'h2', 10x, 'h3')
      end
************************************************************************
************************************************************************
************************************************************************
************************************************************************
      subroutine dhankl(bmax,nb,nrel,tol,ntol,nord,fun1,dwork,
     * dans,arg,nofun1,ierr)
c========================================
      integer nb,nrel,ntol,nord(nrel),nofun1,ierr
      double precision bmax,tol,dwork(801,nrel),dans(nb,nrel),arg(nb)
c=========================================
c
c  purpose
c
c    the purpose of subprogram dhankl is to provide in double precision
c   a general algorithm for fast real hankel transforms of orders
c   0 and 1 using related and lagged convolutions.
c
c  author
c
c   anderson, w.l., u.s. geological survey, denver, colorado.
c
c  references
c
c   1. anderson, w.l., improved digital filters for evaluating
c      fourier and hankel transform integrals. n.t.i.s rept.
c      pb-242-800, springfield, va., 1975.
c
c   2. anderson, w.l., numerical integration of related hankel
```

```
c       transforms of orders 0 and 1 by adaptive digital filtering.
c       geophysics 44 (july 1979), 1287-1305.
c
c  language
c
c    ans-fortran (x3.9-1966) is used, with the exception of the
c    characters <,[,&,:,],> appearing in some comment statements.
c
c  abstract
c
c      by combining both adaptive lagged convolution (see [1]) and
c    adaptive related convolution (see [2]), subprogram dhankl
c    minimizes external fun1 calls (nofun1 at exit) in evaluating a
c    total of nb*nrel real hankel transforms of orders 0 and (or) 1,
c    where nb is the number of lagged convolutions, and nrel is the
c    number of related convolutions.
c      direct convolution methods (see [1],[2]) do not require bessel
c    function evaluations, and hence are generally an order of
c    magnitude faster to compute than most direct numerical
c    integration methods.  by using previously saved transform input
c    function evaluations, both lagged convolution and related
c    convolution further reduce significantly the number of transform
c    input function evaluations required over direct convolution.
c      lagged convolution is selected when nb>1, which defines arg(nb)
c    over any desired transform argument range (bmin,bmax). results are
c    stored at the filter spacing in arrays arg(nb) and dans(nb,nrel)
c    for later use in spline interpolation, etc.  given bmax,nb, the
c    value of bmin (not given) can be computed from the expression
c    bmin=bmax*dexp(-.1d0*(nb-1)), which must be .gt. 0.0d0 for the
c    given machine exponent range.
c      related convolution is selected when nrel>1, and by given
c    simple algebraic relationships between fun1 and each nrel related
c    transform input function, defined as g**i * fun1(g)**j, where
c    fun1(g) is the first transform input function, and array
c    ijrel(2,nrel)= pairs of i,j integers (negative, 0, or positive).
c    the order of all related convolutions must be given in nord(nrel),
c    and must be either 0 or 1, but can be in any desired sequence.
c    higher integer orders may be expressed in terms of orders 0 and 1
c    by using the recurrence relation jn-1(x)+jn+1(x)=2*n*jn(x)/x.
c      the equally-spaced j0,j1 filter abscissas are generated in
c    double-precision (to conserve storage) and are used as arguments
c    in the external double precision function fun1.
c    both j0 and j1 filter response functions (weights) were
```

```
c    designed to have identical abscissa values as in [2].  the j0,j1
c    double-precision filter weights used in subprogram dhankl were
c    redesigned from [2] in quadruple-precision using a vax-11/780 in
c    h-floating arithmetic.  much of the logic used in dhankl follows
c    the coding used in [1] and [2].  the major differences are in the
c    definition of related input functions (see ijrel,dwork), and
c    for handling oscillating functions (see ntol,itol).
c
c      four general cases are possible using subprogram dhankl.
c
c    case 1.  single direct convolution at b=bmax=bmin (nb=1,nrel=1).
c    case 2.  related convolutions at a constant b=bmax (nb=1,nrel>1).
c    case 3.  lagged convolutions in (bmin,bmax) (nb>1,nrel=1).
c    case 4.  both related and lagged convolutions (nb>1,nrel>1).
c
c  machine dependent remarks
c
c      this subprogram was implemented and tested on a 64-bit double-
c    word machine with exp-range approximately 10**-38 to 10**+38 and
c    56-bit mantissa (about 16-decimal digits).  only double-precision
c    and integer operations are used.
c    for machines with other double-word sizes, changes in the number of
c    digits retained in some data statements may be required.
c
c  description of parameters
c
c    input
c
c      bmax - initial hankel transform argument b=bmax>0.d0 (any case),
c             used in integral from 0 to infinity of
c             fun1(g)*jn(g*b)*dg, where jn=bessel function of order n,
c             n=0 or 1, and b>0.0d0. (see fun1 definition below).
c        nb - number of lagged convolutions desired (nb.ge.1).  use
c             nb=1 if b=bmin=bmax (i.e., case 1 or 2).  use
c             nb>1 if b is lagged in (bmin,bmax), where
c             bmin=bmax*dexp(-.1d0*(nb-1)) does not underflow the dexp
c             range.  the b-lagged spacing is .1d0 in log-space. for
c             convenience in spline interpolation later, each b in
c             (bmin,bmax) is returned in array arg(i),i=1,nb, where
c             arg(i+1)/arg(i)=dexp(.1d0) for all i.  if bmax>bmin>0 is
c             given, then an effective value of nb is determined as
c             nb=dint(10.*dlog(bmax/bmin))+i, where i>1 is recommended,
c             particularly if using subsequent spline interpolation for
```

```
c           a different b-spacing than used in the sampled filters. if
c           spline interpolation is to be used later, it is generally
c           best to use dlog(arg(i)) instead of arg(i) -vs- dans(i,j),
c           for i=1,nb, and for any given j between 1 and nrel.  note
c           nb is used as an adjustable dimension in dans(nb,nrel).
c     nrel - number of related convolutions desired (nrel.ge.1).  use
c           nrel=1 if only a single hankel transform is used.
c           nrel>1 requires array ijrel(2,nrel)  (see below).
c           note nrel is used as adjustable dimensions in arrays
c           dans(nb,nrel),dwork(801,nrel),nord(nrel),ijrel(2,nrel).
c      tol - requested truncation tolerance at both filter tails
c           for adaptive convolution for all nb*nrel transforms.  the
c           truncation criterion is established during convolution in
c           a fixed abscissa range (using weights 299-339) of either
c           order filter as the maximum absolute convolved product
c           times tol.  the convolution summation is terminated
c           on either side of the fixed range whenever the absolute
c           product .le. the truncation criterion.  in general, a
c           decreasing tolerance will produce higher accuracy since
c           more filter weights are used (unless exponent underflow
c           occurs in the transform input function evaluation).
c           one may set tol=0.d0 to obtain maximum accuracy for all
c           nb*nrel real hankel transforms in dans(nb,nrel).
c           however, the actual relative errors cannot be expected to
c           be smaller than about .1d-13 regardless of the tolerance
c           value used, since double-precision filter weights and
c           double-precision functions are used. in any event,
c           one should always choose tol< <desired relative error.
c        ** accuracy warning **  some highly oscillatory functions
c           fun1(g) and (or) limiting cases of b near machine-zero
c           (or infinity) should be avoided, otherwise unsatisfactory
c           results (e.g., relative & absolute errors> >tol) may occur.
c     ntol - number of consecutive times the truncation criterion (tol)
c           is to be met at either filter tail before filter
c           truncation occurs.  ntol=1 should be used for input
c           functions that do not have many zeros in (0,infinity). for
c           oscillatory functions with many zeros, ntol>1 may be used
c           to insure a premature cutoff does not occur for truncation
c           (see use of itol,ntol,tol in the code below).
c     nord - integer array nord(nrel) giving the nrel orders (0 or 1)
c           of each related hankel transform. if any nord(i),i=1,nrel,
c           is not 0 or 1, then order 1 will be assumed.
c     fun1 - name of an external double precision function of a real
```

```
c             argument defining the 1st transform input function of the
c             set of related transforms to be evaluated.  an external
c             fun1 statement must appear in the calling program.  the
c             double precision function fun1(g) must be coded by the
c             user and must be a continuous decreasing real function
c             for all real g>0.0d0.  the value of g must be unchanged
c             upon return from fun1. a multiple-pole of fun1(g) at g=0.0
c             can exist, provided the hankel transform converges (note
c             fun1(0d0) is not used).  generally, fun1(g)
c             is defined analytically for all g>0.d0.  however,
c             discretely defined functions may be used if fun1(g)
c             returns a smooth interpolation value (e.g., via cubic
c             splines) which satisfies the continuity condition for all
c             g>0, and provided the proper limiting value of fun1(g) is
c             given as g tends to infinity.  parameters other
c             than g needed in fun1(g) may be included by using labeled
c             common in fun1 and in the users calling program.  if
c             fun1(g) is an oscillating function, then the highest
c             frequency component (in log-space) should not exceed the
c             filter nyquist frequency, 1/(2*0.1d0).  in general,
c             subprogram dhankl performs best when using smooth, well-
c             behaved functions fun1(g), that are characterized as
c             monotonically decreasing functions with relatively few
c             zeros for g>0. (see the accuracy warning under tol, and
c             error condition (4).)
c     ijrel - integer array ijrel(2,nrel) used when nrel>1 to define
c             the pair of i,j integer exponents for each related input
c             function.  the related input functions are assumed
c             to be simply related in terms of fun1 via the integer
c             array ijrel(2,k),k=2,nrel.  that is, we assume the k-th
c             related input function is given (see statement 160) as
c             funk(g)=g**ijrel(1,k) * fun1(g)**ijrel(2,k), where
c             the integer exponents may be positive, zero, or negative.
c             in this way, only fun1 need be declared an external
c             function. more complicated code could be used
c             for the related functions, provided the meaning of ijrel
c             is redefined and statement 160 is changed (also, see
c             error condition (3) below). when nrel=1, array
c             ijrel is a dummy name (i.e., not referenced).
c             if nrel>1, then the statement at label 160
c             is defined only for k=j=2,...,nrel. that is,
c             ijrel(1,1),ijrel(2,1) are not used in this version.
c     dwork - work array dwork(801,nrel), which is used to
```

```
c             hold various computed functional values during related and
c             lagged convolutions.  a storage roll feature using
c             dwork(801,nrel) and internal array key(801) allows for
c             any b range (bmin,bmax) to be used during convolution.
c
c  output
c
c      dans - the array dans(nb,nrel) is returned giving the
c             nb*nrel real hankel transforms, with corresponding
c             b arguments given in array arg(nb).
c       arg - the array arg(nb) is returned giving the resulting
c             b arguments in (bmin,bmax), where arg(i+1)/arg(i)=exp(.1),
c             i=1,nb-1 (this array could be eliminated to save storage
c             and regenerated after the call dhankl, if desired).
c    nofun1 - number of direct fun1 evaluations used for all nb*nrel
c             real hankel transforms.  nofun1 is usually not more
c             than the number of weights needed for a single direct
c             convolution, for any nb and nrel.
c      ierr - error return code.  the following codes are possible --
c             = 0, no error in input parameters. dans,arg computed.
c             = 1, improper input parameters (i.e., nb<1,nrel<1,bmax<=0,
c             or bmax*dexp(-.1d0*(nb-1))<=0.d0).  dans,arg not computed.
c
c
c  error conditions
c
c    (1)  improper input parameters given (see ierr=1 above).
c    (2)  underflow conditions are possible during convolution, due to
c         the behavior of fun1, value of b in (bmin,bmax), tol, and
c         ntol. exponent and (or) arithmetic underflow traps must return
c         a value of 0.d0 for the computer system being used.  note that
c         underflow may also occur in the users external function
c         fun1(g) for any value of g as set by subprogram dhankl.
c    (3)  an unrecoverable overflow condition can occur in executing
c         statement 160, depending on the value of b in (bmin,bmax),
c         tol, or the integer exponents used in ijrel(2,nrel),nrel>1.
c         in general, extremely small or large values of b should be
c         avoided (see accuracy warning under tol above).  also, in many
c         cases, exponent overflow can be avoided by proper choice of
c         fun1 and the related input function ordering defined by
c         the ijrel signed integer exponents.
c    (4)  undetected errors are possible if fun1 is improperly coded, or
c         does not yield double-precision accuracy, or is not
```

```
c         a continuous decreasing real function for all g>0.d0.
c
c  usage
c
c    subprogram dhankl is called as follows (use numerical values for
c    <expression>, excluding < and >, in declarations) --
c
c      double precision bmax,tol,dwork(801,<nrel>),dans(<nb>,<nrel>),
c     1 arg(<nb>)
c      dimension nord(<nrel>),ijrel(2,<nrel>)
c      external dfun1
cc-----read/load input parameters for dhankl as required
c      ...
c      call dhankl(bmax,nb,nrel,tol,ntol,nord,dfun1,ijrel,dwork,
c     * dans,arg,nofun1,ierr)
c      if(ierr.eq.1) stop
c      ...
c      end
c      double precision function dfun1(g)
c      double precision g
cc-----insert user supplied code for evaluation of dfun1(g),g>0.0d0
c      end
c
c=======================================
      double precision abscis,c,cmax,dsum,e,er,fun1,g,y,y1,wt0,wt1
      dimension key(801),wt0(801),wt1(801)
      external fun1
c-----we define c,cmax for use in the truncation criterion tests,
c     where c is any convolution product and cmax is the maximum
c     convolved product in the fixed abscissa range (see parameter tol).
c-----abscis=base constant for filter abscissa generation
      data abscis/0.7059431685223780d0/
c-----e=dexp(.1d0), er=1.0d0/e (also used in abscissa generation)
      data e/1.1051709180756476 d0/,er/.9048374180359595 73 d0/
c-----wt0(i)=j0 hankel transform filter weights for i=1,801
      data
     *wt0(  1)/ 2.103562053838982d-29/,wt0(  2)/-1.264469361608894d-14/,
     *wt0(  3)/ 4.615731256788567d-14/,wt0(  4)/-2.798703374257668d-14/,
     *wt0(  5)/ 5.465764965410841d-14/,wt0(  6)/-2.652933109928729d-14/,
     *wt0(  7)/ 5.674913434067321d-14/,wt0(  8)/-2.157276828977208d-14/,
     *wt0(  9)/ 5.831846086773976d-14/,wt0( 10)/-1.546589284868783d-14/,
     *wt0( 11)/ 6.057302455652974d-14/,wt0( 12)/-8.502531259083065d-15/,
     *wt0( 13)/ 6.388018061147645d-14/,wt0( 14)/-5.659657635010288d-16/,
*wt0( 15)/ 6.848500604791407d-14/,wt0( 16)/ 8.572897732168277d-15/,
*wt0( 17)/ 7.465068154681814d-14/,wt0( 18)/ 1.920837293261338d-14/,
*wt0( 19)/ 8.269345428975771d-14/,wt0( 20)/ 3.170116562922900d-14/,
*wt0( 21)/ 9.300004039695208d-14/,wt0( 22)/ 4.649069639417992d-14/,
*wt0( 23)/ 1.060441944490564d-13/,wt0( 24)/ 6.411216589597457d-14/,
*wt0( 25)/ 1.224060834001701d-13/,wt0( 26)/ 8.521776751507022d-14/,
*wt0( 27)/ 1.427957940487172d-13/,wt0( 28)/ 1.106026606968463d-13/,
*wt0( 29)/ 1.680820203098405d-13/,wt0( 30)/ 1.412367028159546d-13/,
*wt0( 31)/ 1.993271011776308d-13/,wt0( 32)/ 1.783032042964119d-13/,
*wt0( 33)/ 2.378298196799422d-13/,wt0( 34)/ 2.232462602765097d-13/,
*wt0( 35)/ 2.851776817107034d-13/,wt0( 36)/ 2.778285575785995d-13/,
*wt0( 37)/ 3.433107735233557d-13/,wt0( 38)/ 3.442019764139949d-13/
 data
*wt0( 39)/ 4.145997612327839d-13/,wt0( 40)/ 4.249938198224969d-13/,
*wt0( 41)/ 5.019411631949951d-13/,wt0( 42)/ 5.234121310773422d-13/,
*wt0( 43)/ 6.088737193247560d-13/,wt0( 44)/ 6.433743253810275d-13/,
*wt0( 45)/ 7.397205280796933d-13/,wt0( 46)/ 7.896642978865920d-13/,
*wt0( 47)/ 8.997626559623209d-13/,wt0( 48)/ 9.681243129571424d-13/,
*wt0( 49)/ 1.095451187471558d-12/,wt0( 50)/ 1.185889375491455d-12/,
*wt0( 51)/ 1.334666226164323d-12/,wt0( 52)/ 1.451673490118585d-12/,
*wt0( 53)/ 1.627033241780647d-12/,wt0( 54)/ 1.776119296526221d-12/,
*wt0( 55)/ 1.984309459865773d-12/,wt0( 56)/ 2.172225112712636d-12/,
*wt0( 57)/ 2.420855801356264d-12/,wt0( 58)/ 2.655866524621122d-12/,
*wt0( 59)/ 2.954213313000758d-12/,wt0( 60)/ 3.246433455110429d-12/,
*wt0( 61)/ 3.605807223054322d-12/,wt0( 62)/ 3.967608279821300d-12/,
*wt0( 63)/ 4.401806878720721d-12/,wt0( 64)/ 4.848316218220904d-12/,
*wt0( 65)/ 5.374176077884770d-12/,wt0( 66)/ 5.923886142128417d-12/,
*wt0( 67)/ 6.561955948855011d-12/,wt0( 68)/ 7.237468388830243d-12/,
*wt0( 69)/ 8.012831864792648d-12/,wt0( 70)/ 8.841766480402105d-12/,
*wt0( 71)/ 9.785047278800097d-12/,wt0( 72)/ 1.080115224902499d-11/,
*wt0( 73)/ 1.194974128877562d-11/,wt0( 74)/ 1.319424925552153d-11/,
*wt0( 75)/ 1.459380374689331d-11/,wt0( 76)/ 1.611708818260074d-11/
 data
*wt0( 77)/ 1.782336249944076d-11/,wt0( 78)/ 1.968696083966218d-11/,
*wt0( 79)/ 2.176804271234846d-11/,wt0( 80)/ 2.404712745375440d-11/,
*wt0( 81)/ 2.658616922424555d-11/,wt0( 82)/ 2.937256616666064d-11/,
*wt0( 83)/ 3.247112071587450d-11/,wt0( 84)/ 3.587699548548411d-11/,
*wt0( 85)/ 3.965909071112399d-11/,wt0( 86)/ 4.382145152220635d-11/,
*wt0( 87)/ 4.843856688602439d-11/,wt0( 88)/ 5.352476425684008d-11/,
*wt0( 89)/ 5.916190912377549d-11/,wt0( 90)/ 6.537635327328955d-11/,
*wt0( 91)/ 7.225949098391667d-11/,wt0( 92)/ 7.985185650562155d-11/,
*wt0( 93)/ 8.825697213255412d-11/,wt0( 94)/ 9.753221923111176d-11/,
*wt0( 95)/ 1.077963949370144d-10/,wt0( 96)/ 1.191270094182892d-10/,
*wt0( 97)/ 1.316619519054340d-10/,wt0( 98)/ 1.455028951566729d-10/,
*wt0( 99)/ 1.608114581091983d-10/,wt0(100)/ 1.777184270673645d-10/,
*wt0(101)/ 1.964147916871289d-10/,wt0(102)/ 2.170665216346822d-10/,
*wt0(103)/ 2.399008451839042d-10/,wt0(104)/ 2.651263504640323d-10/,
*wt0(105)/ 2.930148720448538d-10/,wt0(106)/ 3.238267179640594d-10/,
*wt0(107)/ 3.578885297833856d-10/,wt0(108)/ 3.955234710219298d-10/,
*wt0(109)/ 4.371254308993604d-10/,wt0(110)/ 4.830940473937580d-10/,
*wt0(111)/ 5.339056350072159d-10/,wt0(112)/ 5.900529573690070d-10/,
*wt0(113)/ 6.521132758098962d-10/,wt0(114)/ 7.206928333934838d-10/
 data
*wt0(115)/ 7.964924450372341d-10/,wt0(116)/ 8.802567084675067d-10/,
*wt0(117)/ 9.728375895186283d-10/,wt0(118)/ 1.075148437456224d-09/,
*wt0(119)/ 1.188226062693121d-09/,wt0(120)/ 1.313189706258061d-09/,
*wt0(121)/ 1.451302163665563d-09/,wt0(122)/ 1.603933943511600d-09/,
*wt0(123)/ 1.772624063293560d-09/,wt0(124)/ 1.959049733219870d-09/,
*wt0(125)/ 2.165087540667258d-09/,wt0(126)/ 2.392789115986871d-09/,
*wt0(127)/ 2.644443536014764d-09/,wt0(128)/ 2.922559573439235d-09/,
*wt0(129)/ 3.229930291248597d-09/,wt0(130)/ 3.569622651576204d-09/,
*wt0(131)/ 3.945045448172924d-09/,wt0(132)/ 4.359947261255963d-09/,
*wt0(133)/ 4.818489091363688d-09/,wt0(134)/ 5.325251901762953d-09/,
*wt0(135)/ 5.885315583343685d-09/,wt0(136)/ 6.504277635547376d-09/,
*wt0(137)/ 7.188340419242570d-09/,wt0(138)/ 7.944342903082902d-09/,
*wt0(139)/ 8.779858562959109d-09/,wt0(140)/ 9.703242578022093d-09/,
*wt0(141)/ 1.072374322768940d-08/,wt0(142)/ 1.185156747840095d-08/,
*wt0(143)/ 1.309800933225301d-08/,wt0(144)/ 1.447553742402158d-08/,
*wt0(145)/ 1.599794451372024d-08/,wt0(146)/ 1.768046154055318d-08/,
*wt0(147)/ 1.953993335487118d-08/,wt0(148)/ 2.159496468450500d-08/,
*wt0(149)/ 2.386612830616206d-08/,wt0(150)/ 2.637614961034523d-08/,
*wt0(151)/ 2.915015476269831d-08/,wt0(152)/ 3.221590205565798d-08/
 data
*wt0(153)/ 3.560407926098509d-08/,wt0(154)/ 3.934859178954436d-08/,
*wt0(155)/ 4.348692045365787d-08/,wt0(156)/ 4.806047869438007d-08/,
*wt0(157)/ 5.311504443749314d-08/,wt0(158)/ 5.870120138001748d-08/,
*wt0(159)/ 6.487486163571268d-08/,wt0(160)/ 7.169780940885934d-08/,
*wt0(161)/ 7.923833480505012d-08/,wt0(162)/ 8.757190229426675d-08/,
*wt0(163)/ 9.678192055835553d-08/,wt0(164)/ 1.069605631204864d-07/,
*wt0(165)/ 1.182097045925483d-07/,wt0(166)/ 1.306419269237675d-07/,
*wt0(167)/ 1.443816591198488d-07/,wt0(168)/ 1.595664099835792d-07/,
*wt0(169)/ 1.763481565722233d-07/,wt0(170)/ 1.948948533650376d-07/,
*wt0(171)/ 2.153921247351847d-07/,wt0(172)/ 2.380451115468297d-07/,
*wt0(173)/ 2.630805351444854d-07/,wt0(174)/ 2.907489558998551d-07/,
*wt0(175)/ 3.213272911558462d-07/,wt0(176)/ 3.551215767529809d-07/,
*wt0(177)/ 3.924700396067688d-07/,wt0(178)/ 4.337464734078393d-07/,
*wt0(179)/ 4.793639887921159d-07/,wt0(180)/ 5.297791390370199d-07/,
*wt0(181)/ 5.854964979982190d-07/,wt0(182)/ 6.470737017046583d-07/,
*wt0(183)/ 7.151270374758372d-07/,wt0(184)/ 7.903376040582035d-07/,
*wt0(185)/ 8.734581359370587d-07/,wt0(186)/ 9.653205295304359d-07/,
*wt0(187)/ 1.066844176299330d-06/,wt0(188)/ 1.179045157323447d-06/,
*wt0(189)/ 1.303046419382764d-06/,wt0(190)/ 1.440089007182073d-06/
 data
*wt0(191)/ 1.591544490556806d-06/,wt0(192)/ 1.758928685376697d-06/,
*wt0(193)/ 1.943916830388476d-06/,wt0(194)/ 2.148360347694608d-06/,
*wt0(195)/ 2.374305378111399d-06/,wt0(196)/ 2.624013254094558d-06/,
*wt0(197)/ 2.899983137292810d-06/,wt0(198)/ 3.204977025773239d-06/,
*wt0(199)/ 3.542047402097807d-06/,wt0(200)/ 3.914567778667102d-06/,
*wt0(201)/ 4.326266465748067d-06/,wt0(202)/ 4.781263881007758d-06/,
*wt0(203)/ 5.284113792545945d-06/,wt0(204)/ 5.839848890150422d-06/,
*wt0(205)/ 6.454031158395685d-06/,wt0(206)/ 7.132807538712648d-06/,
*wt0(207)/ 7.882971454044484d-06/,wt0(208)/ 8.712030795792633d-06/,
*wt0(209)/ 9.628283069078871d-06/,wt0(210)/ 1.064089843325841d-05/,
*wt0(211)/ 1.176001148348469d-05/,wt0(212)/ 1.299682267761921d-05/,
*wt0(213)/ 1.436371043746991d-05/,wt0(214)/ 1.587435503282041d-05/,
*wt0(215)/ 1.754387550120777d-05/,wt0(216)/ 1.938898096105121d-05/,
*wt0(217)/ 2.142813784487508d-05/,wt0(218)/ 2.368175471430264d-05/,
*wt0(219)/ 2.617238651818130d-05/,wt0(220)/ 2.892496032668667d-05/,
*wt0(221)/ 3.196702481167932d-05/,wt0(222)/ 3.532902595927333d-05/,
*wt0(223)/ 3.904461178455365d-05/,wt0(224)/ 4.315096908756330d-05/,
*wt0(225)/ 4.768919563102351d-05/,wt0(226)/ 5.270471145419851d-05/,
*wt0(227)/ 5.824771344935570d-05/,wt0(228)/ 6.437367773954276d-05/
 data
*wt0(229)/ 7.114391489568718d-05/,wt0(230)/ 7.862618353777022d-05/,
*wt0(231)/ 8.689536847209423d-05/,wt0(232)/ 9.603423013681158d-05/,
*wt0(233)/ 1.061342328695059d-04/,wt0(234)/ 1.172964602657162d-04/,
*wt0(235)/ 1.296326268075215d-04/,wt0(236)/ 1.432661958546299d-04/,
*wt0(237)/ 1.583336152150400d-04/,wt0(238)/ 1.749856826064412d-04/,
*wt0(239)/ 1.933890547221590d-04/,wt0(240)/ 2.137279149064782d-04/,
*wt0(241)/ 2.362058162174614d-04/,wt0(242)/ 2.610477181410775d-04/,
*wt0(243)/ 2.885022375064333d-04/,wt0(244)/ 3.188441357870562d-04/,
*wt0(245)/ 3.523770680013592d-04/,wt0(246)/ 3.894366200728159d-04/,
*wt0(247)/ 4.303936656695837d-04/,wt0(248)/ 4.756580748770003d-04/,
*wt0(249)/ 5.256828130316798d-04/,wt0(250)/ 5.809684683619367d-04/,
*wt0(251)/ 6.420682561143772d-04/,wt0(252)/ 7.095935446942389d-04/,
*wt0(253)/ 7.842199637491256d-04/,wt0(254)/ 8.666941465924691d-04/,
*wt0(255)/ 9.578411834759010d-04/,wt0(256)/ 1.058572843543458d-03/,
*wt0(257)/ 1.169896665384001d-03/,wt0(258)/ 1.292925974983324d-03/,
*wt0(259)/ 1.428890965739396d-03/,wt0(260)/ 1.579150889627630d-03/,
*wt0(261)/ 1.745207548623177d-03/,wt0(262)/ 1.928720102656342d-03/,
*wt0(263)/ 2.131521473093367d-03/,wt0(264)/ 2.355636277673631d-03/,
*wt0(265)/ 2.603300731229735d-03/,wt0(266)/ 2.876984272542283d-03/
 data
*wt0(267)/ 3.179413630050106d-03/,wt0(268)/ 3.513598723334281d-03/,
*wt0(269)/ 3.882861625673649d-03/,wt0(270)/ 4.290867254031690d-03/,
*wt0(271)/ 4.741657973486860d-03/,wt0(272)/ 5.239689338467119d-03/,
*wt0(273)/ 5.789870985188969d-03/,wt0(274)/ 6.397607072000340d-03/,
*wt0(275)/ 7.068843783102744d-03/,wt0(276)/ 7.810112796625723d-03/,
*wt0(277)/ 8.628584975720666d-03/,wt0(278)/ 9.532112534855741d-03/,
*wt0(279)/ 1.052928697105230d-02/,wt0(280)/ 1.162947042684567d-02/,
*wt0(281)/ 1.284285301027502d-02/,wt0(282)/ 1.418045400408490d-02/,
*wt0(283)/ 1.565416842684913d-02/,wt0(284)/ 1.727670025860997d-02/,
*wt0(285)/ 1.906157878467033d-02/,wt0(286)/ 2.102295173845023d-02/,
*wt0(287)/ 2.317553622087861d-02/,wt0(288)/ 2.553413681316817d-02/,
*wt0(289)/ 2.811347056705691d-02/,wt0(290)/ 3.092716134472897d-02/,
*wt0(291)/ 3.398734108220733d-02/,wt0(292)/ 3.730266904318316d-02/,
*wt0(293)/ 4.087756584601726d-02/,wt0(294)/ 4.470845458227711d-02/,
*wt0(295)/ 4.878245655564208d-02/,wt0(296)/ 5.307046388837388d-02/,
*wt0(297)/ 5.752521479656082d-02/,wt0(298)/ 6.206889249144453d-02/,
*wt0(299)/ 6.659098690582325d-02/,wt0(300)/ 7.092687097703928d-02/,
*wt0(301)/ 7.485762155329997d-02/,wt0(302)/ 7.807466421172765d-02/,
*wt0(303)/ 8.018887211338042d-02/,wt0(304)/ 8.067640670918657d-02/
 data
*wt0(305)/ 7.891767306422777d-02/,wt0(306)/ 7.412406301630496d-02/,
*wt0(307)/ 6.545864753141331d-02/,wt0(308)/ 5.195771725733346d-02/,
*wt0(309)/ 3.284797274184859d-02/,wt0(310)/ 7.497076325831270d-03/,
*wt0(311)/-2.386612869894549d-02/,wt0(312)/-6.017494378476118d-02/,
*wt0(313)/-9.817899798850670d-02/,wt0(314)/-1.328147797272611d-01/,
*wt0(315)/-1.554628569772562d-01/,wt0(316)/-1.563982157987450d-01/,
*wt0(317)/-1.243049866529050d-01/,wt0(318)/-5.486815986343697d-02/,
*wt0(319)/ 4.686255899170307d-02/,wt0(320)/ 1.511218295806206d-01/,
*wt0(321)/ 2.119315534410599d-01/,wt0(322)/ 1.695134135887796d-01/,
*wt0(323)/ 1.386197420331480d-02/,wt0(324)/-1.869350451838135d-01/,
*wt0(325)/-2.455889606925336d-01/,wt0(326)/-5.309269401899814d-02/,
*wt0(327)/ 2.519998415798595d-01/,wt0(328)/ 1.968224876057428d-01/,
*wt0(329)/-2.014333618969160d-01/,wt0(330)/-2.458450827258603d-01/,
*wt0(331)/ 3.433559314076636d-01/,wt0(332)/-4.770066510626292d-02/,
*wt0(333)/-2.096628473507452d-01/,wt0(334)/ 2.509085123794248d-01/,
*wt0(335)/-1.676412661335192d-01/,wt0(336)/ 7.495199786896762d-02/,
*wt0(337)/-1.695135102751146d-02/,wt0(338)/-8.828524201374996d-03/,
*wt0(339)/ 1.616787356183550d-02/,wt0(340)/-1.556411755992529d-02/,
*wt0(341)/ 1.251311125216370d-02/,wt0(342)/-9.299315246978098d-03/
```

```
 data
*wt0(343)/ 6.650525996779493d-03/,wt0(344)/-4.667707408671884d-03/,
*wt0(345)/ 3.248885545556855d-03/,wt0(346)/-2.255529108544675d-03/,
*wt0(347)/ 1.566875211100915d-03/,wt0(348)/-1.091132486368091d-03/,
*wt0(349)/ 7.625383049040949d-04/,wt0(350)/-5.352540577820408d-04/,
*wt0(351)/ 3.777104567214670d-04/,wt0(352)/-2.682528211535297d-04/,
*wt0(353)/ 1.920265274996755d-04/,wt0(354)/-1.388213723208944d-04/,
*wt0(355)/ 1.015998917810005d-04/,wt0(356)/-7.549725062631259d-05/,
*wt0(357)/ 5.714172920052936d-05/,wt0(358)/-4.419160967680634d-05/,
*wt0(359)/ 3.501776199578583d-05/,wt0(360)/-2.848496954139990d-05/,
*wt0(361)/ 2.380104817324115d-05/,wt0(362)/-2.041274924032166d-05/,
*wt0(363)/ 1.793363489791860d-05/,wt0(364)/-1.609367780777977d-05/,
*wt0(365)/ 1.470394692017765d-05/,wt0(366)/-1.363208540087839d-05/,
*wt0(367)/ 1.278540851480116d-05/,wt0(368)/-1.209910323276592d-05/,
*wt0(369)/ 1.152780746372085d-05/,wt0(370)/-1.103964249967709d-05/,
*wt0(371)/ 1.061214724478712d-05/,wt0(372)/-1.022954672255804d-05/,
*wt0(373)/ 9.880801917269063d-06/,wt0(374)/-9.558136268373164d-06/,
*wt0(375)/ 9.255989462874843d-06/,wt0(376)/-8.970368039834961d-06/,
*wt0(377)/ 8.698439226217636d-06/,wt0(378)/-8.438199899378174d-06/,
*wt0(379)/ 8.188185039708096d-06/,wt0(380)/-7.947275976752668d-06/
 data
*wt0(381)/ 7.714620489277084d-06/,wt0(382)/-7.489590334670121d-06/,
*wt0(383)/ 7.271712845563131d-06/,wt0(384)/-7.060595202720040d-06/,
*wt0(385)/ 6.855890958634290d-06/,wt0(386)/-6.657307826365352d-06/,
*wt0(387)/ 6.464610635947412d-06/,wt0(388)/-6.277596705248834d-06/,
*wt0(389)/ 6.096069004564178d-06/,wt0(390)/-5.919835014912806d-06/,
*wt0(391)/ 5.748721389173238d-06/,wt0(392)/-5.582576428368341d-06/,
*wt0(393)/ 5.421255729073172d-06/,wt0(394)/-5.264611483233620d-06/,
*wt0(395)/ 5.112497827757443d-06/,wt0(396)/-4.964780650195476d-06/,
*wt0(397)/ 4.821336526159266d-06/,wt0(398)/-4.682043530422997d-06/,
*wt0(399)/ 4.546777587820041d-06/,wt0(400)/-4.415418000076004d-06/,
*wt0(401)/ 4.287852503112911d-06/,wt0(402)/-4.163974668109847d-06/,
*wt0(403)/ 4.043678407829827d-06/,wt0(404)/-3.926857526517075d-06/,
*wt0(405)/ 3.813409851360361d-06/,wt0(406)/-3.703239239834851d-06/,
*wt0(407)/ 3.596252927182967d-06/,wt0(408)/-3.492358600906126d-06/,
*wt0(409)/ 3.391465179914198d-06/,wt0(410)/-3.293485431241940d-06/,
*wt0(411)/ 3.198336356446399d-06/,wt0(412)/-3.105937110348406d-06/,
*wt0(413)/ 3.016207679069661d-06/,wt0(414)/-2.929069911562291d-06/,
*wt0(415)/ 2.844448954094881d-06/,wt0(416)/-2.762272935820411d-06/,
*wt0(417)/ 2.682471568415877d-06/,wt0(418)/-2.604975709140556d-06/
 data
*wt0(419)/ 2.529718233664062d-06/,wt0(420)/-2.456634690985054d-06/,
*wt0(421)/ 2.385662793855871d-06/,wt0(422)/-2.316741584624232d-06/,
*wt0(423)/ 2.249811415880699d-06/,wt0(424)/-2.184814552602377d-06/,
*wt0(425)/ 2.121695378987067d-06/,wt0(426)/-2.060399926949900d-06/,
*wt0(427)/ 2.000875435893980d-06/,wt0(428)/-1.943070483347481d-06/,
*wt0(429)/ 1.886935339579619d-06/,wt0(430)/-1.832421953831803d-06/,
*wt0(431)/ 1.779483599068986d-06/,wt0(432)/-1.728074671861562d-06/,
*wt0(433)/ 1.678150840818710d-06/,wt0(434)/-1.629669220867866d-06/,
*wt0(435)/ 1.582588274939725d-06/,wt0(436)/-1.536867577717458d-06/,
*wt0(437)/ 1.492467744182388d-06/,wt0(438)/-1.449350544031671d-06/,
*wt0(439)/ 1.407478963215944d-06/,wt0(440)/-1.366817091697906d-06/,
*wt0(441)/ 1.327329980403758d-06/,wt0(442)/-1.288983628354532d-06/,
*wt0(443)/ 1.251745052743577d-06/,wt0(444)/-1.215582289623042d-06/,
*wt0(445)/ 1.180464297731392d-06/,wt0(446)/-1.146360878232118d-06/,
*wt0(447)/ 1.113242681755176d-06/,wt0(448)/-1.081081241360643d-06/,
*wt0(449)/ 1.049848946307979d-06/,wt0(450)/-1.019518969919736d-06/,
*wt0(451)/ 9.900652269648073d-07/,wt0(452)/-9.614623823829486d-07/,
*wt0(453)/ 9.336858590396268d-07/,wt0(454)/-9.067118039187765d-07/,
*wt0(455)/ 8.805170379415897d-07/,wt0(456)/-8.550790335506564d-07/
 data
*wt0(457)/ 8.303759185665738d-07/,wt0(458)/-8.063864695073603d-07/,
*wt0(459)/ 7.830900795968434d-07/,wt0(460)/-7.604667251660013d-07/,
*wt0(461)/ 7.384969528074763d-07/,wt0(462)/-7.171618775580271d-07/,
*wt0(463)/ 6.964431695613518d-07/,wt0(464)/-6.763230273695452d-07/,
*wt0(465)/ 6.567841555251206d-07/,wt0(466)/-6.378097555683096d-07/,
*wt0(467)/ 6.193835199757372d-07/,wt0(468)/-6.014896169720738d-07/,
*wt0(469)/ 5.841126695566173d-07/,wt0(470)/-5.672377400509635d-07/,
*wt0(471)/ 5.508503223216395d-07/,wt0(472)/-5.349363333399711d-07/,
*wt0(473)/ 5.194820986824491d-07/,wt0(474)/-5.044743364413132d-07/,
*wt0(475)/ 4.899001458890472d-07/,wt0(476)/-4.757469997761242d-07/,
*wt0(477)/ 4.620027360551677d-07/,wt0(478)/-4.486555431391317d-07/,
*wt0(479)/ 4.356939495368206d-07/,wt0(480)/-4.231068139067277d-07/,
*wt0(481)/ 4.108833178901058d-07/,wt0(482)/-3.990129570237777d-07/,
*wt0(483)/ 3.874855299952715d-07/,wt0(484)/-3.762911289578004d-07/,
*wt0(485)/ 3.654201320223786d-07/,wt0(486)/-3.548631961765860d-07/,
*wt0(487)/ 3.446112489455190d-07/,wt0(488)/-3.346554794873414d-07/,
*wt0(489)/ 3.249873307934947d-07/,wt0(490)/-3.155984931421240d-07/,
*wt0(491)/ 3.064808974928670d-07/,wt0(492)/-2.976267081233949d-07/,
*wt0(493)/ 2.890283152696976d-07/,wt0(494)/-2.806783286653362d-07/
 data
*wt0(495)/ 2.725695717362821d-07/,wt0(496)/-2.646950756050221d-07/,
*wt0(497)/ 2.570480727284082d-07/,wt0(498)/-2.496219907786807d-07/,
*wt0(499)/ 2.424104471687142d-07/,wt0(500)/-2.354072438896643d-07/,
*wt0(501)/ 2.286063621819694d-07/,wt0(502)/-2.220019570969968d-07/,
*wt0(503)/ 2.155883523600859d-07/,wt0(504)/-2.093600356601861d-07/,
*wt0(505)/ 2.033116540781646d-07/,wt0(506)/-1.974380094205501d-07/,
*wt0(507)/ 1.917340535876188d-07/,wt0(508)/-1.861948842172547d-07/,
*wt0(509)/ 1.808157405985162d-07/,wt0(510)/-1.755919996494557d-07/,
*wt0(511)/ 1.705191718700136d-07/,wt0(512)/-1.655928973946416d-07/,
*wt0(513)/ 1.608089422670497d-07/,wt0(514)/-1.561631948835526d-07/,
*wt0(515)/ 1.516516624771349d-07/,wt0(516)/-1.472704676262496d-07/,
*wt0(517)/ 1.430158448819191d-07/,wt0(518)/-1.388841375627619d-07/,
*wt0(519)/ 1.348717946589134d-07/,wt0(520)/-1.309753677747614d-07/,
*wt0(521)/ 1.271915081246862d-07/,wt0(522)/-1.235169636417206d-07/,
*wt0(523)/ 1.199485762099940d-07/,wt0(524)/-1.164832789732080d-07/,
*wt0(525)/ 1.131180936862105d-07/,wt0(526)/-1.098501281312535d-07/,
*wt0(527)/ 1.066765736321394d-07/,wt0(528)/-1.035947026599537d-07/,
*wt0(529)/ 1.006018664975167d-07/,wt0(530)/-9.769549295058651d-08/,
*wt0(531)/ 9.487308412474974d-08/,wt0(532)/-9.213221428331440d-08/
 data
*wt0(533)/ 8.947052777472484d-08/,wt0(534)/-8.688573700943115d-08/,
*wt0(535)/ 8.437562048443172d-08/,wt0(536)/-8.193802086874367d-08/,
*wt0(537)/ 7.957084315467929d-08/,wt0(538)/-7.727205286392824d-08/,
*wt0(539)/ 7.503967429747869d-08/,wt0(540)/-7.287178883148888d-08/,
*wt0(541)/ 7.076653326692979d-08/,wt0(542)/-6.872209823264986d-08/,
*wt0(543)/ 6.673672663335082d-08/,wt0(544)/-6.480871213716780d-08/,
*wt0(545)/ 6.293639770567428d-08/,wt0(546)/-6.111817417008917d-08/,
*wt0(547)/ 5.935247885127626d-08/,wt0(548)/-5.763779421770836d-08/,
*wt0(549)/ 5.597264657919956d-08/,wt0(550)/-5.435560481863042d-08/,
*wt0(551)/ 5.278527916288309d-08/,wt0(552)/-5.126031999018991d-08/,
*wt0(553)/ 4.977941667022704d-08/,wt0(554)/-4.834129643623350d-08/,
*wt0(555)/ 4.694472329047102d-08/,wt0(556)/-4.558849694284832d-08/,
*wt0(557)/ 4.427145178025873d-08/,wt0(558)/-4.299245586445184d-08/,
*wt0(559)/ 4.175040995827635d-08/,wt0(560)/-4.054424658081963d-08/,
*wt0(561)/ 3.937292909065092d-08/,wt0(562)/-3.823545079527342d-08/,
*wt0(563)/ 3.713083408552377d-08/,wt0(564)/-3.605812959486601d-08/,
*wt0(565)/ 3.501641538356351d-08/,wt0(566)/-3.400479614676831d-08/,
*wt0(567)/ 3.302240244515663d-08/,wt0(568)/-3.206838995736202d-08/,
*wt0(569)/ 3.114193875409117d-08/,wt0(570)/-3.024225259359901d-08/
 data
*wt0(571)/ 2.936855823761587d-08/,wt0(572)/-2.852010478676213d-08/,
*wt0(573)/ 2.769616303496246d-08/,wt0(574)/-2.689602484264953d-08/,
*wt0(575)/ 2.611900252829978d-08/,wt0(576)/-2.536442827753140d-08/,
*wt0(577)/ 2.463165356908288d-08/,wt0(578)/-2.392004861730570d-08/,
*wt0(579)/ 2.322900183088905d-08/,wt0(580)/-2.255791928733344d-08/,
*wt0(581)/ 2.190622422255186d-08/,wt0(582)/-2.127335653510145d-08/,
*wt0(583)/ 2.065877230473273d-08/,wt0(584)/-2.006194332493974d-08/,
*wt0(585)/ 1.948235664905881d-08/,wt0(586)/-1.891951414942437d-08/,
*wt0(587)/ 1.837293208920017d-08/,wt0(588)/-1.784214070660012d-08/,
*wt0(589)/ 1.732668381117887d-08/,wt0(590)/-1.682611839179446d-08/,
*wt0(591)/ 1.634001423585344d-08/,wt0(592)/-1.586795355952947d-08/,
*wt0(593)/ 1.540953064868929d-08/,wt0(594)/-1.496435151022419d-08/,
*wt0(595)/ 1.453203353344868d-08/,wt0(596)/-1.411220516125227d-08/,
*wt0(597)/ 1.370450557074419d-08/,wt0(598)/-1.330858436314438d-08/,
*wt0(599)/ 1.292410126264865d-08/,wt0(600)/-1.255072582398340d-08/,
*wt0(601)/ 1.218813714839246d-08/,wt0(602)/-1.183602360782992d-08/,
*wt0(603)/ 1.149408257713515d-08/,wt0(604)/-1.116202017395058d-08/,
*wt0(605)/ 1.083955100614394d-08/,wt0(606)/-1.052639792651900d-08/,
*wt0(607)/ 1.022229179461678d-08/,wt0(608)/-9.926971245406734d-09/
 data
*wt0(609)/ 9.640182464660921d-09/,wt0(610)/-9.361678970810806d-09/,
*wt0(611)/ 9.091221403102823d-09/,wt0(612)/-8.828577315878212d-09/,
*wt0(613)/ 8.573520978800615d-09/,wt0(614)/-8.325833182853652d-09/,
*wt0(615)/ 8.085301051938885d-09/,wt0(616)/-7.851717859915976d-09/,
*wt0(617)/ 7.624882852931747d-09/,wt0(618)/-7.404601076884107d-09/,
*wt0(619)/ 7.190683209868762d-09/,wt0(620)/-6.982945399464102d-09/,
*wt0(621)/ 6.781209104717332d-09/,wt0(622)/-6.585300942697731d-09/,
*wt0(623)/ 6.395052539483568d-09/,wt0(624)/-6.210300385452473d-09/,
*wt0(625)/ 6.030885694751333d-09/,wt0(626)/-5.856654268826800d-09/,
*wt0(627)/ 5.687456363899640d-09/,wt0(628)/-5.523146562267662d-09/,
*wt0(629)/ 5.363583647325650d-09/,wt0(630)/-5.208630482195534d-09/,
*wt0(631)/ 5.058153891863612d-09/,wt0(632)/-4.912024548723426d-09/,
*wt0(633)/ 4.770116861425010d-09/,wt0(634)/-4.632308866934630d-09/,
*wt0(635)/ 4.498482125712853d-09/,wt0(636)/-4.368521619921419d-09/,
*wt0(637)/ 4.242315654571151d-09/,wt0(638)/-4.119755761525383d-09/,
*wt0(639)/ 4.000736606276379d-09/,wt0(640)/-3.885155897415073d-09/,
*wt0(641)/ 3.772914298716528d-09/,wt0(642)/-3.663915343765296d-09/,
*wt0(643)/ 3.558065353046999d-09/,wt0(644)/-3.455273353434971d-09/,
*wt0(645)/ 3.355451000003056d-09/,wt0(646)/-3.258512500097394d-09/
 data
*wt0(647)/ 3.164374539601736d-09/,wt0(648)/-3.072956211332787d-09/,
*wt0(649)/ 2.984178945504153d-09/,wt0(650)/-2.897966442199276d-09/,
*wt0(651)/ 2.814244605795313d-09/,wt0(652)/-2.732941481281439d-09/,
*wt0(653)/ 2.653987192416816d-09/,wt0(654)/-2.577313881675167d-09/,
*wt0(655)/ 2.502855651924429d-09/,wt0(656)/-2.430548509791296d-09/,
*wt0(657)/ 2.360330310661908d-09/,wt0(658)/-2.292140705271406d-09/,
*wt0(659)/ 2.225921087836540d-09/,wt0(660)/-2.161614545686758d-09/,
*wt0(661)/ 2.099165810350450d-09/,wt0(662)/-2.038521210054267d-09/,
*wt0(663)/ 1.979628623594711d-09/,wt0(664)/-1.922437435542387d-09/,
*wt0(665)/ 1.866898492740427d-09/,wt0(666)/-1.812964062059653d-09/,
*wt0(667)/ 1.760587789374176d-09/,wt0(668)/-1.709724659722168d-09/,
*wt0(669)/ 1.660330958617607d-09/,wt0(670)/-1.612364234479730d-09/,
*wt0(671)/ 1.565783262147887d-09/,wt0(672)/-1.520548007450427d-09/,
*wt0(673)/ 1.476619592797189d-09/,wt0(674)/-1.433960263766035d-09/,
*wt0(675)/ 1.392533356654690d-09/,wt0(676)/-1.352303266970005d-09/,
*wt0(677)/ 1.313235418827548d-09/,wt0(678)/-1.275296235235242d-09/,
*wt0(679)/ 1.238453109235514d-09/,wt0(680)/-1.202674375881138d-09/,
*wt0(681)/ 1.167929285020692d-09/,wt0(682)/-1.134187974870230d-09/,
*wt0(683)/ 1.101421446348478d-09/,wt0(684)/-1.069601538153474d-09/
 data
*wt0(685)/ 1.038700902559248d-09/,wt0(686)/-1.008692981911727d-09/,
*wt0(687)/ 9.795519858036667d-10/,wt0(688)/-9.512528689090021d-10/,
*wt0(689)/ 9.237713094575600d-10/,wt0(690)/-8.970836883316330d-10/,
*wt0(691)/ 8.711670687664494d-10/,wt0(692)/-8.459991766370928d-10/,
*wt0(693)/ 8.215583813149323d-10/,wt0(694)/-7.978236770771082d-10/,
*wt0(695)/ 7.747746650530930d-10/,wt0(696)/-7.523915356928136d-10/,
*wt0(697)/ 7.306550517412669d-10/,wt0(698)/-7.095465317049981d-10/,
*wt0(699)/ 6.890478337962292d-10/,wt0(700)/-6.691413403408390d-10/,
*wt0(701)/ 6.498099426367938d-10/,wt0(702)/-6.310370262500165d-10/,
*wt0(703)/ 6.128064567350563d-10/,wt0(704)/-5.951025657682865d-10/,
*wt0(705)/ 5.779101376817111d-10/,wt0(706)/-5.612143963858077d-10/,
*wt0(707)/ 5.450009926701687d-10/,wt0(708)/-5.292559918710250d-10/,
*wt0(709)/ 5.139658618950525d-10/,wt0(710)/-4.991174615891697d-10/,
*wt0(711)/ 4.846980294463296d-10/,wt0(712)/-4.706951726376001d-10/,
*wt0(713)/ 4.570968563611062d-10/,wt0(714)/-4.438913934986793d-10/,
*wt0(715)/ 4.310674345713248d-10/,wt0(716)/-4.186139579848742d-10/,
*wt0(717)/ 4.065202605574395d-10/,wt0(718)/-3.947759483205284d-10/,
*wt0(719)/ 3.833709275859127d-10/,wt0(720)/-3.722953962705749d-10/,
*wt0(721)/ 3.615398354722752d-10/,wt0(722)/-3.510950012884991d-10/
 data
*wt0(723)/ 3.409519168717552d-10/,wt0(724)/-3.311018647143929d-10/,
*wt0(725)/ 3.215363791563124d-10/,wt0(726)/-3.122472391091240d-10/,
*wt0(727)/ 3.032264609905071d-10/,wt0(728)/-2.944662918626933d-10/,
*wt0(729)/ 2.859592027691788d-10/,wt0(730)/-2.776978822639370d-10/,
*wt0(731)/ 2.696752301275724d-10/,wt0(732)/-2.618843512650130d-10/,
*wt0(733)/ 2.543185497794985d-10/,wt0(734)/-2.469713232177689d-10/,
*wt0(735)/ 2.398363569815098d-10/,wt0(736)/-2.329075189002498d-10/,
*wt0(737)/ 2.261788539610461d-10/,wt0(738)/-2.196445791904288d-10/,
*wt0(739)/ 2.132990786842053d-10/,wt0(740)/-2.071368987808525d-10/,
*wt0(741)/ 2.011527433743496d-10/,wt0(742)/-1.953414693624216d-10/,
*wt0(743)/ 1.896980822262837d-10/,wt0(744)/-1.842177317380848d-10/,
*wt0(745)/ 1.788957077923634d-10/,wt0(746)/-1.737274363579317d-10/,
*wt0(747)/ 1.687084755467094d-10/,wt0(748)/-1.638345117961281d-10/,
*wt0(749)/ 1.591013561618254d-10/,wt0(750)/-1.545049407174433d-10/,
      *wt0(751)/ 1.500413150584345d-10/,wt0(752)/-1.457066429068751d-10/,
      *wt0(753)/ 1.414971988143619d-10/,wt0(754)/-1.374093649601637d-10/,
      *wt0(755)/ 1.334396280418735d-10/,wt0(756)/-1.295845762558885d-10/,
      *wt0(757)/ 1.258408963651246d-10/,wt0(758)/-1.222053708514437d-10/,
      *wt0(759)/ 1.186748751503467d-10/,wt0(760)/-1.152463749655566d-10/
       data
      *wt0(761)/ 1.119169236611821d-10/,wt0(762)/-1.086836597292220d-10/,
      *wt0(763)/ 1.055438043302322d-10/,wt0(764)/-1.024946589050432d-10/,
      *wt0(765)/ 9.953360285547502d-11/,wt0(766)/-9.665809129205519d-11/,
      *wt0(767)/ 9.386565284680683d-11/,wt0(768)/-9.115388754922578d-11/,
      *wt0(769)/ 8.852046476362457d-11/,wt0(770)/-8.596312118607507d-11/,
      *wt0(771)/ 8.347965889923978d-11/,wt0(772)/-8.106794348344660d-11/,
      *wt0(773)/ 7.872590218244558d-11/,wt0(774)/-7.645152212241484d-11/,
      *wt0(775)/ 7.424284858301879d-11/,wt0(776)/-7.209798331980772d-11/,
      *wt0(777)/ 7.001508293831718d-11/,wt0(778)/-6.799235732265070d-11/,
      *wt0(779)/ 6.602806812690510d-11/,wt0(780)/-6.412052735069306d-11/,
      *wt0(781)/ 6.226809604991307d-11/,wt0(782)/-6.046918330329692d-11/,
      *wt0(783)/ 5.872224571634798d-11/,wt0(784)/-5.702578811834639d-11/,
      *wt0(785)/ 5.537836697680089d-11/,wt0(786)/-5.377860007122251d-11/,
      *wt0(787)/ 5.222519065363231d-11/,wt0(788)/-5.071698520520750d-11/,
      *wt0(789)/ 4.925310917157278d-11/,wt0(790)/-4.783328375532226d-11/,
      *wt0(791)/ 4.645856316442775d-11/,wt0(792)/-4.513304831495452d-11/,
      *wt0(793)/ 4.386786828115582d-11/,wt0(794)/-4.269042848860555d-11/,
      *wt0(795)/ 4.166589074066043d-11/,wt0(796)/-4.094706131972530d-11/,
      *wt0(797)/ 4.089025606285087d-11/,wt0(798)/-4.232439520086882d-11/
       data
      *wt0(799)/ 4.717597028661840d-11/,wt0(800)/-5.992051472297203d-11/,
      *wt0(801)/ 9.095360729014628d-11/
c-----wt1(i)=j1 hankel transform filter weights for i=1,801
       data
      *wt1(  1)/-2.377900110058238d-29/,wt1(  2)/ 7.972119293423644d-28/,
      *wt1(  3)/ 3.832566186338494d-28/,wt1(  4)/ 8.493403321392027d-28/,
      *wt1(  5)/ 8.232164371695149d-28/,wt1(  6)/ 1.100368717019849d-27/,
      *wt1(  7)/ 1.325428732821688d-27/,wt1(  8)/ 1.592246428628784d-27/,
      *wt1(  9)/ 1.998601965098251d-27/,wt1( 10)/ 2.368546581454427d-27/,
      *wt1( 11)/ 2.979879339217500d-27/,wt1( 12)/ 3.543101964370716d-27/,
      *wt1( 13)/ 4.425741130789444d-27/,wt1( 14)/ 5.318079447058835d-27/,
      *wt1( 15)/ 6.555619006665152d-27/,wt1( 16)/ 7.994776219905923d-27/,
      *wt1( 17)/ 9.707158322300619d-27/,wt1( 18)/ 1.200497993577129d-26/,
      *wt1( 19)/ 1.440665350963159d-26/,wt1( 20)/ 1.797046711269925d-26/,
      *wt1( 21)/ 2.145390397713542d-26/,wt1( 22)/ 2.681614737113612d-26/,
      *wt1( 23)/ 3.203783629081399d-26/,wt1( 24)/ 3.992219752668547d-26/,
      *wt1( 25)/ 4.793780377063212d-26/,wt1( 26)/ 5.935124745092138d-26/,
*wt1( 27)/ 7.177334354418767d-26/,wt1( 28)/ 8.825226769587027d-26/,
*wt1( 29)/ 1.073646008000832d-25/,wt1( 30)/ 1.313923424181352d-25/,
*wt1( 31)/ 1.603838995307793d-25/,wt1( 32)/ 1.958633055912045d-25/,
*wt1( 33)/ 2.393489595056268d-25/,wt1( 34)/ 2.921690713642289d-25/,
*wt1( 35)/ 3.570455561097498d-25/,wt1( 36)/ 4.359157800465477d-25/,
*wt1( 37)/ 5.325840709133780d-25/,wt1( 38)/ 6.503774912184682d-25/
 data
*wt1( 39)/ 7.944617848334684d-25/,wt1( 40)/ 9.702980761516990d-25/,
*wt1( 41)/ 1.185162186264197d-24/,wt1( 42)/ 1.447534903673577d-24/,
*wt1( 43)/ 1.768047044621983d-24/,wt1( 44)/ 2.159464672432506d-24/,
*wt1( 45)/ 2.637627922624120d-24/,wt1( 46)/ 3.221525590342443d-24/,
*wt1( 47)/ 3.934898912364447d-24/,wt1( 48)/ 4.805930563105760d-24/,
*wt1( 49)/ 5.870195744781909d-24/,wt1( 50)/ 7.169601400607034d-24/,
*wt1( 51)/ 8.757287433857846d-24/,wt1( 52)/ 1.069583032926807d-23/,
*wt1( 53)/ 1.306426797568440d-23/,wt1( 54)/ 1.595639689868334d-23/,
*wt1( 55)/ 1.948949017485241d-23/,wt1( 56)/ 2.380425857645875d-23/,
*wt1( 57)/ 2.907479573699533d-23/,wt1( 58)/ 3.551187751264232d-23/,
*wt1( 59)/ 4.337441481530691d-23/,wt1( 60)/ 5.297756660121858d-23/,
*wt1( 61)/ 6.470696377141974d-23/,wt1( 62)/ 7.903329008477021d-23/,
*wt1( 63)/ 9.653141660115531d-23/,wt1( 64)/ 1.179038165380973d-22/,
*wt1( 65)/ 1.440079827014168d-22/,wt1( 66)/ 1.758917653247009d-22/,
*wt1( 67)/ 2.148347376559587d-22/,wt1( 68)/ 2.623996157283884d-22/,
*wt1( 69)/ 3.204958075285148d-22/,wt1( 70)/ 3.914542160210567d-22/,
*wt1( 71)/ 4.781235486108842d-22/,wt1( 72)/ 5.839810952393910d-22/,
*wt1( 73)/ 7.132764830587759d-22/,wt1( 74)/ 8.711974589175415d-22/,
*wt1( 75)/ 1.064083434862305d-21/,wt1( 76)/ 1.299673924385260d-21/
 data
*wt1( 77)/ 1.587425911126261d-21/,wt1( 78)/ 1.938885693339251d-21/,
*wt1( 79)/ 2.368161151141687d-21/,wt1( 80)/ 2.892477586360832d-21/,
*wt1( 81)/ 3.532881273577120d-21/,wt1( 82)/ 4.315069519949315d-21/,
*wt1( 83)/ 5.270439518682773d-21/,wt1( 84)/ 6.437327295395713d-21/,
*wt1( 85)/ 7.862571844418139d-21/,wt1( 86)/ 9.603363814287760d-21/,
*wt1( 87)/ 1.172957893208765d-20/,wt1( 88)/ 1.432653521781635d-20/,
*wt1( 89)/ 1.749847566646075d-20/,wt1( 90)/ 2.137267893871469d-20/,
*wt1( 91)/ 2.610465830070486d-20/,wt1( 92)/ 3.188429016442547d-20/,
*wt1( 93)/ 3.894357412134842d-20/,wt1( 94)/ 4.756577134959383d-20/,
*wt1( 95)/ 5.809698595129943d-20/,wt1( 96)/ 7.095979207053098d-20/,
*wt1( 97)/ 8.667051908059619d-20/,wt1( 98)/ 1.058595695124327d-19/,
*wt1( 99)/ 1.292972220966051d-19/,wt1(100)/ 1.579239186764568d-19/,
*wt1(101)/ 1.928887914036691d-19/,wt1(102)/ 2.355947994119052d-19/,
*wt1(103)/ 2.877562665708650d-19/,wt1(104)/ 3.514661357021616d-19/,
*wt1(105)/ 4.292819113549590d-19/,wt1(106)/ 5.243258540458620d-19/,
*wt1(107)/ 6.404133671732880d-19/,wt1(108)/ 7.822022466379286d-19/,
*wt1(109)/ 9.553844929891425d-19/,wt1(110)/ 1.166908611051965d-18/,
*wt1(111)/ 1.425266206309626d-18/,wt1(112)/ 1.740823055783105d-18/,
*wt1(113)/ 2.126247365091890d-18/,wt1(114)/ 2.597002781312310d-18/
 data
*wt1(115)/ 3.171988392298073d-18/,wt1(116)/ 3.874272813855638d-18/,
*wt1(117)/ 4.732050718776532d-18/,wt1(118)/ 5.779735750225090d-18/,
*wt1(119)/ 7.059390282316958d-18/,wt1(120)/ 8.622352349869446d-18/,
*wt1(121)/ 1.053137301019041d-17/,wt1(122)/ 1.286303788941302d-17/,
*wt1(123)/ 1.571096273151045d-17/,wt1(124)/ 1.918939713846172d-17/,
*wt1(125)/ 2.343800282007509d-17/,wt1(126)/ 2.862721583723883d-17/,
*wt1(127)/ 3.496539240998871d-17/,wt1(128)/ 4.270678642844850d-17/,
*wt1(129)/ 5.216223744747131d-17/,wt1(130)/ 6.371103687921334d-17/,
*wt1(131)/ 7.781691646330707d-17/,wt1(132)/ 9.504569536565720d-17/,
*wt1(133)/ 1.160892016025935d-16/,wt1(134)/ 1.417915110596655d-16/,
*wt1(135)/ 1.731847439832610d-16/,wt1(136)/ 2.115280706868521d-16/,
*wt1(137)/ 2.583612876753947d-16/,wt1(138)/ 3.155627883343085d-16/,
*wt1(139)/ 3.854297646671823d-16/,wt1(140)/ 4.707643426724462d-16/,
*wt1(141)/ 5.749936655677889d-16/,wt1(142)/ 7.022978436839158d-16/,
*wt1(143)/ 8.577897883808816d-16/,wt1(144)/ 1.047705221625211d-15/,
*wt1(145)/ 1.279672050432180d-15/,wt1(146)/ 1.562992451553929d-15/,
*wt1(147)/ 1.909046462722709d-15/,wt1(148)/ 2.331710624424154d-15/,
*wt1(149)/ 2.847962809294981d-15/,wt1(150)/ 3.478503311969800d-15/,
*wt1(151)/ 4.248661490032314d-15/,wt1(152)/ 5.189316858234294d-15/
 data
*wt1(153)/ 6.338258511920505d-15/,wt1(154)/ 7.741550588468052d-15/,
*wt1(155)/ 9.455571172640662d-15/,wt1(156)/ 1.154903563040828d-14/,
*wt1(157)/ 1.410605553114248d-14/,wt1(158)/ 1.722913542283047d-14/,
*wt1(159)/ 2.104376349301754d-14/,wt1(160)/ 2.570284789891445d-14/,
*wt1(161)/ 3.139360843035402d-14/,wt1(162)/ 3.834414037586157d-14/,
*wt1(163)/ 4.683376407804295d-14/,wt1(164)/ 5.720273100006960d-14/,
*wt1(165)/ 6.986777175159004d-14/,wt1(166)/ 8.533643955954289d-14/,
*wt1(167)/ 1.042304766775628d-13/,wt1(168)/ 1.273069965554705d-13/,
*wt1(169)/ 1.554936139359936d-13/,wt1(170)/ 1.899197032959808d-13/,
*wt1(171)/ 2.319692366825060d-13/,wt1(172)/ 2.833268748946759d-13/,
*wt1(173)/ 3.460574729307441d-13/,wt1(174)/ 4.226739834760657d-13/,
*wt1(175)/ 5.162571427996572d-13/,wt1(176)/ 6.305554147709428d-13/,
*wt1(177)/ 7.701652476024983d-13/,wt1(178)/ 9.406780256740293d-13/,
*wt1(179)/ 1.148951682734905d-12/,wt1(180)/ 1.403326528650366d-12/,
*wt1(181)/ 1.714034726435434d-12/,wt1(182)/ 2.093516885212530d-12/,
*wt1(183)/ 2.557039701256063d-12/,wt1(184)/ 3.123159736572392d-12/,
*wt1(185)/ 3.814655555107956d-12/,wt1(186)/ 4.659206104975550d-12/,
*wt1(187)/ 5.690798281976283d-12/,wt1(188)/ 6.950717591317659d-12/,
*wt1(189)/ 8.489674870123959d-12/,wt1(190)/ 1.036925035218241d-11/
```

```
 data
*wt1(191)/ 1.266510893213373d-11/,wt1(192)/ 1.546910089464080d-11/,
*wt1(193)/ 1.889412592258518d-11/,wt1(194)/ 2.307718220992490d-11/,
*wt1(195)/ 2.818672942203327d-11/,wt1(196)/ 3.442710316037366d-11/,
*wt1(197)/ 4.204966817018048d-11/,wt1(198)/ 5.135919134271227d-11/,
*wt1(199)/ 6.273074786797763d-11/,wt1(200)/ 7.661889201263122d-11/,
*wt1(201)/ 9.358330171196327d-11/,wt1(202)/ 1.143019267753882d-10/,
*wt1(203)/ 1.396099167826153d-10/,wt1(204)/ 1.705183920059801d-10/,
*wt1(205)/ 2.082735788845278d-10/,wt1(206)/ 2.543834767842491d-10/,
*wt1(207)/ 3.107077590612055d-10/,wt1(208)/ 3.794954396016648d-10/,
*wt1(209)/ 4.635216515055463d-10/,wt1(210)/ 5.661404892806104d-10/,
*wt1(211)/ 6.914932735763221d-10/,wt1(212)/ 8.445820788220271d-10/,
*wt1(213)/ 1.031587101304291d-09/,wt1(214)/ 1.259967952077770d-09/,
*wt1(215)/ 1.538947681019300d-09/,wt1(216)/ 1.879650592174711d-09/,
*wt1(217)/ 2.295841053082368d-09/,wt1(218)/ 2.804108039269473d-09/,
*wt1(219)/ 3.424993797807946d-09/,wt1(220)/ 4.183235823312564d-09/,
*wt1(221)/ 5.109492567813164d-09/,wt1(222)/ 6.240651650854630d-09/,
*wt1(223)/ 7.622470723630660d-09/,wt1(224)/ 9.309953703014609d-09/,
*wt1(225)/ 1.137139562283767d-08/,wt1(226)/ 1.388881160360346d-08/,
*wt1(227)/ 1.696413754044688d-08/,wt1(228)/ 2.071966056701105d-08/
 data
*wt1(229)/ 2.530753298545001d-08/,wt1(230)/ 3.091008276019684d-08/,
*wt1(231)/ 3.775442398336976d-08/,wt1(232)/ 4.611239491146397d-08/,
*wt1(233)/ 5.632301501798812d-08/,wt1(234)/ 6.879156104001002d-08/,
*wt1(235)/ 8.402411522137500d-08/,wt1(236)/ 1.026248703632157d-07/,
*wt1(237)/ 1.253493263032310d-07/,wt1(238)/ 1.530981849317344d-07/,
*wt1(239)/ 1.869993328577540d-07/,wt1(240)/ 2.283954331971998d-07/,
*wt1(241)/ 2.789703820965002d-07/,wt1(242)/ 3.407255717347445d-07/,
*wt1(243)/ 4.161751163160901d-07/,wt1(244)/ 5.083021657140526d-07/,
*wt1(245)/ 6.208605585127383d-07/,wt1(246)/ 7.582965471628845d-07/,
*wt1(247)/ 9.262152932479593d-07/,wt1(248)/ 1.131243348429578d-06/,
*wt1(249)/ 1.381750677989317d-06/,wt1(250)/ 1.687612654971312d-06/,
*wt1(251)/ 2.061328504830195d-06/,wt1(252)/ 2.517614221705925d-06/,
*wt1(253)/ 3.075136493715169d-06/,wt1(254)/ 3.755823026355034d-06/,
*wt1(255)/ 4.587552788474290d-06/,wt1(256)/ 5.602996673366901d-06/,
*wt1(257)/ 6.843794758360242d-06/,wt1(258)/ 8.358620144706847d-06/,
*wt1(259)/ 1.020967008292656d-05/,wt1(260)/ 1.246945025422475d-05/,
*wt1(261)/ 1.523086887102219d-05/,wt1(262)/ 1.860192083272764d-05/,
*wt1(263)/ 2.272139580071132d-05/,wt1(264)/ 2.775012613215501d-05/,
*wt1(265)/ 3.389544779882663d-05/,wt1(266)/ 4.139684113647092d-05/
 data
*wt1(267)/ 5.056404631634212d-05/,wt1(268)/ 6.175357572143610d-05/,
*wt1(269)/ 7.542814171915086d-05/,wt1(270)/ 9.211833800758343d-05/,
*wt1(271)/ 1.125153468149895d-04/,wt1(272)/ 1.374085645976000d-04/,
*wt1(273)/ 1.678302935465055d-04/,wt1(274)/ 2.049544027080777d-04/,
*wt1(275)/ 2.503222397064943d-04/,wt1(276)/ 3.056782429102864d-04/,
*wt1(277)/ 3.733227575027256d-04/,wt1(278)/ 4.558456931739724d-04/,
*wt1(279)/ 5.566775872841220d-04/,wt1(280)/ 6.796584276296773d-04/,
*wt1(281)/ 8.298982533418059d-04/,wt1(282)/ 1.013079388381259d-03/,
*wt1(283)/ 1.236799879071809d-03/,wt1(284)/ 1.509444128879166d-03/,
*wt1(285)/ 1.842278291863989d-03/,wt1(286)/ 2.247620855799141d-03/,
*wt1(287)/ 2.742106207595588d-03/,wt1(288)/ 3.343716405309918d-03/,
*wt1(289)/ 4.076852784788600d-03/,wt1(290)/ 4.967507347193811d-03/,
*wt1(291)/ 6.051117627655160d-03/,wt1(292)/ 7.364681573940744d-03/,
*wt1(293)/ 8.958863753805689d-03/,wt1(294)/ 1.088505212285743d-02/,
*wt1(295)/ 1.321388763870185d-02/,wt1(296)/ 1.601387913655473d-02/,
*wt1(297)/ 1.937949548174205d-02/,wt1(298)/ 2.339556612555552d-02/,
*wt1(299)/ 2.817934173583488d-02/,wt1(300)/ 3.382079060619734d-02/,
*wt1(301)/ 4.044464120111045d-02/,wt1(302)/ 4.810974644754662d-02/,
*wt1(303)/ 5.689934182811928d-02/,wt1(304)/ 6.675168423575902d-02/
 data
*wt1(305)/ 7.759114407311523d-02/,wt1(306)/ 8.904761297446566d-02/,
*wt1(307)/ 1.006540591855666d-01/,wt1(308)/ 1.114010542073269d-01/,
*wt1(309)/ 1.200693293287907d-01/,wt1(310)/ 1.245859746888521d-01/,
*wt1(311)/ 1.226876554954843d-01/,wt1(312)/ 1.111594205067191d-01/,
*wt1(313)/ 8.731995478664586d-02/,wt1(314)/ 4.846500030224664d-02/,
*wt1(315)/-4.956156637375708d-03/,wt1(316)/-6.947108182097863d-02/,
*wt1(317)/-1.335126953345830d-01/,wt1(318)/-1.795750728565332d-01/,
*wt1(319)/-1.819110629504884d-01/,wt1(320)/-1.209847646201776d-01/,
*wt1(321)/ 5.791427226916448d-03/,wt1(322)/ 1.534812881363312d-01/,
*wt1(323)/ 2.352209731069856d-01/,wt1(324)/ 1.515040026640386d-01/,
*wt1(325)/-8.279085782145131d-02/,wt1(326)/-2.711543104015952d-01/,
*wt1(327)/-1.331279806376289d-01/,wt1(328)/ 2.242058279665979d-01/,
*wt1(329)/ 2.304292248782090d-01/,wt1(330)/-2.572813803130152d-01/,
*wt1(331)/-1.475549438192476d-01/,wt1(332)/ 3.611935526731244d-01/,
*wt1(333)/-2.435371552288723d-01/,wt1(334)/ 3.811454725614711d-02/,
*wt1(335)/ 8.273650321459338d-02/,wt1(336)/-1.091467781119748d-01/,
*wt1(337)/ 8.964328522861212d-02/,wt1(338)/-6.097788815932206d-02/,
*wt1(339)/ 3.780123181837066d-02/,wt1(340)/-2.243862121105374d-02/,
*wt1(341)/ 1.315627818991557d-02/,wt1(342)/-7.789693311677149d-03/
 data
*wt1(343)/ 4.737032461705975d-03/,wt1(344)/-2.996529369596568d-03/,
*wt1(345)/ 1.987854486046977d-03/,wt1(346)/-1.386982713556120d-03/,
*wt1(347)/ 1.015740041573320d-03/,wt1(348)/-7.763183089745113d-04/,
*wt1(349)/ 6.146040263274792d-04/,wt1(350)/-5.002069052247894d-04/,
*wt1(351)/ 4.157061680153099d-04/,wt1(352)/-3.508644775971589d-04/,
*wt1(353)/ 2.994936403367197d-04/,wt1(354)/-2.577357341490762d-04/,
*wt1(355)/ 2.231048844827660d-04/,wt1(356)/-1.939428674265101d-04/,
*wt1(357)/ 1.691035669098717d-04/,wt1(358)/-1.477663053511028d-04/,
*wt1(359)/ 1.293230643516167d-04/,wt1(360)/-1.133088634015300d-04/,
*wt1(361)/ 9.935787087369453d-05/,wt1(362)/-8.717517169548774d-05/,
*wt1(363)/ 7.651820346514748d-05/,wt1(364)/-6.718422692060818d-05/,
*wt1(365)/ 5.900162548547053d-05/,wt1(366)/-5.182370043717230d-05/,
*wt1(367)/ 4.552413144789780d-05/,wt1(368)/-3.999355821930881d-05/,
*wt1(369)/ 3.513692405823457d-05/,wt1(370)/-3.087135626461035d-05/,
*wt1(371)/ 2.712444449271282d-05/,wt1(372)/-2.383282162264958d-05/,
*wt1(373)/ 2.094097402272335d-05/,wt1(374)/-1.840022834334601d-05/,
*wt1(375)/ 1.616788131114735d-05/,wt1(376)/-1.420645034584182d-05/,
*wt1(377)/ 1.248302592309649d-05/,wt1(378)/-1.096870811434504d-05/,
*wt1(379)/ 9.638113747637144d-06/,wt1(380)/-8.468945120923146d-06/
 data
*wt1(381)/ 7.441613142631071d-06/,wt1(382)/-6.538907689594796d-06/,
*wt1(383)/ 5.745708304710312d-06/,wt1(384)/-5.048729864716120d-06/,
*wt1(385)/ 4.436299286415519d-06/,wt1(386)/-3.898159774165185d-06/,
*wt1(387)/ 3.425299055300640d-06/,wt1(388)/-3.009798361246221d-06/,
*wt1(389)/ 2.644699605727544d-06/,wt1(390)/-2.323888735130234d-06/,
*wt1(391)/ 2.041993364758909d-06/,wt1(392)/-1.794292864274530d-06/,
*wt1(393)/ 1.576639283761010d-06/,wt1(394)/-1.385387831873578d-06/,
*wt1(395)/ 1.217335820899756d-06/,wt1(396)/-1.069669061011715d-06/,
*wt1(397)/ 9.399147533431300d-07/,wt1(398)/-8.259000682217990d-07/,
*wt1(399)/ 7.257157398665060d-07/,wt1(400)/-6.376840939530841d-07/,
*wt1(401)/ 5.603309658957991d-07/,wt1(402)/-4.923610193285461d-07/,
*wt1(403)/ 4.326360479415582d-07/,wt1(404)/-3.801559094539231d-07/,
*wt1(405)/ 3.340417795380239d-07/,wt1(406)/-2.935214398955969d-07/,
*wt1(407)/ 2.579163472426386d-07/,wt1(408)/-2.266302675538730d-07/,
*wt1(409)/ 1.991392901933410d-07/,wt1(410)/-1.749830560542944d-07/,
*wt1(411)/ 1.537570500204901d-07/,wt1(412)/-1.351058266793548d-07/,
*wt1(413)/ 1.187170569339987d-07/,wt1(414)/-1.043162975874935d-07/,
*wt1(415)/ 9.166239627647569d-08/,wt1(416)/-8.054345360702865d-08/,
*wt1(417)/ 7.077327442848185d-08/,wt1(418)/-6.218824944698919d-08/
 data
*wt1(419)/ 5.464461550533503d-08/,wt1(420)/-4.801604844364623d-08/,
*wt1(421)/ 4.219154778426137d-08/,wt1(422)/-3.707357778095240d-08/,
*wt1(423)/ 3.257643398813585d-08/,wt1(424)/-2.862480813390940d-08/,
*wt1(425)/ 2.515252712714758d-08/,wt1(426)/-2.210144493659511d-08/,
*wt1(427)/ 1.942046882050334d-08/,wt1(428)/-1.706470371165871d-08/,
*wt1(429)/ 1.499470045020363d-08/,wt1(430)/-1.317579520863365d-08/,
*wt1(431)/ 1.157752900730261d-08/,wt1(432)/-1.017313762883149d-08/,
*wt1(433)/ 8.939103427741988d-09/,wt1(434)/-7.854761525494339d-09/,
*wt1(435)/ 6.901953769038680d-09/,wt1(436)/-6.064724650448546d-09/,
*wt1(437)/ 5.329054110861765d-09/,wt1(438)/-4.682622765482983d-09/,
*wt1(439)/ 4.114605612289784d-09/,wt1(440)/-3.615490760147216d-09/,
*wt1(441)/ 3.176920140880805d-09/,wt1(442)/-2.791549544501460d-09/,
*wt1(443)/ 2.452925636291220d-09/,wt1(444)/-2.155377892876758d-09/,
*wt1(445)/ 1.893923644634260d-09/,wt1(446)/-1.664184635389759d-09/,
*wt1(447)/ 1.462313704613378d-09/,wt1(448)/-1.284930364574083d-09/,
*wt1(449)/ 1.129064192077534d-09/,wt1(450)/-9.921050860675488d-10/,
*wt1(451)/ 8.717595588633507d-10/,wt1(452)/-7.660123299045589d-10/,
*wt1(453)/ 6.730925786777026d-10/,wt1(454)/-5.914442910883912d-10/,
*wt1(455)/ 5.197002025874266d-10/,wt1(456)/-4.566589020735369d-10/
 data
*wt1(457)/ 4.012647133825055d-10/,wt1(458)/-3.525900173049568d-10/,
*wt1(459)/ 3.098197178898629d-10/,wt1(460)/-2.722375929324441d-10/,
*wt1(461)/ 2.392143002251476d-10/,wt1(462)/-2.101968387708856d-10/,
*wt1(463)/ 1.846992883946521d-10/,wt1(464)/-1.622946726187645d-10/,
*wt1(465)/ 1.426078085707891d-10/,wt1(466)/-1.253090242443995d-10/,
*wt1(467)/ 1.101086379038488d-10/,wt1(468)/-9.675210714907983d-11/,
*wt1(469)/ 8.501576639446501d-11/,wt1(470)/-7.470308140061063d-11/,
*wt1(471)/ 6.564135815534322d-11/,wt1(472)/-5.767885098561786d-11/,
*wt1(473)/ 5.068222145975428d-11/,wt1(474)/-4.453430552590912d-11/,
*wt1(475)/ 3.913215150436401d-11/,wt1(476)/-3.438529608259476d-11/,
*wt1(477)/ 3.021424943923932d-11/,wt1(478)/-2.654916412468056d-11/,
*wt1(479)/ 2.332866540856460d-11/,wt1(480)/-2.049882351089608d-11/,
*wt1(481)/ 1.801225050660065d-11/,wt1(482)/-1.582730677850963d-11/,
*wt1(483)/ 1.390740372895185d-11/,wt1(484)/-1.222039107416370d-11/,
*wt1(485)/ 1.073801846245205d-11/,wt1(486)/-9.435462400371624d-12/,
*wt1(487)/ 8.290910564095319d-12/,wt1(488)/-7.285196534694753d-12/,
*wt1(489)/ 6.401478841143664d-12/,wt1(490)/-5.624958936714699d-12/,
*wt1(491)/ 4.942633385892820d-12/,wt1(492)/-4.343076111680976d-12/,
*wt1(493)/ 3.816247056854583d-12/,wt1(494)/-3.353324055246731d-12/
 data
*wt1(495)/ 2.946555097697522d-12/,wt1(496)/-2.589128518642891d-12/,
*wt1(497)/ 2.275058929421262d-12/,wt1(498)/-1.999086988193079d-12/,
*wt1(499)/ 1.756591328117270d-12/,wt1(500)/-1.543511168961402d-12/,
*wt1(501)/ 1.356278316168970d-12/,wt1(502)/-1.191757408630524d-12/,
*wt1(503)/ 1.047193414582761d-12/,wt1(504)/-9.201654964381461d-13/,
*wt1(505)/ 8.085464719703853d-13/,wt1(506)/-7.104671929827940d-13/,
*wt1(507)/ 6.242852449483966d-13/,wt1(508)/-5.485574434836370d-13/,
*wt1(509)/ 4.820156670945356d-13/,wt1(510)/-4.235456214910329d-13/,
*wt1(511)/ 3.721681798548054d-13/,wt1(512)/-3.270229865892332d-13/,
*wt1(513)/ 2.873540499880031d-13/,wt1(514)/-2.524970825616983d-13/,
*wt1(515)/ 2.218683770237039d-13/,wt1(516)/-1.949550316531413d-13/,
*wt1(517)/ 1.713063613517743d-13/,wt1(518)/-1.505263505679493d-13/,
*wt1(519)/ 1.322670217058939d-13/,wt1(520)/-1.162226079680830d-13/,
*wt1(521)/ 1.021244330498886d-13/,wt1(522)/-8.973641194353509d-14/,
*wt1(523)/ 7.885109750931469d-14/,wt1(524)/-6.928620661062564d-14/,
*wt1(525)/ 6.088156763982168d-14/,wt1(526)/-5.349643831874401d-14/,
*wt1(527)/ 4.700714885865928d-14/,wt1(528)/-4.130503101260246d-14/,
*wt1(529)/ 3.629459834036019d-14/,wt1(530)/-3.189194721306437d-14/,
*wt1(531)/ 2.802335178098858d-14/,wt1(532)/-2.462402937628328d-14/
 data
*wt1(533)/ 2.163705567637205d-14/,wt1(534)/-1.901241146150821d-14/,
*wt1(535)/ 1.670614500365970d-14/,wt1(536)/-1.467963606029139d-14/,
*wt1(537)/ 1.289894914807883d-14/,wt1(538)/-1.133426526661138d-14/,
*wt1(539)/ 9.959382555826870d-15/,wt1(540)/-8.751277525286314d-15/,
*wt1(541)/ 7.689719507742115d-15/,wt1(542)/-6.756931880732829d-15/,
*wt1(543)/ 5.937294383089198d-15/,wt1(544)/-5.217081541398457d-15/,
*wt1(545)/ 4.584232826172386d-15/,wt1(546)/-4.028150688809716d-15/,
*wt1(547)/ 3.539523097325829d-15/,wt1(548)/-3.110167599068186d-15/,
*wt1(549)/ 2.732894299122300d-15/,wt1(550)/-2.401385459874979d-15/,
*wt1(551)/ 2.110089705537303d-15/,wt1(552)/-1.854129059999304d-15/,
*wt1(553)/ 1.629217261290832d-15/,wt1(554)/-1.431587984759346d-15/,
*wt1(555)/ 1.257931773006046d-15/,wt1(556)/-1.105340616423372d-15/,
*wt1(557)/ 9.712592562913251d-16/,wt1(558)/-8.534423949641643d-16/,
*wt1(559)/ 7.499170965983152d-16/,wt1(560)/-6.589497487925016d-16/,
*wt1(561)/ 5.790170318870357d-16/,wt1(562)/-5.087804097802586d-16/,
*wt1(563)/ 4.470637150906226d-16/,wt1(564)/-3.928334533103688d-16/,
*wt1(565)/ 3.451814961285280d-16/,wt1(566)/-3.033098741093958d-16/,
*wt1(567)/ 2.665174140678718d-16/,wt1(568)/-2.341879973739602d-16/,
*wt1(569)/ 2.057802425625493d-16/,wt1(570)/-1.808184394757190d-16/
 data
*wt1(571)/ 1.588845831226758d-16/,wt1(572)/-1.396113738580090d-16/,
*wt1(573)/ 1.226760666607451d-16/,wt1(574)/-1.077950665155628d-16/,
*wt1(575)/ 9.471917939160380d-17/,wt1(576)/-8.322943929278698d-17/,
*wt1(577)/ 7.313344150029316d-17/,wt1(578)/-6.426212060448496d-17/,
*wt1(579)/ 5.646691937188211d-17/,wt1(580)/-4.961730103765172d-17/,
*wt1(581)/ 4.359856336499476d-17/,wt1(582)/-3.830991786613391d-17/,
*wt1(583)/ 3.366280201995165d-17/,wt1(584)/-2.957939622304163d-17/,
*wt1(585)/ 2.599132063935437d-17/,wt1(586)/-2.283849012615860d-17/,
*wt1(587)/ 2.006810806115271d-17/,wt1(588)/-1.763378222157133d-17/,
*wt1(589)/ 1.549474790998944d-17/,wt1(590)/-1.361518531745643d-17/,
*wt1(591)/ 1.196361969267083d-17/,wt1(592)/-1.051239427255871d-17/,
*wt1(593)/ 9.237207148064648d-18/,wt1(594)/-8.116704309607398d-18/,
*wt1(595)/ 7.132122057424711d-18/,wt1(596)/-6.266972788687375d-18/,
*wt1(597)/ 5.506768899643017d-18/,wt1(598)/-4.838780179316216d-18/,
*wt1(599)/ 4.251820632131413d-18/,wt1(600)/-3.736061159634172d-18/,
*wt1(601)/ 3.282864964518770d-18/,wt1(602)/-2.884642920675208d-18/,
*wt1(603)/ 2.534726487315339d-18/,wt1(604)/-2.227256039024718d-18/,
*wt1(605)/ 1.957082741736910d-18/,wt1(606)/-1.719682331465223d-18/,
*wt1(607)/ 1.511079351949921d-18/,wt1(608)/-1.327780582522198d-18/
 data
*wt1(609)/ 1.166716541382233d-18/,wt1(610)/-1.025190084716866d-18/,
*wt1(611)/ 9.008312409083032d-19/,wt1(612)/-7.915575234972099d-19/,
*wt1(613)/ 6.955390583093758d-19/,wt1(614)/-6.111679407722684d-19/,
*wt1(615)/ 5.370313102870695d-19/,wt1(616)/-4.718876907676534d-19/,
*wt1(617)/ 4.146462011511975d-19/,wt1(618)/-3.643482877427443d-19/,
*wt1(619)/ 3.201516724850009d-19/,wt1(620)/-2.813162483527929d-19/,
*wt1(621)/ 2.471916856791182d-19/,wt1(622)/-2.172065418717432d-19/,
*wt1(623)/ 1.908586921555019d-19/,wt1(624)/-1.677069210972190d-19/,
*wt1(625)/ 1.473635341074691d-19/,wt1(626)/-1.294878651921690d-19/,
*wt1(627)/ 1.137805722345515d-19/,wt1(628)/-9.997862427642855d-20/,
*wt1(629)/ 8.785089685693581d-20/,wt1(630)/-7.719430165048124d-20/,
*wt1(631)/ 6.783038559324186d-20/,wt1(632)/-5.960234254861157d-20/,
*wt1(633)/ 5.237238746867620d-20/,wt1(634)/-4.601944907799879d-20/,
*wt1(635)/ 4.043714244041481d-20/,wt1(636)/-3.553198745789200d-20/,
*wt1(637)/ 3.122184346997662d-20/,wt1(638)/-2.743453374142427d-20/,
*wt1(639)/ 2.410663680492846d-20/,wt1(640)/-2.118242441910240d-20/,
*wt1(641)/ 1.861292835641397d-20/,wt1(642)/-1.635512039302220d-20/,
*wt1(643)/ 1.437119176831859d-20/,wt1(644)/-1.262792004764438d-20/,
*wt1(645)/ 1.109611278531706d-20/,wt1(646)/-9.750118671903616d-21/
 data
*wt1(647)/ 8.567397980805881d-21/,wt1(648)/-7.528145122423124d-21/,
*wt1(649)/ 6.614956985587211d-21/,wt1(650)/-5.812541511365789d-21/,
*wt1(651)/ 5.107461617652140d-21/,wt1(652)/-4.487910185482492d-21/,
*wt1(653)/ 3.943512339081440d-21/,wt1(654)/-3.465151709308229d-21/,
*wt1(655)/ 3.044817771736880d-21/,wt1(656)/-2.675471703023044d-21/,
*wt1(657)/ 2.350928509369901d-21/,wt1(658)/-2.065753454081455d-21/,
*wt1(659)/ 1.815171050414263d-21/,wt1(660)/-1.594985094847466d-21/,
*wt1(661)/ 1.401508399670877d-21/,wt1(662)/-1.231501046949432d-21/,
*wt1(663)/ 1.082116130537259d-21/,wt1(664)/-9.508520796066248d-22/,
*wt1(665)/ 8.355107675059663d-22/,wt1(666)/-7.341607053512435d-22/,
*wt1(667)/ 6.451047024956878d-22/,wt1(668)/-5.668514489455267d-22/,
*wt1(669)/ 4.980905412312518d-22/,wt1(670)/-4.376705343564336d-22/,
*wt1(671)/ 3.845796559678223d-22/,wt1(672)/-3.379288625754182d-22/,
*wt1(673)/ 2.969369534739024d-22/,wt1(674)/-2.609174906141295d-22/,
*wt1(675)/ 2.292673041856001d-22/,wt1(676)/-2.014563919875952d-22/,
*wt1(677)/ 1.770190438740656d-22/,wt1(678)/-1.555460422525817d-22/,
*wt1(679)/ 1.366778079672955d-22/,wt1(680)/-1.200983777985825d-22/,
*wt1(681)/ 1.055301137906078d-22/,wt1(682)/-9.272905558580399d-23/,
*wt1(683)/ 8.148083664639517d-23/,wt1(684)/-7.159709489187570d-23/
 data
*wt1(685)/ 6.291231754727859d-23/,wt1(686)/-5.528106802589832d-23/,
*wt1(687)/ 4.857554932085610d-23/,wt1(688)/-4.268346388064600d-23/,
*wt1(689)/ 3.750613425528731d-23/,wt1(690)/-3.295685226039235d-23/,
*wt1(691)/ 2.895942787876105d-23/,wt1(692)/-2.544691315369192d-23/,
*wt1(693)/ 2.236048039475426d-23/,wt1(694)/-1.964843688844607d-23/,
*wt1(695)/ 1.726535979915737d-23/,wt1(696)/-1.517133602902566d-23/,
*wt1(697)/ 1.333129356639782d-23/,wt1(698)/-1.171441336528463d-23/,
*wt1(699)/ 1.029361299133160d-23/,wt1(700)/-9.045094266181886d-24/,
*wt1(701)/ 7.947947207266389d-24/,wt1(702)/-6.983802597398395d-24/,
*wt1(703)/ 6.136526075969871d-24/,wt1(704)/-5.391947793289710d-24/,
*wt1(705)/ 4.737623457434066d-24/,wt1(706)/-4.162624052945447d-24/,
*wt1(707)/ 3.657351341563861d-24/,wt1(708)/-3.213375054547230d-24/,
*wt1(709)/ 2.823287629843971d-24/,wt1(710)/-2.480574676640252d-24/,
*wt1(711)/ 2.179502108849238d-24/,wt1(712)/-1.915020796639139d-24/,
*wt1(713)/ 1.682686753114865d-24/,wt1(714)/-1.478592609598710d-24/,
*wt1(715)/ 1.299306570857684d-24/,wt1(716)/-1.141816901768704d-24/,
*wt1(717)/ 1.003481245022785d-24/,wt1(718)/-8.819804660957208d-25/,
*wt1(719)/ 7.752768477813106d-25/,wt1(720)/-6.815767858002522d-25/,
*wt1(721)/ 5.992985900665248d-25/,wt1(722)/-5.270459063734392d-25/
 data
*wt1(723)/ 4.635862411179793d-25/,wt1(724)/-4.078328923510551d-25/,
*wt1(725)/ 3.588287019442058d-25/,wt1(726)/-3.157312905756895d-25/,
*wt1(727)/ 2.778003181153750d-25/,wt1(728)/-2.443872105993099d-25/,
*wt1(729)/ 2.149270912330287d-25/,wt1(730)/-1.889318548953116d-25/,
*wt1(731)/ 1.659828344765530d-25/,wt1(732)/-1.457219798327836d-25/,
*wt1(733)/ 1.278418979568063d-25/,wt1(734)/-1.120761165538443d-25/,
*wt1(735)/ 9.819063583398694d-26/,wt1(736)/-8.597706957547351d-26/,
*wt1(737)/ 7.524761802539771d-26/,wt1(738)/-6.583229692872437d-26/,
*wt1(739)/ 5.757815308418353d-26/,wt1(740)/-5.034908817517119d-26/,
*wt1(741)/ 4.402475606598968d-26/,wt1(742)/-3.849819197101101d-26/,
*wt1(743)/ 3.367329111558304d-26/,wt1(744)/-2.946341179122434d-26/,
*wt1(745)/ 2.579121784723996d-26/,wt1(746)/-2.258873118181162d-26/,
*wt1(747)/ 1.979667076757990d-26/,wt1(748)/-1.736329567227616d-26/,
*wt1(749)/ 1.524356928842631d-26/,wt1(750)/-1.339878795333426d-26/,
*wt1(751)/ 1.179604582048012d-26/,wt1(752)/-1.040714749090827d-26/,
*wt1(753)/ 9.207194488904074d-27/,wt1(754)/-8.173124929559144d-27/,
*wt1(755)/ 7.282515970348745d-27/,wt1(756)/-6.513100644712918d-27/,
*wt1(757)/ 5.843030126248539d-27/,wt1(758)/-5.251393008101120d-27/,
*wt1(759)/ 4.718791967930222d-27/,wt1(760)/-4.228406914229241d-27/
 data
```

```
     *wt1(761)/ 3.767695474750273d-27/,wt1(762)/-3.329913400113583d-27/,
     *wt1(763)/ 2.914300433863011d-27/,wt1(764)/-2.524366823406565d-27/,
     *wt1(765)/ 2.164690315762398d-27/,wt1(766)/-1.837673402805216d-27/,
     *wt1(767)/ 1.542301736687901d-27/,wt1(768)/-1.275894419058812d-27/,
     *wt1(769)/ 1.037166283039543d-27/,wt1(770)/-8.274817350070647d-28/,
     *wt1(771)/ 6.490942784063843d-28/,wt1(772)/-5.023466637656253d-28/,
     *wt1(773)/ 3.845211097313078d-28/,wt1(774)/-2.909163620505845d-28/,
     *wt1(775)/ 2.167800946716682d-28/,wt1(776)/-1.587457861911164d-28/,
     *wt1(777)/ 1.153739423502031d-28/,wt1(778)/-8.690036418978894d-29/,
     *wt1(779)/ 7.435572366263697d-29/,wt1(780)/-7.847142831542739d-29/,
     *wt1(781)/ 9.894985212873060d-29/,wt1(782)/-1.342720280592225d-28/,
     *wt1(783)/ 1.817424273402145d-28/,wt1(784)/-2.376361960950870d-28/,
     *wt1(785)/ 2.977819810711053d-28/,wt1(786)/-3.586836968065507d-28/,
     *wt1(787)/ 4.184476666502118d-28/,wt1(788)/-4.766085106688037d-28/,
     *wt1(789)/ 5.328777445258146d-28/,wt1(790)/-5.858824985834212d-28/,
     *wt1(791)/ 6.329409859706737d-28/,wt1(792)/-6.709268971860476d-28/,
     *wt1(793)/ 6.975416172179417d-28/,wt1(794)/-7.122929261229363d-28/,
     *wt1(795)/ 7.167821355397955d-28/,wt1(796)/-7.142209321177599d-28/,
     *wt1(797)/ 7.084785743816030d-28/,wt1(798)/-7.032486471839363d-28/
      data
     *wt1(799)/ 7.017650479151096d-28/,wt1(800)/-7.069608472164614d-28/,
     *wt1(801)/ 7.214920505613761d-28/
c
      nofun1=0
c-----error checks
      if(nb.lt.1.or.nrel.lt.1.or.bmax.le.0.0d0) go to 9999
      y=bmax*er**(nb-1)
      if(y.le.0.0d0) go to 9999
      ierr=0
c-----initialize related convolution within lagged convolution loops
      do 10 i=1,801
   10 key(i)=0
      nb1=nb+1
      lag=-1
c-----preset initial filter abscissa for starting bmax, the argument
c     used in the external function fun1(g).  note the abscissas
c     are equally spaced (e=dexp(.1d0), er=1.0d0/e) in log-space.
      y1=abscis/bmax
c-----lagged convolution, outermost loop 1010
      do 1010 ilag=1,nb
        lag=lag+1
        istore=nb1-ilag
        if(lag.gt.0) y1=y1*e
      arg(istore)=abscis/y1
c-------related convolution, innermost loop 1000
      do 1000 jrel=1,nrel
c---------special case flag none=1 is set if fun1(g)=0 for all g in
c         filter fixed range (using weights 229-339).
        none=0
        itol=ntol
        dsum=0.0d0
        cmax=0.0d0
        y=y1
c---------begin right side convolution at weight 299 (m=return label)
        assign 20 to m
        i=299
        y=y*e
c---------call pseudo subroutine at 100 (return to 20 via m assigned)
        go to 100
  20     cmax=dmax1(dabs(c),cmax)
        i=i+1
        y=y*e
c---------call pseudo subroutine at 100 (return to 20 via m assigned)
        if(i.le.339) go to 100
        if(cmax.eq.0.0d0) none=1
c---------establish truncation criterion
        cmax=tol*cmax
        assign 30 to m
c---------call pseudo subroutine at 100 (return to 30 via m assigned)
        go to 100
c---------check for filter truncation at right end
  30     if(dabs(c).le.cmax) go to 50
        itol=ntol
  40     i=i+1
        y=y*e
c---------call pseudo subroutine at 100 (return to 30 via m assigned)
        if(i.le.801) go to 100
  50     itol=itol-1
        if(itol.gt.0.and.i.lt.801) go to 40
        itol=ntol
        y=y1
c---------continue with left side convolution at weight 298
        assign 60 to m
        i=298
c---------call pseudo subroutine at 100 (return to 60 via m assigned)
        go to 100
```

```
c---------check for filter truncation at left end
   60   if(dabs(c).le.cmax.and.none.eq.0) go to 80
        itol=ntol
   70   i=i-1
        y=y*er
c---------call pseudo subroutine at 100 (return to 60 via m assigned)
        if(i.gt.0) go to 100
   80   itol=itol-1
        if(itol.gt.0.and.i.gt.1) go to 70
c---------normalize dsum by arg(istore) to account for integration
c         range change, and store in dans(istore,jrel)
        dans(istore,jrel)=dsum/arg(istore)
c---------skip over pseudo subroutine to end of do 1000 innermost loop
        go to 1000
c
c=============================================
=============================
c=====store/retrieve pseudo subroutine for related/lagged convolution.
c     the internal (pseudo) subroutine entry is label 100, and returns
c     to the label assigned to m.  this calling mechanism could occur
c     a maximum of 801*nb*nrel times, where parameters nb>0 and nrel>0
c     can be arbitrarily large.  if a more-structured standard fortran
c     subroutine call was used, then the usual compiler linkage
c     convention could generate a maximum of 801*nb*nrel machine-
c     language instructions for register saves/restores and other
c     memory references.  for most compilers, timing tests reveal that
c     the pseudo-call method used here generated faster machine code
c     than with using external subroutine calls (e.g., call linkage
c     versus pseudo-call ratio was 2.6:1 on a vax-11/780 using
c     nb=50,nrel=61, and nofun1=199).
c
c=========pseudo-call entry point at 100 (returns via go to m below)
  100   look=i+lag
        iq=look/802
        ir=mod(look,802)
        if(ir.eq.0) ir=1
        iroll=iq*801
        if(key(ir).le.iroll) go to 150
c==========use existing saved functional values in dwork(ir,jrel)
  110   if(nord(jrel)) 130,120,130
  120   c=dwork(ir,jrel)*wt0(i)
        go to 140
  130   c=dwork(ir,jrel)*wt1(i)
```

```
 140    dsum=dsum+c
c=========return convolution control via assigned m value, and with
c        the last convolution product (c)
        go to m,(20,30,60)
c=========compute external fun1 directly only when necessary
 150    key(ir)=iroll+ir
        g=y
        call fun1(g, dwork, nrel, ir)
*         dwork(ir,1)=fun1(g)
        nofun1=nofun1+1
        go to 110
c=========end of pseudo subroutine (entry 100, return go to m above)
c===========================================
============================
c
c-------end loop 1000 (get remaining related convolutions for this arg)
 1000   continue
c-----end loop 1010 (get remaining lagged convolutions for next arg)
 1010 continue
c-----exit with dans(nb,nrel),arg(nb) completed with minimal fun1 calls
      return
 9999 ierr=1
      return
      end
************************************************************************
************************************************************************
************************************************************************
*** from netlib, Sat Feb  1 06:58:51 EST 1992 ***
*
      SUBROUTINE CALCK0(ARG,RESULT,JINT)
*
C----------------------------------------------------------------------
C
C This packet computes modified Bessel functions of the second kind
C   and order zero, K0(X) and EXP(X)*K0(X), for real
C   arguments X.  It contains two function type subprograms, BESK0
C   and BESEK0, and one subroutine type subprogram, CALCK0.
C   the calling statements for the primary entries are
C
C                   Y=BESK0(X)
C   and
C                   Y=BESEK0(X)
C
```

```
C   where the entry points correspond to the functions K0(X) and
C   EXP(X)*K0(X), respectively.  The routine CALCK0 is
C   intended for internal packet use only, all computations within
C   the packet being concentrated in this routine.  The function
C   subprograms invoke CALCK0 with the statement
C          CALL CALCK0(ARG,RESULT,JINT)
C   where the parameter usage is as follows
C
C      Function                  Parameters for CALCK0
C       Call              ARG                  RESULT          JINT
C
C     BESK0(ARG)   0 .LT. ARG .LE. XMAX       K0(ARG)           1
C     BESEK0(ARG)     0 .LT. ARG           EXP(ARG)*K0(ARG)     2
C
C   The main computation evaluates slightly modified forms of near
C   minimax rational approximations generated by Russon and Blair,
C   Chalk River (Atomic Energy of Canada Limited) Report AECL-3461,
C   1969.  This transportable program is patterned after the
C   machine-dependent FUNPACK packet NATSK0, but cannot match that
C   version for efficiency or accuracy.  This version uses rational
C   functions that theoretically approximate K-SUB-0(X) to at
C   least 18 significant decimal digits.  The accuracy achieved
C   depends on the arithmetic system, the compiler, the intrinsic
C   functions, and proper selection of the machine-dependent
C   constants.
C
C*******************************************************************
C*******************************************************************
C
C Explanation of machine-dependent constants
C
C   beta   = Radix for the floating-point system
C   minexp = Smallest representable power of beta
C   maxexp = Smallest power of beta that overflows
C   XSMALL = Argument below which BESK0 and BESEK0 may
C            each be represented by a constant and a log.
C            largest X such that  1.0 + X = 1.0  to machine
C            precision.
C   XINF   = Largest positive machine number; approximately
C            beta**maxexp
C   XMAX   = Largest argument acceptable to BESK0;  Solution to
C            equation:
C               W(X) * (1-1/8X+9/128X**2) = beta**minexp
```

```
C          where  W(X) = EXP(-X)*SQRT(PI/2X)
C
C
C     Approximate values for some important machines are:
C
C
C                        beta       minexp      maxexp
C
C  CRAY-1        (S.P.)       2        -8193        8191
C  Cyber 180/185
C    under NOS   (S.P.)       2         -975        1070
C  IEEE (IBM/XT,
C    SUN, etc.)  (S.P.)       2         -126         128
C  IEEE (IBM/XT,
C    SUN, etc.)  (D.P.)       2        -1022        1024
C  IBM 3033      (D.P.)      16          -65          63
C  VAX D-Format  (D.P.)       2         -128         127
C  VAX G-Format  (D.P.)       2        -1024        1023
C
C
C                      XSMALL       XINF         XMAX
C
C CRAY-1        (S.P.)    3.55E-15   5.45E+2465    5674.858
C Cyber 180/855
C   under NOS   (S.P.)    1.77E-15   1.26E+322      672.788
C IEEE (IBM/XT,
C   SUN, etc.)  (S.P.)    5.95E-8    3.40E+38        85.337
C IEEE (IBM/XT,
C   SUN, etc.)  (D.P.)    1.11D-16   1.79D+308      705.342
C IBM 3033      (D.P.)    1.11D-16   7.23D+75       177.852
C VAX D-Format  (D.P.)    6.95D-18   1.70D+38        86.715
C VAX G-Format  (D.P.)    5.55D-17   8.98D+307      706.728
*
* sun 3/50
C
C*******************************************************************
C*******************************************************************
C
C Error returns
C
C  The program returns the value XINF for ARG .LE. 0.0, and the
C  BESK0 entry returns the value 0.0 for ARG .GT. XMAX.
C
```

```
C
C  Intrinsic functions required are:
C
C     EXP, LOG, SQRT
C
C  Latest modification: March 19, 1990
C
C  Authors: W. J. Cody and Laura Stoltz
C           Mathematics and Computer Science Division
C           Argonne National Laboratory
C           Argonne, IL 60439
C
C----------------------------------------------------------------------
      INTEGER I, JINT
      DOUBLE PRECISION
     1    ARG,F,G,ONE,P,PP,Q,QQ,RESULT,SUMF,SUMG,SUMP,SUMQ,TEMP,
     2    X,XINF,XMAX,XSMALL,XX,ZERO
      DIMENSION P(6),Q(2),PP(10),QQ(10),F(4),G(3)
C----------------------------------------------------------------------
C  Mathematical constants
C----------------------------------------------------------------------
      DATA ONE/1.0D0/,ZERO/0.0D0/
C----------------------------------------------------------------------
C  Machine-dependent constants
C----------------------------------------------------------------------
CD    DATA XSMALL/1.11D-16/,XINF/1.79D+308/,XMAX/705.342D0/
      data xsmall/1.11D-16/,xinf/1.79D+308/,xmax/705.342d0/

C----------------------------------------------------------------------
C
C     Coefficients for XSMALL .LE.  ARG  .LE. 1.0
C
C----------------------------------------------------------------------
      DATA   P/ 5.85992214128261000000D-04, 1.31660525649895718850D-01,
     1          1.19994637249107141090D+01, 4.68509012019348321880D+02,
     2          5.91690598522705123120D+03, 2.47081527203995526790D+03/
      DATA   Q/-2.49944189728323036460D+02, 2.13127143038491203800D+04/
      DATA   F/-1.64144528372990641000D+00,-2.96016578929588438660D+02,
     1        -1.77337846849529858860D+04,-4.03203407611454822980D+05/
      DATA   G/-2.50649724458779927300D+02, 2.98657131630540254890D+04,
     1        -1.61281363044581939980D+06/
C----------------------------------------------------------------------
C
```

```
C     Coefficients for  1.0 .LT. ARG
C
C----------------------------------------------------------------------
      DATA  PP/ 1.13949805573847781 74D+02, 3.68325899573402679 40D+03,
     1          3.10754089806843923 99D+04, 1.05770689480340219 57D+05,
     2          1.73988679025656862 51D+05, 1.50976463532899145 39D+05,
     3          7.15570627837640375 41D+04, 1.83215258701835377 25D+04,
     4          2.34447387641993150 21D+03, 1.16002494250760355 58D+02/
      DATA  QQ/ 2.00134430649492424 91D+02, 4.43296288897464088 58D+03,
     1          3.14746557502952788 25D+04, 9.74188297622680757 84D+04,
     2          1.51446446735201578 01D+05, 1.26898395879775987 27D+05,
     3          5.88246167858570277 52D+04, 1.48472283718023609 57D+04,
     4          1.88218908409827136 96D+03, 9.25565991773048398 11D+01/
C----------------------------------------------------------------------
      X = ARG
      IF (X .GT. ZERO) THEN
          IF (X .LE. ONE) THEN
C----------------------------------------------------------------------
C     0.0 .LT.  ARG  .LE. 1.0
C----------------------------------------------------------------------
              TEMP = LOG(X)
              IF (X .LT. XSMALL) THEN
C----------------------------------------------------------------------
C     Return for small ARG
C----------------------------------------------------------------------
                  RESULT = P(6)/Q(2) - TEMP
                ELSE
                  XX = X * X
                  SUMP = ((((P(1)*XX + P(2))*XX + P(3))*XX +
     1                   P(4))*XX + P(5))*XX + P(6)
                  SUMQ = (XX + Q(1))*XX + Q(2)
                  SUMF = ((F(1)*XX + F(2))*XX + F(3))*XX + F(4)
                  SUMG = ((XX + G(1))*XX + G(2))*XX + G(3)
                  RESULT = SUMP/SUMQ - XX*SUMF*TEMP/SUMG - TEMP
                  IF (JINT .EQ. 2) RESULT = RESULT * EXP(X)
              END IF
            ELSE IF ((JINT .EQ. 1) .AND. (X .GT. XMAX)) THEN
C----------------------------------------------------------------------
C     Error return for ARG .GT. XMAX
C----------------------------------------------------------------------
              RESULT = ZERO
            ELSE
C----------------------------------------------------------------------
```

```
C    1.0 .LT. ARG
C----------------------------------------------------------------------
              XX = ONE / X
              SUMP = PP(1)
              DO 120 I = 2, 10
                SUMP = SUMP*XX + PP(I)
 120           CONTINUE
              SUMQ = XX
              DO 140 I = 1, 9
                SUMQ = (SUMQ + QQ(I))*XX
 140           CONTINUE
              SUMQ = SUMQ + QQ(10)
              RESULT = SUMP / SUMQ / SQRT(X)
              IF (JINT .EQ. 1) RESULT = RESULT * EXP(-X)
          END IF
        ELSE
C----------------------------------------------------------------------
C    Error return for ARG .LE. 0.0
C----------------------------------------------------------------------
          RESULT = XINF
      END IF
C----------------------------------------------------------------------
C    Update error counts, etc.
C----------------------------------------------------------------------
      RETURN
C---------- Last line of CALCK0 ----------
      END
      DOUBLE PRECISION
     1    FUNCTION BESK0(X)
C----------------------------------------------------------------------
C
C This function program computes approximate values for the
C   modified Bessel function of the second kind of order zero
C   for arguments 0.0 .LT. ARG .LE. XMAX (see comments heading
C   CALCK0).
C
C  Authors: W. J. Cody and Laura Stoltz
C
C  Latest Modification: January 19, 1988
C
C----------------------------------------------------------------------
      INTEGER JINT
      DOUBLE PRECISION
     1    X, RESULT
C----------------------------------------------------------------------
     JINT = 1
     CALL CALCK0(X,RESULT,JINT)
     BESK0 = RESULT
     RETURN
C---------- Last line of BESK0 ----------
     END
     DOUBLE PRECISION
     1    FUNCTION BESEK0(X)
C----------------------------------------------------------------------
C
C This function program computes approximate values for the
C   modified Bessel function of the second kind of order zero
C   multiplied by the Exponential function, for arguments
C   0.0 .LT. ARG.
C
C  Authors: W. J. Cody and Laura Stoltz
C
C  Latest Modification: January 19, 1988
C
C----------------------------------------------------------------------
     INTEGER JINT
     DOUBLE PRECISION
     1    X, RESULT
C----------------------------------------------------------------------
     JINT = 2
     CALL CALCK0(X,RESULT,JINT)
     BESEK0 = RESULT
     RETURN
C---------- Last line of BESEK0 ----------
     END
************************************************************************
************************************************************************
************************************************************************
************************************************************************
     FUNCTION W(U)
C**********************************************************************
C
C THIS FUNCTION IS AN APPROXIMATION OF THE WELL FUNCTION
(EXPONENTIIAL C
C INTEGRAL) USE EQUATION 5.153 AND EQUATION 5.156 IN ABRAMOWITZ AND
C STEGUN, 1970.
```

```
C
C***********************************************************************

      IMPLICIT DOUBLE PRECISION(A-H,Q-Z)

      DATA A0,A1,A2,A3,A4,A5/ -0.57721566, 0.99999193, -0.24991055,
     +0.05519968, -0.00976004, 0.00107857/

      DATA B1,B2,B3,B4/ 9.5733223454, 25.6329561486, 21.0996530827,
     +3.9584969228/

      DATA C1,C2,C3,C4/ 8.5733287401, 18.0590169730, 8.6347608925,
     +0.2677737343/

      IF(U.LE.0.0) THEN
        WRITE(6,10)
 10     FORMAT(5X,'ERROR; THE ARGUMENT OF W(U) IS NEGATIVE')
        STOP
      ELSEIF(U.LE.1.0) THEN
        W=A0-LOG(U)+U*(A1+U*(A2+U*(A3+U*(A4+U*A5))))
      ELSE
        DUM1=C4+U*(C3+U*(C2+U*(C1+U)))
        DUM2=U*(B4+U*(B3+U*(B2+U*(B1+U))))
        W=(DUM1/DUM2)*DEXP(-U)
      ENDIF
      RETURN
      END
************************************************************************
************************************************************************
      subroutine pppt(h2a, nu, b, d, l, p, r, z, zu)
************************************************************************
**  This subroutine is used to compute the partial penetration effect  *
**  when the observation well is treated as a point.                   *
************************************************************************
*
      double precision b, besk0, bk, besarg, besarg1, d, error, h2a, ip,
     +     kkb2, kr, kz, linv, l, n, nu, old, p, pi, pilb, pidb,
     +     pizb, r, sinn, skr, ss, sy, tol, wt(18), z, zu
*
      logical done
*
      common /c1/ kr, ss, kz, sy
      common /cpi/ pi
      common /weight/ wt
*
      external function besk0
*
      done = .false.
      h2a  = 0.d0
      kkb2 = kz*pi*pi/kr/b/b
      n    = 0.d0
      pilb = pi*l/b
      pidb = pi*d /b
      pizb = pi*(b-z)/b
      skr  = ss/kr

 100  if(.not.done) then
        n    = n + 1.d0
        sinn = dcos(n*pizb) *(dsin(n*pilb) - dsin(n*pidb))
        besarg1 = n*n * kkb2 + nu
        ip   = 0.d0
        linv = 0.d0
        do 500 i = 1, 18
          ip = ip + p
          besarg = r * dsqrt(ip*skr + besarg1)
          bk = besk0(besarg)/i
          linv = linv + wt(i) * bk
 500     continue
        linv  = linv / n
        old   = h2a
        h2a   = h2a + sinn * linv
        error = dabs(old - h2a)
        tol   = dabs(h2a*10.d-10)
        if(n .ge. 500.d0) done = .true.
        if( error .lt. tol .and. n .gt. 15.d0) done = .true.
        goto 100
      endif
*
      return
      end
************************************************************************
      subroutine ppva(h2a, nu, b, d, l, p, r, zl, zu)
************************************************************************
** This subroutine is used to compute the partial penetration effect   *
** When drawdown is vertically averaged (zu > zl).                     *
************************************************************************
```

```
*
      double precision b, besk0, bk, besarg, besarg1, d, h2a, ip,
     +     kkb2, kr, kz, linv, l, n, nu, old, p, pi, pilb, pidb,
     +     pizlb, pizub, r, sinn, skr, ss, sy, wt(18), zl, zu
*
      logical done
*
      common /c1/ kr, ss, kz, sy
      common /cpi/ pi
      common /weight/ wt
*
      external function besk0
*
      done = .false.
      h2a  = 0.d0
      kkb2 = kz*pi*pi/kr/b/b
      n    = 0.d0
      pilb = pi*l/b
      pidb = pi*d /b
      pizlb = pi*(b-zl)/b
      pizub = pi*(b-zu)/b
      skr  = ss/kr

100   if(.not.done) then
         n    = n + 1.d0
         sinn = (dsin(n*pizlb)-dsin(n*pizub)) *
     +        (dsin(n*pilb) - dsin(n*pidb))
         besarg1 = n*n * kkb2 + nu
         ip   = 0.d0
         linv = 0.d0
         do 500 i = 1, 18
            ip = ip + p
            besarg = r * dsqrt(ip*skr + besarg1)
            bk = besk0(besarg)/i
            linv = linv + wt(i) * bk
500      continue
         linv  = linv / n / n
         old   = h2a
         h2a   = h2a + sinn * linv
         error = dabs(old - h2a)
         tol   = dabs(h2a*10.d-10)
         if(n .ge. 500.d0) done = .true.
         if(n .ge. 15 .and. error .lt. tol) done = .true.
```

```
        goto 100
      endif
*
      return
      end
************************************************************************
      subroutine f3pt(a, dwork, nrel, ir)
************************************************************************
**  This subroutine is called by dhankl.  It isused in computing       *
**  the unconfined (water table) effect, when the pumping well is      *
**  fully penetrating (d =0 and l = b) and the observation interval is *
**  treated as a point, (zl = zu).                                     *
************************************************************************
      double precision a, dwork(801, nrel),
     +     kr, ss, kz, sy,
     +     ader, ader1, ader2, aderb, aderbz, aderz, alphay,
     +     b, bz, f3, f3inv, f3a, ip, p, t, wt(18), z, zu
*
      common /c1/ kr, ss, kz, sy
      common /c2/ b, p, t, z, zu
      common /weight/ wt
*
      ader1  = a * a * kr
      ader2  = (ader1 + p*ss) / kz
      ader   = dsqrt(ader2)
      aderb  = ader * b
      aderz  = ader * z
      alphay = kz / sy
*
      f3inv = 0.0d0
      ip    = 0.0d0
*
      if(aderz.le.20.d0) then
        do 100 i = 1, 18
          ip    = ip + p
          ader2 = (ader1 + ip*ss) /kz
          ader  = dsqrt(ader2)
          aderb = ader * b
          aderz = ader * z
          f3a   = (alphay*ader*dsinh(aderb)+ ip*dcosh(aderb))*ader2
          f3    = cosh(aderz) / f3a
          f3inv  = f3inv + wt(i) * f3
 100     continue
```

```
      else
         bz = b - z
         do 200 i = 1, 18
            ip     = ip + p
            ader2  = (ader1 + ip*ss) /kz
            ader   = dsqrt(ader2)
            aderbz = ader * bz
            f3a    = (alphay*ader + ip) * ader2 * dexp(aderbz)
            f3inv  = f3inv + wt(i) / f3a
 200     continue
      endif
*
      dwork(ir,1) = p * f3inv * a
*
      return
      end
***********************************************************************
***********************************************************************
      subroutine f3fp(a, dwork, nrel, ir)
***********************************************************************
**  This routine is called by dhankl.  It is used in computing the      *
**  unconfined (water table) effect when the pumping and observation    *
**  wells are fully penetrating (d = 0. l = b, zl = 0, and zu = b).     *
***********************************************************************
      double precision kr, ss, kz, sy,
     +     a, dwork(801, nrel),
     +     ader, ader1, ader2, aderb, alphay,b, f3a,
     +     f3inv,ip, p, t, wt(18), zl, zu
*
      common /c1/ kr, ss, kz, sy
      common /c2/ b, p, t, zl, zu
      common /weight/ wt
*
      ader1  = a * a * kr
      ader2 = (ader1 + p*ss) / kz
      ader  = dsqrt(ader2)
      aderb = ader*b
      alphay = kz / sy
*
      f3inv  = 0.0d0
      ip     = 0.0d0
*
      if(aderb.lt.20.d0) then
```

```
          do 100 i = 1, 18
            ip    = ip + p
            ader2 = (ader1 + ip*ss) /kz
            ader  = dsqrt(ader2)
            aderb = ader * b
            f3a   = (alphay * ader + ip / dtanh(aderb)) * ader2*ader
            f3inv = f3inv + wt(i) / f3a
 100      continue
        else
          do 200 i = 1, 18
            ip    = ip + p
            ader2 = (ader1 + ip*ss) /kz
            ader  = dsqrt(ader2)
            f3a   = (alphay*ader + ip) * ader2*ader
            f3inv = f3inv + wt(i) / f3a
 200      continue
        endif
*
        dwork(ir,1) = p * f3inv * a
*
        return
        end
************************************************************************
************************************************************************
        subroutine f3va(a, dwork, nrel, ir)
************************************************************************
**  This subroutine is called by dhankl.  It is used in computing h2,      *
**  the unconfined (watertable) effect, when the pumping well is fully     *
**  penetrating, (d = 0, l = b), and the observation well is partially    *
**  penetrating (zl ne zu, and zl ne 0 or zu ne b).                        *
************************************************************************
*
        double precision a, dwork(801, nrel), p, ip, wt(18),
       +     kr, ss, kz, sy,
       +     ader, ader1, ader2, aderb, aderbz, aderzl, aderzu,
       +     alphay, azuml, b, bz, f3, f3a, f3inv,
       +     t, zl, zlu, zu
*
        common /c1/ kr, ss, kz, sy
        common /c2/ b, p, t, zl, zu
        common /weight/ wt
*
*
```

```
      ader1  = a * a * kr
      ader2  = (ader1 + p*ss) /kz
      ader   = dsqrt(ader2)
      aderb  = ader * b
      aderzu = ader * zu
      alphay = kz / sy
      zlu    = zl - zu
      azuml  = - ader * zlu / 2.d0
*
      f3inv = 0.0d0
      ip    = 0.0d0
*
      if(azuml .ge. 20.d0) then
        bz = b - zu
        do 100 i = 1, 18
          ip     = ip + p
          ader2  = (ader1 + ip*ss) /kz
          ader   = dsqrt(ader2)
          aderbz = ader*bz
          f3a    = (alphay*ader + ip) * dexp(aderbz) * ader2*ader
          f3inv  = f3inv + wt(i) / f3a
 100    continue
*
      elseif(aderzu .ge. 20.d0) then
        bz =   (b - zu)
        do 200 i = 1, 18
          ip     = ip + p
          ader2  = (ader1 + ip*ss) /kz
          ader   = dsqrt(ader2)
          aderbz = ader * bz
          aderzl = ader * zlu
          f3a    = (alphay*ader + ip) * ader2*ader * dexp(aderbz)
          f3     =  (1.d0 - dexp(aderzl)) / f3a
          f3inv  = f3inv + wt(i) * f3
 200    continue
*
      elseif(aderb .ge. 20.d0) then
         do 300 i = 1, 18
          ip     = ip + p
          ader2  = (ader1 + ip*ss) /kz
          ader   = dsqrt(ader2)
          aderb  = ader * b
          aderzl = ader * zl
          aderzu = ader * zu
          f3a    = (alphay*ader + ip) * ader2*ader * dexp(aderb)
          f3     = (dsinh(aderzu)-dsinh(aderzl)) / f3a
          f3inv  = f3inv + wt(i) * f3
 300     continue
        f3inv = 2.d0 * f3inv
      else
        do 400 i = 1, 18
          ip     = ip + p
          ader2  = (ader1 + ip*ss) /kz
          ader   = dsqrt(ader2)
          aderb  = ader * b
          aderzl = ader * zl
          aderzu = ader * zu
          f3a    = (alphay*ader*dsinh(aderb) + ip*dcosh(aderb))
          f3     = (dsinh(aderzu) - dsinh(aderzl)) / f3a / ader2/ader
          f3inv  = f3inv + wt(i) * f3
 400     continue
      endif
*
      dwork(ir,1) = p * f3inv * a
*
      return
      end
**********************************************************************
**********************************************************************
      subroutine ppf3pt(a, dwork, nrel, ir)
**********************************************************************
**  This subroutine is called by dhankl.  It is used in computing    *
**  the unconfined (water table) effect, when the pumping well is    *
**  partially penetrating and the observation well is treat as a     *
**  point (zl = zu).                                                  *
**********************************************************************
      double precision a, dwork(801, nrel), p, ip, wt(18)
*
      double precision kr, ss, kz, sy,
     +     ader, ader1, ader2, aderb, aderbd, aderbl, aderbz, aderd,
     +     aderl, aderz, alphay, b, bz, d, f3inv, f3a, fp, fpf3, l, t,
     +     z, zu

*
      common /c1/ kr, ss, kz, sy
      common /c2/ b, p, t, z, zu
      common /ld/ l, d
      common /weight/ wt
*
      alphay = kz / sy
      ader1  = a * a * kr
      ader2  = (ader1 + p*ss) / kz
      ader   = dsqrt(ader2)
      aderb  = ader * b
      aderz  = ader * z
*
      f3inv = 0.0d0
      ip    = 0.0d0
*
      if(aderz.ge.20.d0) then
        bz = b - z
        do 100 i = 1, 18
          ip     = ip + p
          ader2  = (ader1 + ip*ss) /kz
          ader   =  dsqrt(ader2)
          aderbz = ader * bz
          aderd  = ader * d
          aderl  = ader * l
          fp     =  dexp(-aderd) - dexp(-aderl)
          f3a    = (alphay*ader + ip) * ader2 * dexp(aderbz)
          fpf3   = fp / f3a
          f3inv  = f3inv + wt(i) * fpf3
 100    continue
      elseif(aderb .ge. 20.d0) then
        do 200 i = 1, 18
          ip  = ip + p
          ader2 = (ader1 + ip*ss) /kz
          ader  =  dsqrt(ader2)
          aderb = ader * b
          aderd = ader * d
          aderl = ader * l
          aderz = ader * z
          fp    = dexp(-aderd) - dexp(-aderl)
          f3a   = (alphay*ader + ip) * ader2 * dexp(aderb)
          fpf3  = fp * dcosh(aderz) / f3a
          f3inv = f3inv + wt(i) * fpf3
 200    continue
        f3inv = 2.d0 * f3inv
        else
```

```
            bd = b - d
            bl = b - l
         do 300 i = 1, 18
            ip     = ip + p
            ader2  = (ader1 + ip*ss) /kz
            ader   = dsqrt(ader2)
            aderb  = ader * b
            aderbd = ader * bd
            aderbl = ader * bl
            aderz  = ader * z
            fp     = (dsinh(aderbd) - dsinh(aderbl) ) / dsinh(aderb)
            f3a    = (alphay*ader*dsinh(aderb) + ip*dcosh(aderb))*ader2
            fpf3   = fp * cosh(aderz) / f3a
            f3inv  = f3inv + wt(i) * fpf3
 300       continue
      endif
*
      dwork(ir,1) = p * f3inv * a
*
      return
      end
************************************************************************
************************************************************************
      subroutine ppf3va(a, dwork, nrel, ir)
************************************************************************
** This subroutine is called by dhankl.  It is used in computing h2,   *
** the unconfined (watertable) effect when the pumping well and the    *
** observation well are partially penetrating (d ne 0 or l ne b and    *
** zl ne 0 or zu ne b).                                                *
************************************************************************
      double precision a, dwork(801, nrel), p, ip, wt(18),
     +     kr, ss, kz, sy,
     +     ader, ader1, ader2, aderb, aderbd, aderbl, aderbz, aderd,
     +     aderl, aderzl, aderzu, alphay, azuml, b, bd, bl, bz, d, f3a,
     +     f3inv, fp, fpf3, l, t, zl, zlu, zu
*
      common /c1/ kr, ss, kz, sy
      common /c2/ b, p, t, zl, zu
      common /ld/ l, d
      common /weight/ wt
*
      ader1  = a * a * kr
      ader2  = (ader1 + p*ss) /kz
      ader    = dsqrt(ader2)
      aderb   = ader * b
      aderzu  = ader * zu
      alphay  = kz / sy
      zlu     = zl - zu
      azuml   = - ader * zlu / 2.d0
*
      f3inv = 0.0d0
      ip    = 0.0d0
*
      if(azuml .ge. 20.d0) then
        bz = b - zu
        do 100 i = 1, 18
          ip     = ip + p
          ader2  = (ader1 + ip*ss) /kz
          ader   = dsqrt(ader2)
          aderbz = ader*bz
          aderd  = ader * d
          aderl  = ader * l
          fp     =  dexp(-aderd) - dexp(-aderl)
          f3a    = (alphay*ader + ip)*dexp(aderbz)*ader2*ader
          fpf3   = fp / f3a
          f3inv  = f3inv + wt(i) * fpf3
 100    continue
*
      elseif(aderzu .ge. 20.d0) then
        bz =   (b - zu)
        do 200 i = 1, 18
          ip     = ip + p
          ader2  = (ader1 + ip*ss) /kz
          ader   = dsqrt(ader2)
          aderbz = ader * bz
          aderd  = ader * d
          aderl  = ader * l
          aderzl = ader * zlu
          fp     =  dexp(-aderd) - dexp(-aderl)
          f3a    = (alphay*ader + ip) * ader2*ader*dexp(aderbz)
          fpf3   =  fp*(1.d0 - dexp(aderzl)) / f3a
          f3inv  = f3inv + wt(i) * fpf3
 200    continue
*
      elseif(aderb .ge. 20.d0) then
        do 300 i = 1, 18
```

```
          ip     = ip + p
          ader2  = (ader1 + ip*ss) /kz
          ader   = dsqrt(ader2)
          aderb  = ader * b
          aderd  = ader * d
          aderl  = ader * l
          aderzl = ader * zl
          aderzu = ader * zu
          fp     =  dexp(-aderd) - dexp(-aderl)
          f3a    = (alphay*ader + ip) * ader2*ader * dexp(aderb)
          fpf3   = fp * (dsinh(aderzu)-dsinh(aderzl)) / f3a
          f3inv  = f3inv + wt(i) * fpf3
 300     continue
*        f3inv = 2.d0 * f3inv
      else
        bd = b - d
        bl = b - l
        do 400 i = 1, 18
          ip     = ip + p
          ader2  = (ader1 + ip*ss) /kz
          ader   = dsqrt(ader2)
          aderb  = ader * b
          aderbd = ader * bd
          aderbl = ader * bl
          aderzl = ader * zl
          aderzu = ader * zu
          fp     = (dsinh(aderbd) - dsinh(aderbl)) /dsinh(aderb)
          f3a    = (alphay*ader*dsinh(aderb) + ip*dcosh(aderb))
          fpf3   = fp*(dsinh(aderzu) - dsinh(aderzl))/f3a/ader2/ader
          f3inv  = f3inv + wt(i) * fpf3
 400     continue
      endif
        dwork(ir,1) = p * f3inv * a
        return
      end
************************************************************************
      subroutine ppf3fp(a, dwork, nrel, ir)
************************************************************************
**  This subroutine is called by dhankl.  It is used in computing         *
**  the unconfined (watertable) effect when the pumping well is          *
**  partially penetrating (d ne 0 or l ne b) and the observation well    *
**  is fully screened (zl = 0 and zu ne b).                               *
************************************************************************
```

```
      double precision a, dwork(801, nrel), p, ip, wt(18),
     +     kr, ss, kz, sy,
     +     ader, ader1, ader2, aderb, aderbd, aderbl, aderd, aderl,
     +     alphay, b, bd, bl, d, f3a, f3inv, fp, fpf3, 1, t, zl, zu
*
      common /c1/ kr, ss, kz, sy
      common /c2/ b, p, t, zl, zu
      common /ld/ 1, d
      common /weight/ wt
*
      ader1  = a * a * kr
      ader2  = (ader1 + p*ss) /kz
      ader   = dsqrt(ader2)
      aderb  = ader * b
      alphay = kz / sy
*
      f3inv = 0.0d0
      ip    = 0.0d0
*
      if(aderb .ge. 20.d0) then
        do 100 i = 1, 18
          ip     = ip + p
          ader2  = (ader1 + ip*ss) /kz
          ader   = dsqrt(ader2)
          aderb  = ader * b
          aderd  = ader * d
          aderl  = ader * 1
          fp     =  dexp(-aderd) - dexp(-aderl)
          f3a    = (alphay*ader + ip)*ader2*ader
          fpf3   = fp / f3a
          f3inv  = f3inv + wt(i) * fpf3
  100    continue
*
      else
        bd = b - d
        bl = b - l
        do 400 i = 1, 18
          ip     = ip + p
          ader2  = (ader1 + ip*ss) /kz
          ader   = dsqrt(ader2)
          aderb  = ader * b
          aderbd = ader * bd
          aderbl = ader * bl
          fp      = (dsinh(aderbd) - dsinh(aderbl))
          f3a     = (alphay*ader*dsinh(aderb) + ip*dcosh(aderb))
          fpf3    = fp/f3a/ader2/ader
          f3inv   = f3inv + wt(i) * fpf3
 400      continue
      endif
        dwork(ir,1) = p * f3inv * a
        return
      end
************************************************************************
************************************************************************
************************************************************************
************************************************************************
      block data
*
**  Weighting factors for Stehfest Method.
*
      common /weight/ wt
      double precision wt(18)
      data wt
     *    / 0.49603174603174603132D-04, -0.60957341269841280873D+00,
     *      0.27459404761904761472D+03, -0.26306956746031748480D+05,
     *      0.95725720138888887595D+06, -0.17358694845833331347D+08,
     *      0.18242122264722222090D+09, -0.12185332883091268539D+10,
     *      0.54916800252830352783D+10, -0.17362131115206844330D+11,
     *      0.39455096903527381897D+11, -0.65266516985175003052D+11,
     *      0.78730068328220825195D+11, -0.68556444196120834351D+11,
     *      0.41984343475053573608D+11, -0.17160934711839284897D+11,
     *      0.42045500391026787758D+10, -0.46717222656696426868D+09/
*
      end
**************************************************************************
**************************************************************************
```