# APPENDIX A

# nSIGHTS VERSION 2.41A

Functional Description and Theoretical Development.

Companion to nSIGHTS, Version 2.41a Design Document

ERMS #555650

## TABLE OF CONTENTS

## LIST OF FIGURES



## LIST OF TABLES

# 1 INTRODUCTION

nSIGHTS Version 2.41a (**n** dimensional **S**tatistical **I**nverse **G**raphical **H**ydraulic **T**est **S**imulator) is a comprehensive well test analysis software package that provides a user-interface, a well test analysis model and many tools to analyze both field and simulated well-test data. It was designed in a modular fashion with segregation of source code modules based on functionality. The nSIGHTS code consists of two independent applications: nPre and nPost. The two applications differ in function, but are similar in their interface. nPre assists the user in model set-up, data pre-processing, running of the model and diagnostics of the simulation. nPost post-processes results calculated in nPre and stored in post-processing files. The code is primarily written in Visual C++ Version 6.0, with some graphics and low-level functions written in C and 80386 Assembler.

This Appendix A offers a complete description of the functionality and theoretical basis of nSIGHTS version 2.41a, the most recent release. The following gives a brief overview of each area:

**Section 2 - Functional Description** This section provides an overview of nSIGHTS's capabilities. It includes a summary of the types of hydrogeologic systems considered, the processes and boundary conditions which can be simulated, the data pre- and post-processing capabilities, and the parameter optimization and sampling capabilities.

**Section 3 - Theoretical Development** The mathematical bases of nSIGHTS's functionality are developed in this section. The section includes: a description of the graph theoretical approach to developing field models, the equations describing system geometry, the equations for fluid and gas flow processes, the approaches used for solution of linear and non-linear matrix equations, data processing procedures, a description of the implementation of the optimization and error analysis capabilities, and an overview of the sampling functionality.

# 2 FUNCTIONAL DESCRIPTION

This section provides an overview of nSIGHTS's capabilities and limitations.

## 2.1 Assumptions

nSIGHTS was designed as a general purpose numeric tool for simulating single-phase borehole hydraulic (liquid) and gas tests in a variety of media, particularly low-permeability media where conventional well-test analyses procedures are inappropriate. nSIGHTS's formulation is based on the following assumptions:

- the formation to be analysed can be a confined or unconfined horizontal aquifer of finite radius ($r_o$).
- initially, the formation is at a constant pressure.
- the formation is completely saturated (liquid tests) or completely unsaturated (gas tests);
- formation fluid properties are constant throughout the formation, or can be defined as a function of pressure. In some cases, formation properties can be defined as a function of time. Some properties can be anisotropic. A dual porosity option is available..
- for 1D radial discretizations, the formation can be bounded above and below by impermeable or leaky boundaries; for 2D confined discretizations the upper and lower boundaries are impermeable; for unconfined simulations the upper boundary is the water table and the lower boundary is impermeable.
- Vertical flow is possible for certain cases, such as those involving partial penetration of wells.;
- a single well or borehole at the center of the formation is the only external influence on formation response.
- external boundary conditions (at r = ro) can be zero flow or fixed pressure.
- wellbore boundary conditions include specified flow and specified pressure with or without wellbore storage.
- the formation is at constant temperature throughout the duration of the simulation. Non-isothermal test-zone effects are incorporated for liquid phase tests only.
- numeric discretization is 1D or 2D radial. Two-dimensional radial discretizations are required for unconfined or for confined with vertical anisotropy or partially penetrating wells. Two-dimensional discretizations are available for liquid simulations only. Vertical flow is possible in 2D discretizations.

Some of these assumptions may limit the applicability of nSIGHTS in specific situations. However, limitations due to these assumptions are generally not significant for the analysis of tests conducted in low-permeability media.

## 2.2 System Geometry

For most applications, nSIGHTS is formulated assuming a one-dimensional radial geometry. In this case one-dimensional refers only to the fact that pressure is calculated at specific radial

distances from the borehole, and that flow within the formation is either towards or away from the borehole (i.e. vertical flow within the formation is not considered). The actual physical geometry of the formation being simulated is of course three-dimensional. The "shape" of the formation is governed by its flow-dimension (Barker, 1988), $n$, where $n$ = 1 corresponds to a linear system (flow system area is independent of distance from the borehole), $n$ = 2 is a traditional radial flow system bounded above and below by parallel aquitards, and $n$ =3 is a spherical system. The flow dimension of the system may also be fractional with a range from -50 (sub-linear) to 50 (super hyper-spherical).

An optional configuration allows for a second one dimensional flow system to be coupled with the radial system. This second system is intended to model flow through a sealing plug in the borehole. Figure 2.1 shows the available configurations.

## 2.3 Heterogeneity

nSIGHTS treats the formation (see Figure 2.1) as a zone with either one or two sets of material properties. In the latter case, a "skin-zone" is placed between the test-zone and the remainder of the formation. The skin zone is of a specified thickness $t_s$ as shown in Figure 2.2.

nSIGHTS also allows for heterogenous properties through use of functional approximations for certain parameters. In these cases, parameter values may be represented as a function of distance from the centre of the test-zone.

## 2.4 Parameter Non-Linearity

Certain formation, fluid-property, and test-zone parameters can be defined as functions of pressure. This causes the overall system of equations to become non-linear, requiring solution with a non-linear matrix solution technique. Non-linear parameters cannot be used in dual-porosity (see Section 2.5) simulations.

## 2.5 Liquid System Flow Porosity

nSIGHTS implements both single- and dual- porosity flow processes in liquid flow systems.

In the single-porosity system, there are two types of flow within the formation itself: advective flow, and flow due to storage. Advective flow is the physical movement of formation fluid either toward or away from the borehole, as dictated by pressure gradients. Flow due to storage involves fluid flow due to compressibility of the formation and the formation fluid itself as driven by pressure changes with time.

In dual-porosity systems, the flow domain is divided by a volumetric proportion constant into two connected systems: the fracture system, and the matrix system. Flow in the fracture system is analogous to flow in a single-porosity system with the addition of a third flow type, flow into/out of the adjacent matrix material. Flow within the matrix can be visualized as a number of

individual single-porosity systems with zero-flow external boundaries. Each matrix system is connected only to the adjacent fracture material. The geometry of the fracture/matrix connection and the behaviour of the internal matrix flow systems are controlled by user-entered parameters.

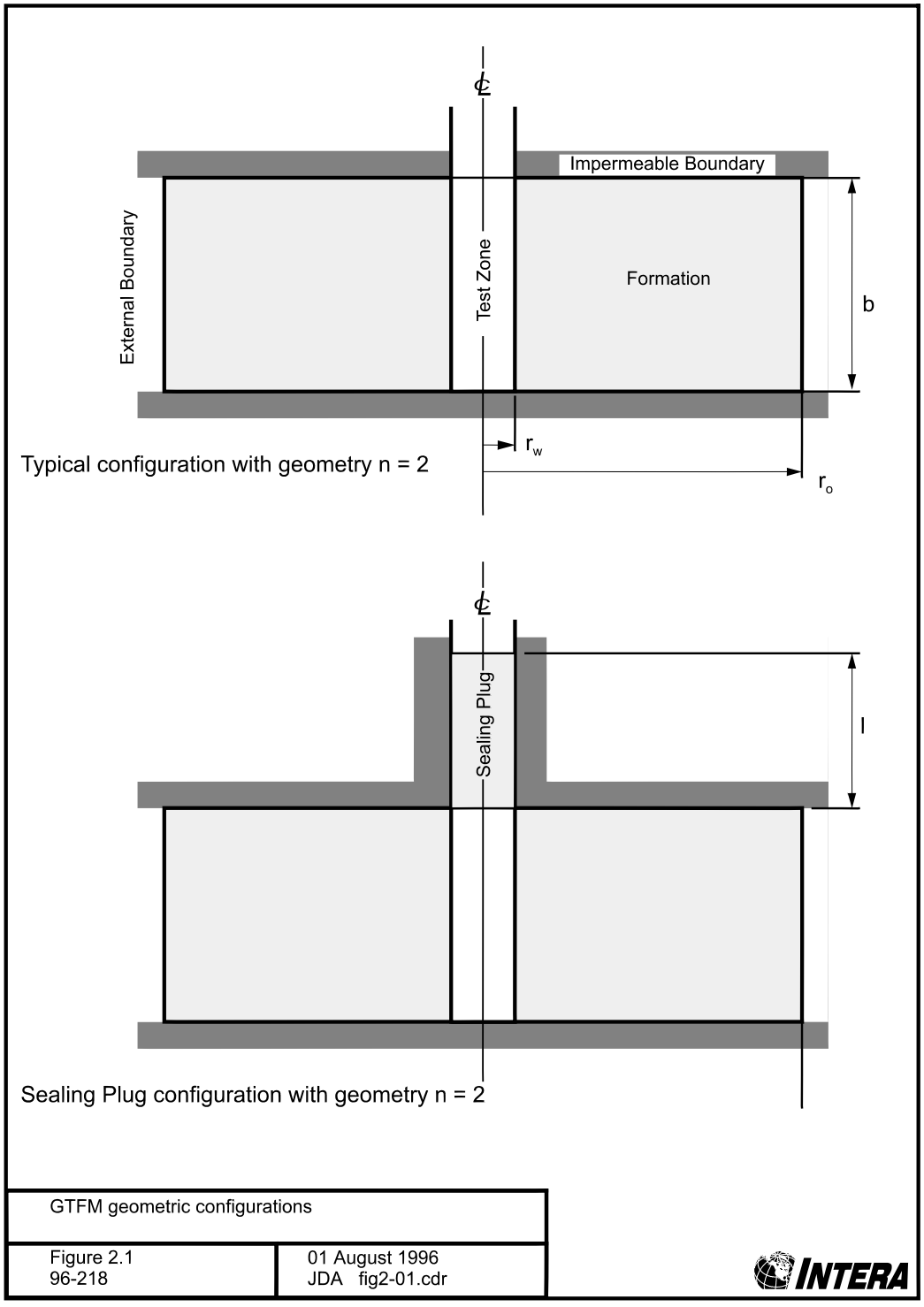



**Figure 2.1 nSIGHTS geometric configurations (identical to GTFM documentation shown).**

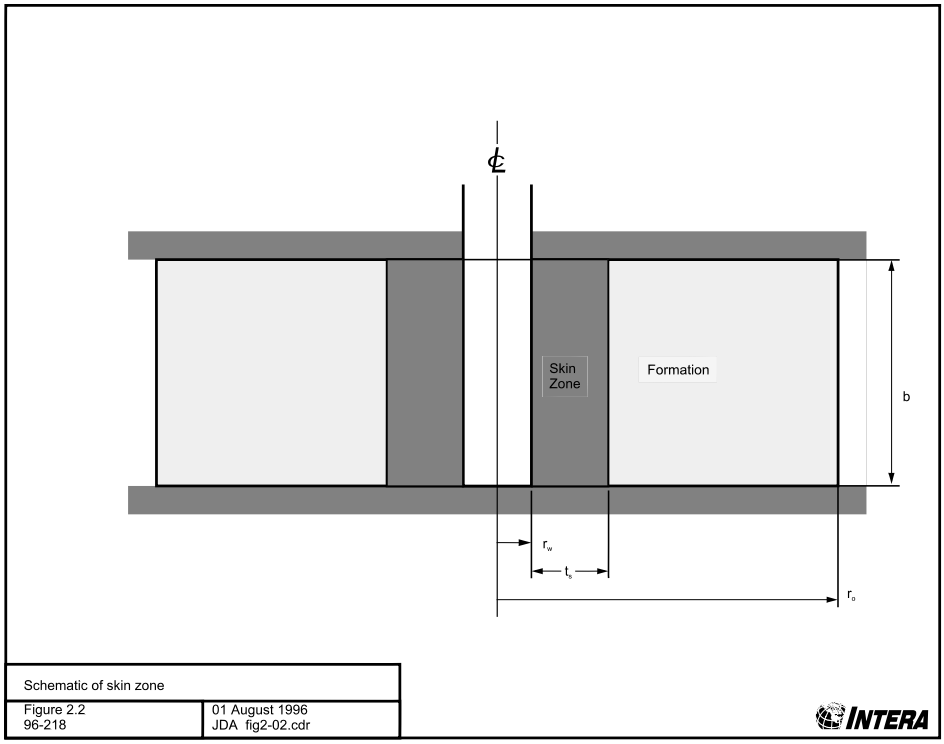



**Figure 2.2 Schematic of skin zone**

## 2.6 Gas System Flow Porosity

The gas flow processes implemented in nSIGHTS are for single-porosity systems only. As stated in Section 2.4, the current implementation of the dual-porosity flow processes (see Section 3.1.5.2) is suitable for linear equations only. As presented in Section 3.1.4.2, the equations describing advective gas flow are intrinsically non-linear. Thus, the dual-porosity formulation cannot currently be used in gas test analyses.

## 2.7 Boundary Conditions

Boundary conditions are applied at: the test-zone, the external formation boundary, and, if a sealing plug is included, at the top of the plug. The following sub-sections describe available boundary conditions.

### 2.7.1 Test-Zone Boundary Conditions

Boundary conditions at the test-zone represent those commonly applied during a well- or borehole-test. nSIGHTS allows different boundary conditions to be applied consecutively, so as to simulate actual test conditions. Available test-zone boundary conditions include:

1) Specified flow - a flow rate to be injected into (+ve) or withdrawn from (-ve) the test-zone is specified. The flow rate can be described as a constant, or as an arbitrary function of time (see Section 3.2). Well-bore storage may also be incorporated in the boundary condition. Two types of well-bore storage are available: open-hole (for liquid systems only), where a tubing string of constant diameter filled with liquid assumed to be connected to the test-zone, and isolated, where the test-zone is filled with a compressive liquid or a gas.

2) Specified pressure, or history - the pressure in the test-zone is specified as a constant or as a function of time.

3) Slug withdrawal/injection (liquid tests only) - The injection or withdrawal of liquid from an open borehole. The response of the formation will be indicated by the changes in liquid level subsequent to the injection/withdrawal as liquid flows into the formation (injection) or out of the formation (withdrawal). The pressure in the wellbore at the test horizon will vary according to the height and density of the column of liquid above the test zone.

4) Pulse injection/withdrawal - a pressure pulse or withdrawal is applied to a physically isolated test zone. Pressure will rise or decay in the test zone according to the formation parameters and the compressibility of the test-zone. For liquid phase tests, test-zone compressibility is usually higher than the fluid compressibility due to equipment compliance effects. For gas phase tests, equipment compliance is assumed negligible relative to the compressibility of the gas.

### 2.7.2 External Boundary Conditions

nSIGHTS provides zero-flow, fixed-pressure, and Carter-Tracey infinite acting boundary conditions to be applied at the external boundary. Carter-Tracey boundary conditions were not used in conducting WIPP PA analyses and are therefore not addressed in this manual.

The pressure value used for fixed-pressure boundary conditions is the user-entered value for static or initial (pre-test) formation pressure.

### 2.7.3 Sealing Plug Boundary Conditions

A fixed-pressure boundary condition is applied at the top of the sealing plug. For liquid systems, the pressure is set at 0.0 Pa. For gas systems, the pressure is set to the user-entered value for atmospheric pressure.

## *2.8 Data Pre- and Post-Processing*

nSIGHTS contains pre-processing facilities which support a number of input data conditioning and standard well-test interpretation diagnostics. Input data conditioning routines include manual and automatic data reduction, data smoothing, and low- and high-pass filtering. Diagnostic utilities include calculation of pressure derivatives and superposition times. Interpretation tools for graphical calculation of line slopes are also available.

Standard post-processing facilities include diagnostic plots that can be used to compare field data and simulation results. Other graphical facilities allow for 2D and 3D visualization of the simulated pressure versus distance into the formation and time.

## *2.9 Goodness-of-Fit Calculation*

nSIGHTS can also determine a measure of the quality of the match between field data and simulation results. Processed field data (i.e. pressure derivatives) can also be compared to processed simulation results and a fit metric determined. The standard fit metric is the Chi-squared value, or sum of the squared residuals. Other available metrics include sum of absolute values of residuals, or the average (normalized to number of points in the field data set) sum of squared or absolute values of residuals. Compound metrics can be calculated by adding two or more single fit calculations together. Components of compound metrics can be scaled individually to change weighting and contribution to overall fit.
Fit surfaces can be generated by calculating fit metrics while varying the values of two simulation parameters. These surfaces can be visualized as 3D surfaces, as contour plots, or as XY plots of fit value versus a single parameter value.

## 2.10 Optimization

nSIGHTS's optimization facilities automatically determine the set of simulation parameters required to minimize a fit metric, or, in other words, to create the best match between field and simulation results. Optimizations require that two or more parameters be specified for optimization. In addition to individual parameters, the Y values of points defining parameters as a function of radius or pressure may also be optimized. Up to 10 parameters, or 20 individual Y components, or any combination of the two may be optimized.

Two minimization algorithms are available. The Simplex algorithm is robust and can be used to minimize any fit metric. The Levenberg-Marquardt algorithm is restricted to Chi-squared fit metrics but can converge on a solution quickly, particularly if the fit metric is well behaved.

Although any of the fit metrics described in section 2.9 can be minimized, the Chi-squared metric is of special significance as it allows for a number of error analyses routines. These include residual analysis, confidence intervals, covariance assessment and Jacobian evaluation to determine the significance of each parameter's contribution to the fit metric.

## 2.11 Sampling

One of nSIGHTS's unique features is the optimization/sampling mode of operation which combines optimization with Latin Hypercube or Monte Carlo variable sampling routines. With sampling, uncertain variables (certain wellbore boundary conditions and model parameters) are assigned distributions, and numerous realizations of input data sets generated. An optimization is performed on each input data set. Optimized parameter results are stored in a data file. These can be post-processed to show ranges and distributions of optimized parameters, and to determine correlations between input parameter uncertainty and optimized parameter ranges.

# 3 THEORETICAL DEVELOPMENT

This section of the manual describes the theoretical and mathematical basis for the functionality available in nSIGHTS. The section is subdivided to address the simulator itself, methods for approximating functional representations, pre-processing and analysis routines, and optimization, sampling and error analysis.

## *3.1 Simulator Theory*

This subsection describes the graph theoretic approach as applied to field models, develops the equations describing system geometry and flow, formulates the algebraic system of equations, and finally, describes solution techniques used for linear and non-linear cases.

### 3.1.1 General Approach

nSIGHTS is based on an application of graph theory as applied to field models. This approach, described in Savage and Kesavan (1976), applies a discretization procedure to a physics model to produce a system of algebraic equations which are amenable to computer solutions. In this sense, it differs from finite element and finite difference models which rely on an intermediary step whereby a continuous model formulation (usually a partial differential equation) is derived from the physics of the problem before discretization is performed.

With nSIGHTS, the physical geometry of the aquifer being simulated is represented as a geometric continuum with a finite volume. This volume is discretized into smaller volumes, each represented by a single node. Adjacent nodes are connected by directed graph edges which represent flow from one volume to another. The direction of each graph edge is indicated by an arrow which represents the assumed direction of positive flow. Additional graph edges represent flow within each volume and flow into or out of the system at the boundaries. In a fluid flow system, these three types of graph edges represent Darcy flow, flow due to storage, and boundary flows respectively. Figure 3.1 illustrates these concepts schematically for a single porosity formation with no skin or sealing plug.

The formulation assumes that the pressure within each nodal volume is represented by a pressure value at each node. Pressures may change as a function of node number and time. The time continuum is also discretized into a series of time steps.

Figure 3.2 shows a segment of the directed graph within a flow system similar to that shown in figure 3.1. Nodes $i, j, k$ are three consecutive nodes. Flow between nodes $i$ and $j$ is represented by the graph edge labelled $d_i$. Flow due to storage within the volume represented by node $j$ is represented by the graph edge labelled $s_j$.

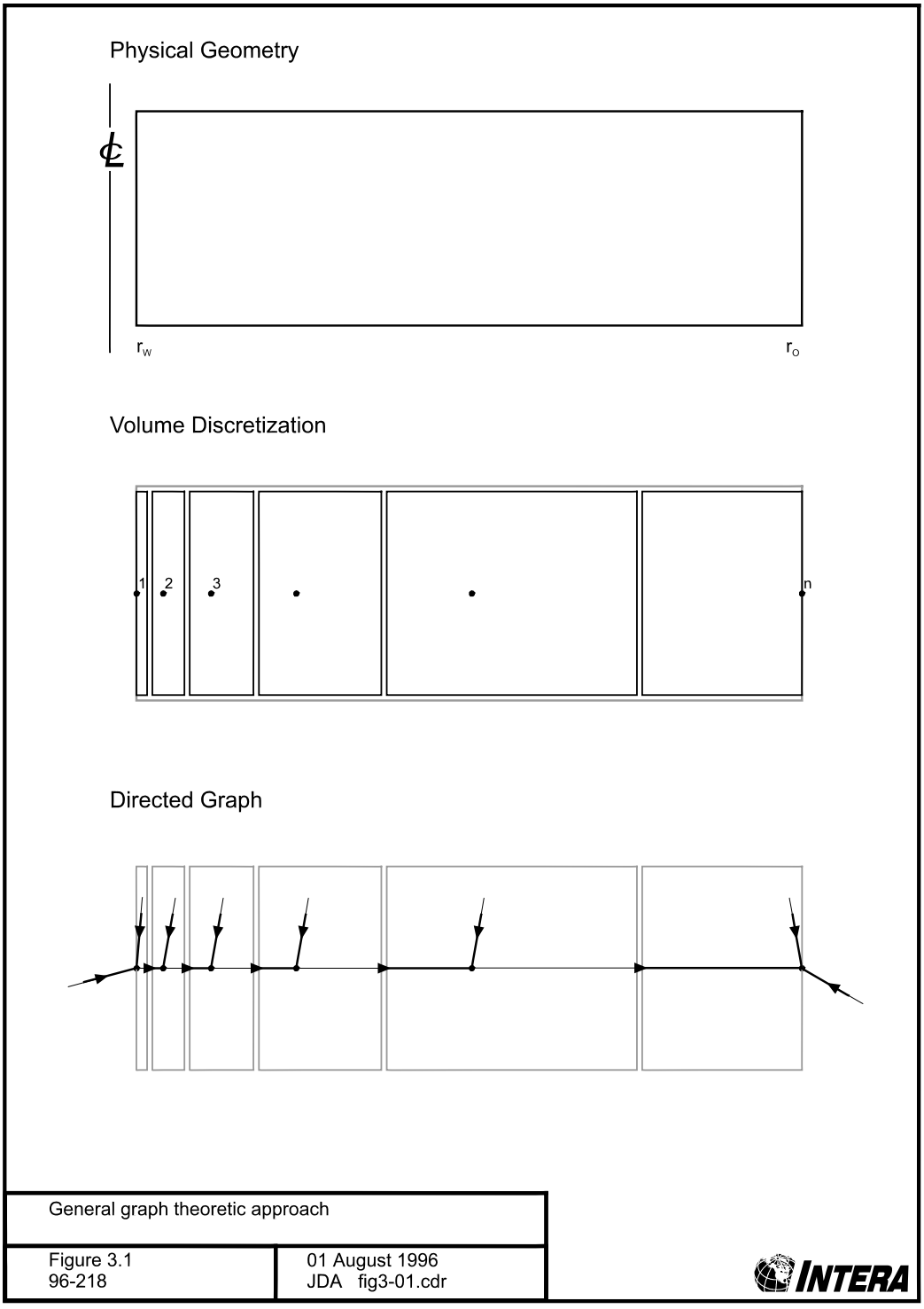



**Figure 3.1 General graph theoretic approach**

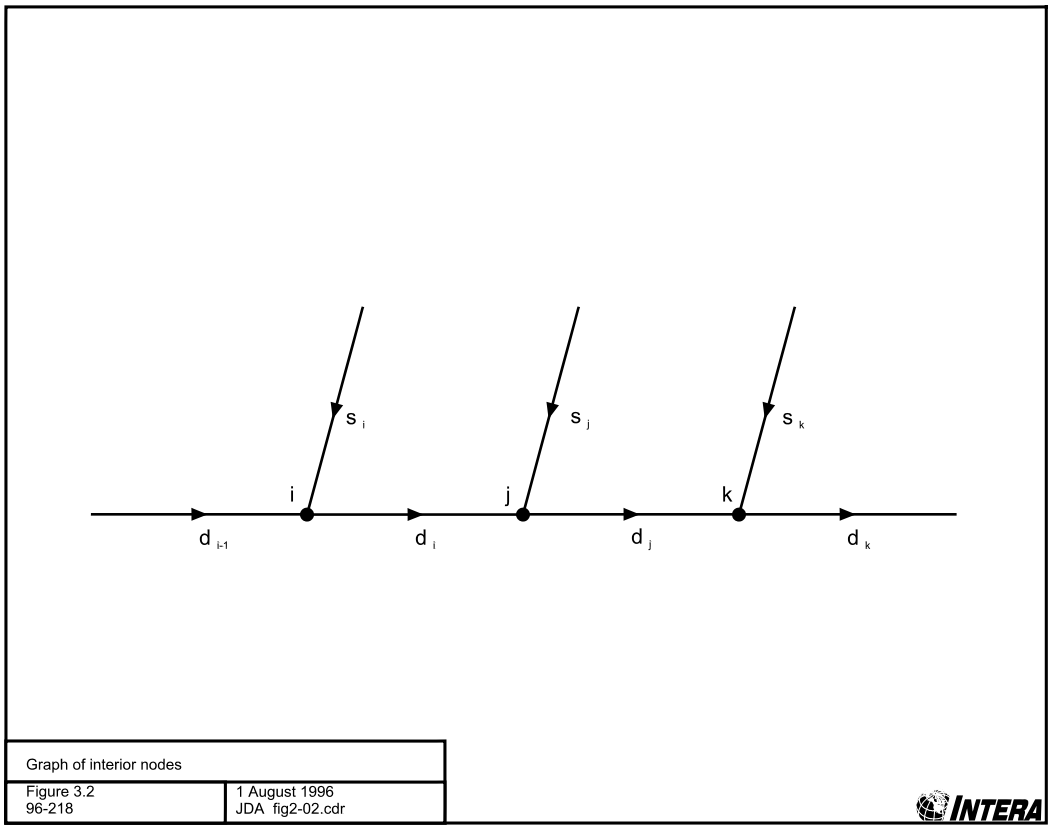



**Figure 3.2 Graph of interior nodes**

Constitutive relationships hold at each node. For example, at any given time $t$, at any interior

$$d_{i-1} + s_i - d_i = 0 \qquad (3.1\text{-}1)$$

node $i$:
When coupled with geometric relationships and the flow equations developed in subsequent sub-sections, the constitutive relationships can be used to create a set of algebraic equations describing the system. Solving the equations yields nodal pressures and graph edge flow rates.

### 3.1.2 System Graphs

This subsection develops the system graphs used in different geometrical and system porosity configurations.

Figure 3.3 shows a basic system formulation - single porosity, no skin, and no sealing plug. The graph edges labelled $b_w$ and $b_o$ represent boundary flows: from the well into the formation, and from the formation out of the external boundary respectively. A logarithmic spacing is used between adjacent nodes. This serves to increase the node density close to the well, where pressure changes are likely to be most rapid. The dashed areas delimit the volume assigned to each node and indicate the radius at which the flow area between nodes is calculated.

Figure 3.4 depicts a sealing plug added to the basic system. Node 1 is located at the top of the plug, while node $n_p$ is the last node within the plug above the well-bore. Pressure in the isolated test-zone is assumed to be uniform, and the pressure of the node at $r_w$ is assumed also to be representative of the pressure in the test-zone at the bottom of the plug.

Figure 3.5 shows the effect of adding a skin to the basic system. Note that graph edges remain conceptually the same, however the spacing of nodes within the skin zone is obviously different from that in the formation.

Figure 3.6 is a dual porosity system with no skin or sealing plug. The total system is divided into two components: fracture and matrix. The relative volumes of the components are determined by a “matrix volume factor” where:

$$V_m = V_s M$$
$$V_f = V_s (1.0 - M) \qquad (3.1\text{-}2)$$

where:

$V_m$ = volume of matrix, $L^3$
$V_f$ = volume of fracture, $L^3$
$V_s$ = volume of system, $L^3$
$M$ = matrix volume factor

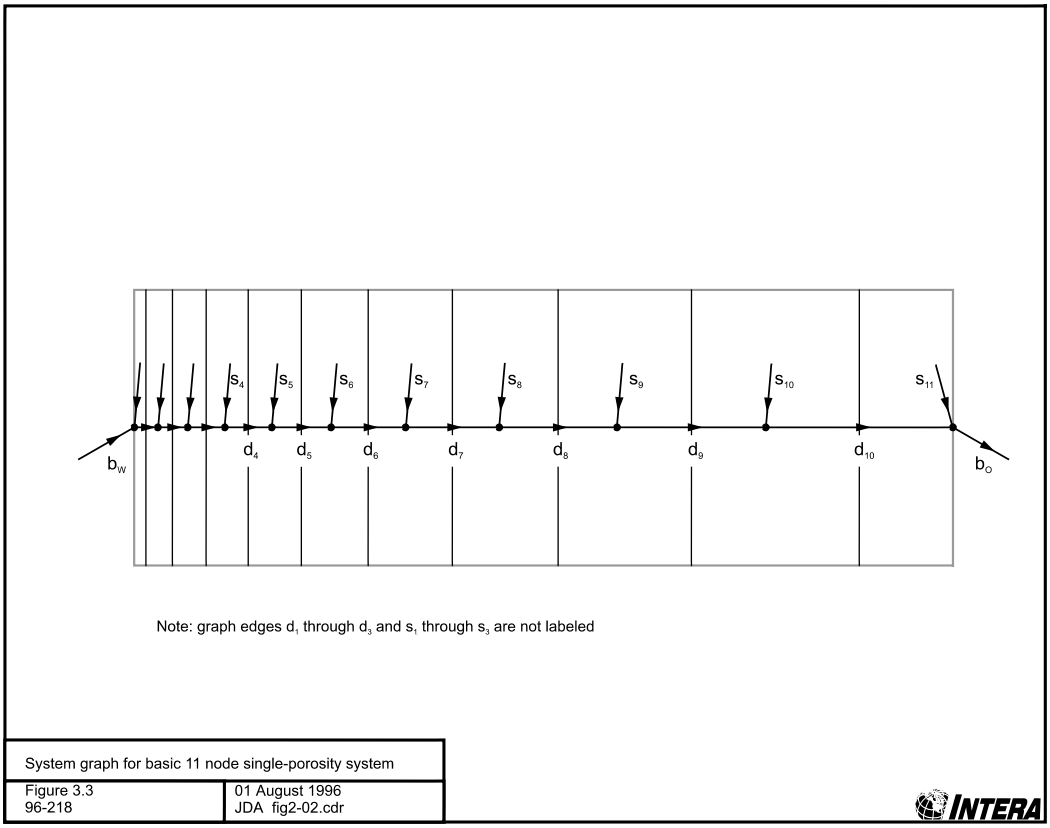



**Figure 3.3 System graph for basic 11 node single-porosity system**

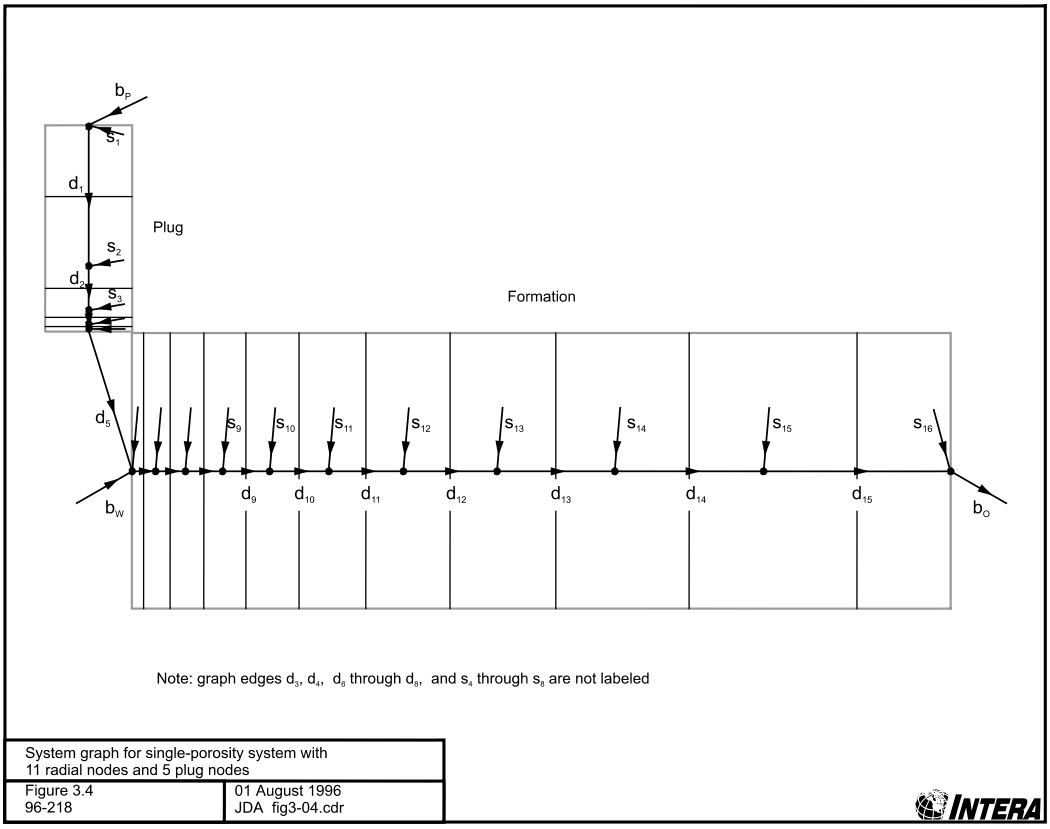



**Figure 3.4 System graph for single-porosity system with 11 radial nodes and 5 plug nodes**

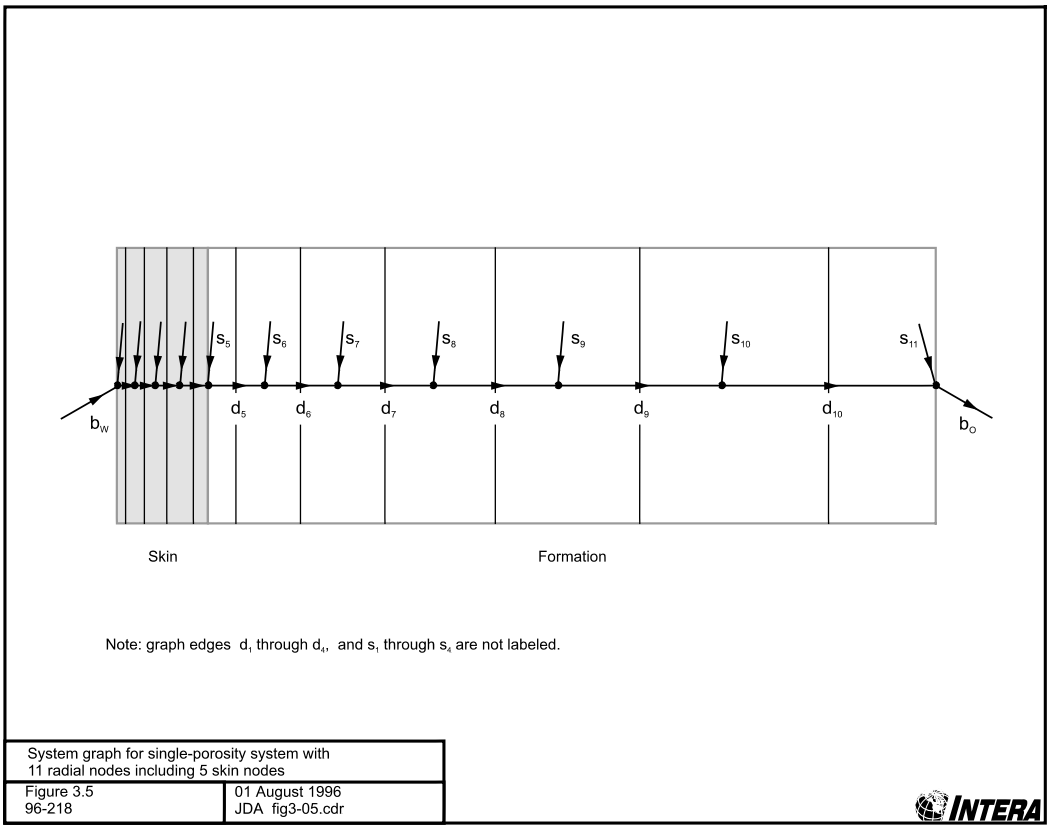



**Figure 3.5 System graph for single-porosity system with 11 radial nodes including 5 skin nodes**

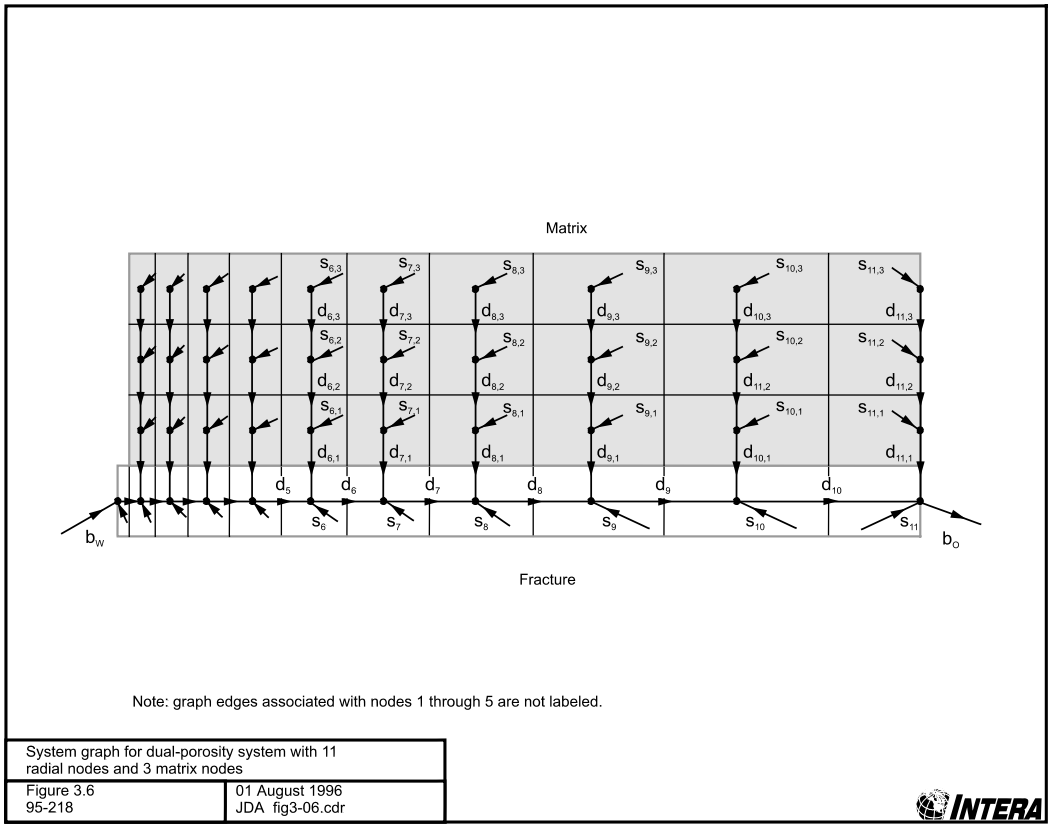



**Figure 3.6 System graph for dual-porosity system with 11 radial nodes and 3 matrix nodes**

Radial flow between nodes and flow in the boundary edges occurs only in the fracture. The matrix flow system can be viewed as a number of 1-D systems with zero flow external boundaries coupling into the fracture system. The number of nodes in the matrix system controls the degree of "transientness" of the double porosity response. A single node corresponds to a pseudo-steady-state (PSS) response, while more than ten nodes yields a "transient" response. PSS and "transient" are terms from the oilfield literature which describe different dual-porosity reservoir responses.

Figure 3.7 shows a system with dual-porosity and skin. The figure indicates that the dual-porosity systems are not incorporated in the skin zone, and that the full volume of the system is assigned to the fracture system within the skin-zone.

### 3.1.3 System Geometry

There are two elements to specifying system geometry: the spacing of the nodes in the system, and the volumes and areas associated with each node.

#### 3.1.3.1 Node Spacing

For a system without a skin a simple logarithmic spacing is used for all nodes within the formation:

$$r_i = r_{i-1} e^{\Delta a} \qquad (3.1\text{-}3)$$

where:

$r_i$ = radius of node i, L

$\Delta a$ = logarithmic grid increment

where:

$$\Delta a = \frac{\ln(r_o) - \ln(r_w)}{n-1} \qquad (3.1\text{-}4)$$

where:

$r_o$ = outside radius of grid, L

$r_w$ = radius of borehole or well, L

$n$ = number of nodes in grid

For systems with a skin, two grid increments are used. Within the skin (i.e. for nodes 1 through $n_s$), equation 3.1-5 applies:

$$\Delta a_s = \frac{\ln( r_w + t_s ) - \ln( r_w )}{n_s - 1} \qquad (3.1\text{-}5)$$

where :

$t_s$ = thickness of skin, L

$n_s$ = number of nodes in skin

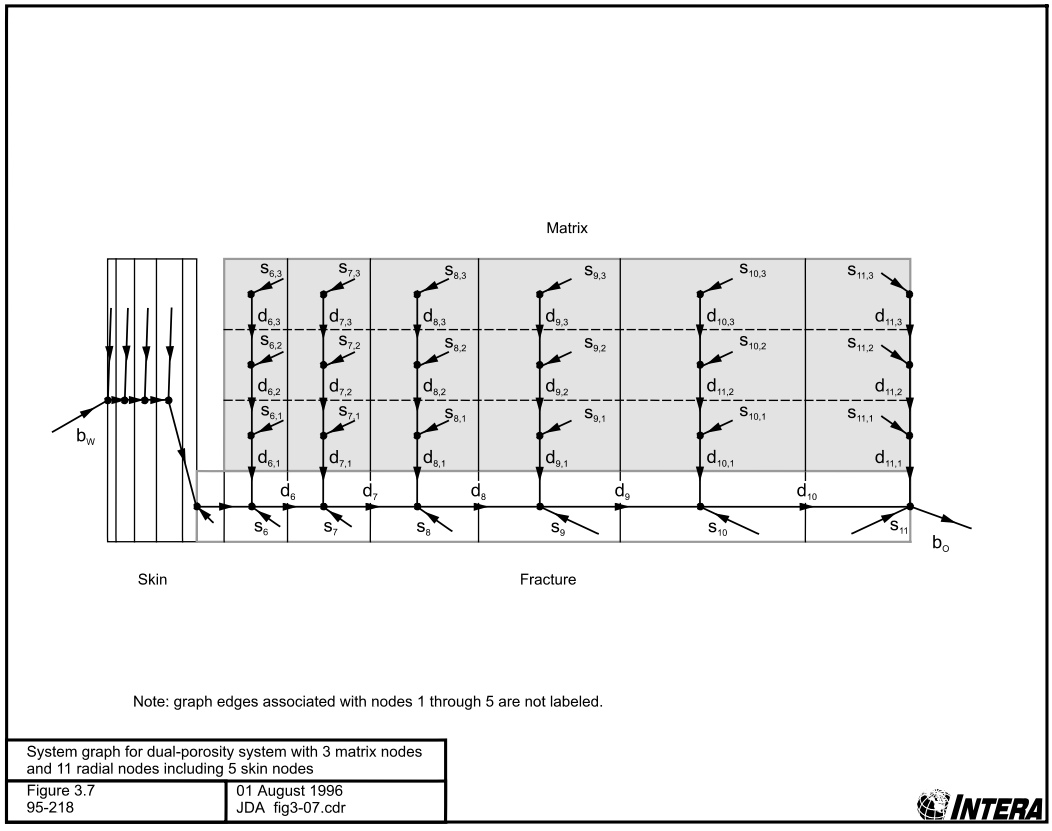



**Figure 3.7 System graph for dual-porosity system with 3 matrix nodes and 11 radial nodes including 5 skin nodes**

Outside the skin (nodes $n_{s+1}$ to node $n$), the grid increment described by equation 3.1-6 is used:

$$\Delta a = \frac{\ln( r_o ) - \ln( r_w + t_s )}{n - n_s - 1} \qquad (3.1\text{-}6)$$

If a sealing plug is included, nodes within the sealing plug are also geometrically spaced. However, in this case a "plug spacing factor" is entered, from which the grid increment is derived:

$$\Delta a_p = \frac{\ln( X_{plug} )}{n_p - 1} \qquad (3.1\text{-}7)$$

where :

$X_{plug}$ = plug spacing factor

$n_p$ = number of nodes in plug

Location of nodes within the plug is then described by:

$$l_i = L_{plug} \left[1 - e^{-\Delta a (i - 1)}\right] \qquad (3.1\text{-}8)$$

where :

$l_i$ = distance of node i from top of plug, L

$L_{plug}$ = length of plug, L

### 3.1.3.2 Flow Areas

The flow area is defined as the area over which advective, or Darcy, flow between adjacent nodes

$$A_i = b \; 2 \; \pi \; r_{a_i} \qquad (3.1\text{-}9)$$

where :

$A_i$ = flow area between nodes i and i + 1, $L^2$

b = thickness of formation, L

$r_{a_i}$ = average radius between node i and i+1, L

$$= \frac{r_i + r_{i+1}}{2}$$

occurs. In a single porosity radial system of flow dimension equal to 2, the flow area is simply:

In a generalized system of flow dimension n, the equation from Barker (1988) describes the flow area of the nodes in the fracture zone (i.e. non-matrix, non-plug nodes}:

$$\alpha_i = \frac{2\ \pi^{n/2}}{\Gamma(n\ /\ 2)}$$

$$A_i = b^{3-n}\ \alpha_i\ r_{a_i}^{n-1} \qquad (3.1\text{-}10)$$

where :

n = system flow dimension

$\Gamma$ = gamma function

Equations 3.1-9 and 3.1-10 are identical for $n = 2$. However, when simulating actual tests, where the flow area at the well is as specified by equation 3.1-9 with $r_{ai} = r_{well}$ , using equation 3.1-10 leads to incorrect flow areas being used. Consequently, nSIGHTS allows for compensation of the area. Compensation is used whenever the flow dimension is specified as a function of radius, or if the well-bore area must match a specified borehole area. In these cases, the area is calculated from the area of the previous node, and the flow dimension at the node :

$$A_i = \frac{A_{i-1}}{r_{i-1}^{n_i-1}}\, r_{a_i}^{n_i-1} \qquad (3.1\text{-}11)$$

where :

$n_i$ = flow dimension at node i

Within the plug portion of a system the flow area is simply the cross-sectional area of the borehole:

$$A_p = \pi\ r_w^2 \qquad (3.1\text{-}12)$$

where :

$A_p$ = flow area of all plug nodes, $L^2$

Dual-porosity systems always assume a flow dimension of 2 in the fracture. The flow area in the matrix nodes associated with each fracture node is the area of the fracture system in contact with the matrix:

$$A_{i_m} = \frac{\pi}{4}\left[\left(r_i + r_{i+1}\right)^2 - \left(r_{i-1} + r_i\right)^2\right] \qquad (3.1\text{-}13)$$

where :

$A_{i_m}$ = flow area of all matrix nodes associated with fracture node i, $L^2$

### 3.1.3.3 Node Volumes

The volumes assigned to each node are used in formulating the equations describing the graph edges representing the contribution of storage flow to the system.

For nodes in the fracture zone the volume assigned to node *i* is calculated as follows:

$$V_i = V_{A_i} - V_{A_{i-1}} \qquad (3.1\text{-}14)$$

where :

$V_{A_i}$ = volume enclosed within flow area i

$V_A$ are determined by integrating equation 3.1-10 or 3.1-11 with respect to *r:*

$$V_{A_i} = \int A_i \, dr_{a_i}$$

or

$$V_{A_i} = \frac{A_i}{n_i} r_{a_i} \qquad (3.1\text{-}15)$$

Plug node volumes are:

$$V_{i_p} = \frac{A_p}{2}\left[ l_{i+1} - l_{i-1} \right] \qquad (3.1\text{-}16)$$

Matrix node volumes are described as:

$$V_{m_i} = \frac{A_{m_i} b M}{n_m} \qquad (3.1\text{-}17)$$

where :

b = formation thickness (assumed constant as f(r)), L

M = matrix volume factor

$n_m$ = number of matrix nodes

## 3.1.4 Edge Equations

Edge equations describe the flow in the graph edges. They are categorized by graph edge type (Darcy, storage, or boundary) and system type (gas or liquid).

### 3.1.4.1 Liquid Flow

Liquid flow equations describe the edge flow in volumetric flow terms ($L^3\ t^{-1}$). Fluid density is assumed to be invariant as a function of pressure so as to ensure that constitutive relationships hold.

Flow in Darcy edges is described as:

$$d_i = \frac{k}{\mu} A_i \frac{\Delta P}{\Delta l}$$

or

$$d_i = \frac{K}{\rho\ g} A_i \frac{\Delta P}{\Delta l} \qquad (3.1\text{-}18)$$

where :

$k$ = permeability, $L^2$

$u$ = viscosity, $ML^{-1}t^{-1}$

$K$ = hydraulic conductivity, $Lt^{-1}$

$\rho$ = formation fluid density, $ML^{-3}$

$g$ = gravitational constant, $Lt^{-2}$

$\Delta P$ = change in pressure between nodes i and i+1, $ML^{-1}t^{-2}$

$\Delta l$ = distance between nodes i and i+1, L

Permeability and/or viscosity or hydraulic conductivity may be described as functions of pressure. Alternatively, for non-plug, non-matrix nodes, permeability and hydraulic conductivity may also be described as functions of radius.

For matrix nodes, the Darcy flow equation is slightly modified to account for different matrix geometries:

$$d_{i,j} = \alpha \frac{k}{\mu} A_{m_i}\ n_m\ t\ \Delta P$$

or

$$d_{i,j} = \alpha \frac{K}{\rho\ g} A_{m_i}\ n_m\ t\ \Delta P \qquad (3.1\text{-}19)$$

where :

$\alpha$ = interporosity flow parameter (shape factor), $L^2$

$t$ = unit thickness, L

Flow due to storage is described as:

$$s_i = ( C_r + \theta C_w ) V_i \frac{\Delta P}{\Delta t}$$

or

$$s_i = \frac{S}{\rho g} V_i \frac{\Delta P}{\Delta t}$$

where :

$C_r$ = rock compressibility, $M^{-1} Lt^2$

$\theta$ = rock porosity

$C_w$ = formation fluid compressibility, $M^{-1} Lt^2$

$S$ = formation storativity, $L^{-1}$

$\Delta P$ = change in pressure between time steps i and i+1, $ML^{-1} t^{-2}$

$\Delta t$ = size of time step, t

3.1-20

Wellbore boundary flows ($b_w$ edges) vary according to the wellbore boundary condition assigned.

For specified pressure:

$b_w$ = unknown (3.1-21)

For iso-thermal pulse sequences:

$$b_w = V_w C_t \frac{\Delta P_w}{\Delta t}$$

(3.1-22)

where :

$V_w$ = volume of test-zone, $L^3$

$C_t$ = test-zone compressibility, $M^{-1} Lt^2$

$C_t$ may be a constant, a function of time, or a function of pressure. $V_w$ may be a constant or a function of time.

For non iso-thermal pulse sequences:

$$b_w = \frac{V_w}{\Delta t}\left(C_t \Delta P_w - C_T \Delta T_w\right) \qquad (3.1\text{-}23)$$

where :

$C_T$ = test- zone fluid thermal expansion coefficient, $T^{-1}$

$\Delta T_w$ = change in test- zone temperature over a single time step, T

$T_w$ is defined as a function of time.

For slug sequences:

$$b_w = \frac{\pi r_c^2}{\rho g}\frac{\Delta P_w}{\Delta t} \qquad (3.1\text{-}24)$$

where :

$r_c$ = radius of casing where liquid level changes are occurring, L

For specified flow with no wellbore storage:
$Q_w$ may be a constant or a function of time.

$$b_w = Q_w \qquad (3.1\text{-}25)$$

where :

$Q_w$ = specified flow , $L^3 t^{-1}$

For specified flow with “isolated” wellbore storage:

$$b_w = Q_w - V_w C_t \frac{\Delta P_w}{\Delta t} \qquad (3.1\text{-}26)$$

For specified flow with “open hole” wellbore storage:

$$b_w = Q_w - \frac{\pi r_c^2}{\rho g}\frac{\Delta P_w}{\Delta t} \qquad (3.1\text{-}27)$$

Flows associated with external boundary conditions, $b_o$, are dependent on the assigned boundary condition:

For specified pressure:

$b_o$ = unknown (3.1-28)

For zero-flow:

$b_o = 0$ (3.1-29)

Flows associated with fixed-pressure plug external boundary conditions, $b_p$, are:

$b_p$ = unknown (3.1.30)

### 3.1.4.2 Gas Flow

Gas flow equations describe the edge flow in mass flow terms ($Mt^{-1}$) or volumetric flow terms ($L^3\ t^{-1}$) at standard temperature and pressure (STP). The two are related by:

$$q_{STP} = q_M \frac{R\ T_s}{m_w\ P_s}$$

where :

$q_{STP}$ = flow rate at standard temperature and pressure, $L^3\ t^{-1}$

$q_M$ = mass flow rate , $M\ t^{-1}$

$R$ = universal gas constant

$m_w$ = molecular weight of gas, $M\ mole^{-1}$

$T_s$ = standard temperature, T

$P_s$ = standard pressure, $M\ L^{-1}\ t^{-2}$ (3.1-31)

Flow in Darcy edges is described as:

$$d_{i_M} = \frac{m_w}{RT} \frac{k}{\mu} A_i\ P \frac{\Delta P}{\Delta l}$$

or

$$d_{i_{STP}} = \frac{T_s}{P_s\ T} \frac{k}{\mu} A_i\ P \frac{\Delta P}{\Delta l} \quad (3.1\text{-}32)$$

where :

$P$ = average pressure between nodes i and i+1, $M\ L^{-1}\ t^{-2}$

$$= \frac{P_i + P_{i+1}}{2}$$

Note that the $P$ term in equation 3.1-32 makes the flow equation intrinsically non-linear. Permeability and/or viscosity can be described as functions of pressure. For non-plug nodes, permeability may be described as a function of radius.

An alternate formulation is available for viscosity:

$$\mu = \mu_0 + b_\mu P \qquad (3.1\text{-}33)$$

where:

$\mu_0$ = viscosity at $P = 0$, $M\ L^{-1}\ t^{-1}$

$b_\mu$ = linear viscosity coefficient, $t^{-1}$

An alternate formulation is also available for permeability:

$$k = k_l\left(1 + \frac{b_k}{P}\right) \qquad (3.1\text{-}34)$$

where:

$k_l$ = permeability to liquid, $L^2$

$b_k$ = Klinkenburg coefficient, $M^{-1}\ t$

Flow due to storage is described as:

$$s_{i_M} = \frac{m_w}{R\,T}\,\theta\,V_i\,\frac{\Delta P}{\Delta t}$$

or

$$s_{i_{STP}} = \frac{T_s}{P_s\,T}\,\theta\,V_i\,\frac{\Delta P}{\Delta t} \qquad (3.1\text{-}35)$$

Boundary condition edge flow equations for pulse tests are:

$$b_{w_M} = \frac{m_w}{R\,T} V_w \frac{\Delta P_w}{\Delta t}$$

or (3.1-36)

$$b_{w_{STP}} = \frac{T_s}{P_S\,T} V_w \frac{\Delta P_w}{\Delta t}$$

Other boundary condition equations are analogous to those developed for liquid systems. Slug wellbore boundary conditions are not available for gas systems.

## 3.1.5 Matrix Equations

### 3.1.5.1 Single Porosity

The basic edge equations can be reduced to the following form:

$$d_i = D_i (P_i - P_{i+1})$$

$$s_i = S_i (P_i - P_{i_{t-\Delta t}})$$

$$b_w = Q_w + B_w (P_w - P_{w_{t-\Delta t}}) \quad (3.1\text{-}37)$$

$$b_o = Q_o$$

where $D_i$, $S_i$, and $B_w$ may be constants or may be functions of pressure ($D_i$ for all gas, otherwise if *f(P)* parameters). $B_w$ may also be a function of time. *Qw*, and *Qo* may be constants or may be functions of time. A fully implicit (backwards in time) time discretization is used in the $s_i$ and $b_w$ equations. Applying constitutive relationships (equation 3.1-1) at node *i* yields:

$$D_{i-1} (P_{i-1} - P_i) + S_i (P_i - P_{i_{t-\Delta t}}) - D_i (P_i - P_{i+1}) = 0$$

or : (3.1-38)

$$D_{i-1}\,P_{i-1} + (-D_{i-1} - D_i)\,P_i + D_i\,P_{i+1} + S_i (P_i - P_{i_{t-\Delta t}}) = 0$$

For a non-single porosity system, applying plug 3.1-38 to the entire system graph leads to the following:

(3.1-39)

$$\begin{bmatrix} -D_1 & D_1 & & & & & \\ D_1 & -D_1-D_2 & D_2 & & & & \\ & D_2 & -D_2-D_3 & D_3 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & D_{n-2} & -D_{n-2}-D_{n-1} & D_{n-1} & \\ & & & & D_{n-1} & -D_{n-1} & \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix} + \begin{bmatrix} S_1 + B_w & & & & \\ & S_2 & & & \\ & & S_3 & & \\ & & & \ddots & \\ & & & & S_{n-1} \\ & & & & S_n \end{bmatrix} \left( \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix}_t - \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix}_{t-\Delta t} \right) = \begin{bmatrix} -Q_w \\ 0 \\ 0 \\ \vdots \\ 0 \\ Q_o \end{bmatrix}$$

Re-arranging yields:

$$\begin{bmatrix} -D_1+S_1+B_w & D_1 & & & & & \\ D_1 & -D_1-D_2+S_2 & D_2 & & & & \\ & D_2 & -D_2-D_3+S_3 & D_3 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & D_{n-2} & -D_{n-2}-D_{n-1}+S_{n-1} & D_{n-1} & \\ & & & & D_{n-1} & -D_{n-1}+S_n \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix} - \begin{bmatrix} -Q_w+(B_w+S_1)P_{1_{t-\Delta t}} \\ S_2 P_{2_{t-\Delta t}} \\ S_3 P_{3_{t-\Delta t}} \\ \vdots \\ S_{n-1} P_{n-1_{t-\Delta t}} \\ Q_o+S_n P_{n_{t-\Delta t}} \end{bmatrix} \quad (3.1\text{-}40)$$

If a plug is included, additional nodes are placed before the well bore node (node 1 in a non-plug system) and a set of equations analogous to 3.1-40 are developed, with node 1 corresponding to the fixed pressure boundary at the top of the plug.

All fixed pressure boundaries (top of plug for plug systems, well bore for history sequences, and external boundary node if not set to zero flow) are implemented by partitioning the matrix and re-formulating for unknown pressures only. For example, a fixed pressure external boundary results in:

$$\begin{bmatrix} -D_1+S_1+B_w & D_1 & & & & \\ D_1 & -D_1-D_2+S_2 & D_2 & & & \\ & D_2 & -D_2-D_3+S_3 & D_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & D_{n-2} & -D_{n-2}-D_{n-1}+S_{n-1} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_{n-1} \end{bmatrix} = \begin{bmatrix} -Q_w+(B_w+S_1)P_{1_{t-\Delta t}} \\ S_2 P_{2_{t-\Delta t}} \\ S_3 P_{3_{t-\Delta t}} \\ \vdots \\ S_{n-1} P_{n-1_{t-\Delta t}} - D_{n-1} P_n \end{bmatrix} \quad (3.1\text{-}41)$$

This ensures that all terms in the vector on the right hand side of the equation are known, and all on the left are unknown. Matrices are re-partitioned at the start of each sequence, depending upon the wellbore boundary condition.

### 3.1.5.2 Dual-Porosity

Dual porosity adds another term to the constitutive equations for a fracture node *i*. Equation 3.1-1 is recast as:

$$d_{i-1} + s_i + d_{i,1} - d_i = 0 \quad (3.1\text{-}42)$$

where $d_{i,1}$ is the flow from the matrix into the fracture.

The matrix system associated with any one fracture node can be considered as a separate and independent one-dimensional system which can be represented as a set of matrix equations

analogous to equation 3.1-39. If the matrix properties are linear and constant, then, due to the geometry of the matrix, the equations representing each matrix system are identical, except for a scaling factor determined by the area of matrix in contact with the fracture system. Consequently the matrix equations can be written once for a unit area system, and then scaled by the contact area. For a matrix system with three nodes the following system of equations results:

$$\begin{bmatrix} -D_{l,1} & D_{l,1} & & \\ D_{l,1} & -D_{l,1}-D_{l,2}+S_{l,1} & D_{l,2} & \\ & -D_{l,2} & -D_{l,2}-D_{l,3}+S_{l,2} & D_{l,3} \\ & & D_{l,3} & -D_{l,3}+S_{l,3} \end{bmatrix} \begin{bmatrix} P_l \\ P_{l,1} \\ P_{l,2} \\ P_{l,3} \end{bmatrix} = \begin{bmatrix} d_{l,1} \\ S_{l,1}P_{l,1_{t-\Delta t}} \\ S_{l,2}P_{l,3_{t-\Delta t}} \\ S_{l,3}P_{l,3_{t-\Delta t}} \end{bmatrix} \qquad (3.1\text{-}43)$$

Gaussian elimination is used to transform the matrix to lower triangular form, allowing $d_{i,1}$ to be expressed as:

$$\begin{aligned} d_{i,1} = A_{m_i}(&X_0\ D_{i,1}\ P_i + X_1\ S_{i,1}\ P_{i,1_{t-\Delta t}} + \\ &+ X_2\ S_{i,2}\ P_{i,2_{t-\Delta t}} + \dots \\ &+ X_{m-1}\ S_{i,m-1}\ P_{i,m-1_{t-\Delta t}} + X_m\ S_{i,m}\ P_{i,m_{t-\Delta t}}) \\ &\text{or :} \\ d_{i,1} = &D_{m_i}\ P_i + C_{m_i} \\ &\text{where :} \end{aligned} \qquad (3.1\text{-}44)$$

$X_j$ = coefficients representing results of Gaussian elimination

Substituting 3.1-44 into 3.1-42 yields the following matrix equations for double porosity systems:

$$\begin{bmatrix} -D_1-S_1-B_w & D_1 & & & & \\ D_1 & -D_1-D_2-S_2-D_{m_2} & D_2 & & & \\ & D_2 & -D_2-D_3-S_3-D_{m_3} & D_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & D_{n-2} & -D_{n-2}-D_{n-1}-S_{n-1}-D_{m_{n-1}} & D_{n-1} \\ & & & & D_{n-1} & -D_{n-1}-D_{m_n}-S_n \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix} - \begin{bmatrix} -Q_w-(B_w-S_1)P_{1_{t-\Delta t}} \\ S_2P_{2_{t-\Delta t}} - C_{m_2} \\ S_3P_{3_{t-\Delta t}} - C_{m_3} \\ \vdots \\ S_{n-1}P_{n-1_{t-\Delta t}} - C_{m_{n-1}} \\ Q_o-S_nP_{n_{t-\Delta t}} - C_{m_n} \end{bmatrix} \qquad (3.1\text{-}45)$$

### 3.1.6 Matrix Solution

Equations 3.1-41 and 3.1-45 can be reduced to:

$$\begin{bmatrix} A_{1,1} & A_{1,2} & & & & & \\ A_{2,1} & A_{2,2} & A_{2,3} & & & & \\ & A_{3,2} & A_{3,3} & A_{3,4} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & A_{n-2,n-3} & A_{n-2,n-2} & A_{n-1,n} & \\ & & & & A_{n,n-1} & A_{n,n} & \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_{n-1} \\ C_n \end{bmatrix} \qquad (3.1\text{-}46)$$

where:

**A** = coefficient matrix
P = unknown pressure vector
C = right-hand side vector of constants

If all elements of ***A*** are constants, 3.1-46 is amenable to direct solution. This will be the case for liquid simulations with linear formation properties and linear boundary conditions. In this case nSIGHTS uses an algorithm which takes advantage of the symmetric tri-diagonal form of the matrices. For dual porosity simulations, matrix system pressures are calculated after solution of the fracture system by back substituting calculated pressures into the diagonalized matrix remaining after Gaussian elimination.

For non-linear systems of equations, which include all gas systems and any liquid system with one or more formation parameters or wellbore boundary condition parameters expressed as a function of pressure, a Newton-Raphson solution technique is used. This is an iterative approach which successively improves the estimate of the solution variable (**P**) with each iteration:

$$P_{new} = P_{old} + \delta P \qquad (3.1\text{-}47)$$

where :

$P_{new}$ = new estimate of solution vector

$P_{old}$ = previous iteration estimate of solution vector

$\delta P$ = change vector

The technique requires calculation of the partial derivatives of each nodal equation with respect to nodal pressure. The following system of equations is then solved for $\delta \boldsymbol{P}$:

$$\begin{bmatrix} \frac{\partial A_{1,1}}{\partial P_1} & \frac{\partial A_{1,2}}{\partial P_2} & & & & \\ \frac{\partial A_{2,1}}{\partial P_1} & \frac{\partial A_{2,2}}{\partial P_2} & \frac{\partial A_{2,3}}{\partial P_3} & & & \\ & \frac{\partial A_{3,2}}{\partial P_2} & \frac{\partial A_{3,3}}{\partial P_3} & \frac{\partial A_{3,4}}{\partial P_4} & & \\ & & & \ddots & & \\ & & & \frac{\partial A_{n-1,n-2}}{\partial P_{n-2}} & \frac{\partial A_{n-1,n-1}}{\partial P_{n-1}} & \frac{\partial A_{n-1,n}}{\partial P_n} \\ & & & & \frac{\partial A_{n,n-1}}{\partial P_{n-1}} & \frac{\partial A_{n,n}}{\partial P_n} \end{bmatrix} \begin{bmatrix} \ddot{a}P_1 \\ \ddot{a}P_2 \\ \ddot{a}P_3 \\ \vdots \\ \ddot{a}P_{n-1} \\ \ddot{a}P_n \end{bmatrix} = \begin{bmatrix} C_1 - A_{1,1}P_{old_1} - A_{1,2}P_{old_2} \\ C_2 - A_{2,1}P_{old_1} - A_{2,2}P_{old_2} - A_{2,3}P_{old_3} \\ C_3 - A_{3,2}P_{old_2} - A_{3,3}P_{old_3} - A_{3,4}P_{old_4} \\ \vdots \\ C_{n-1} - A_{n-1,n-2}P_{old_{n-2}} - A_{n-1,n-1}P_{old_{n-1}} - A_{n-1,n}P_{old_n} \\ C_n - A_{n,n-1}P_{old_{n-1}} - A_{n,n}P_{old_n} \end{bmatrix} \quad (3.1\text{-}48)$$

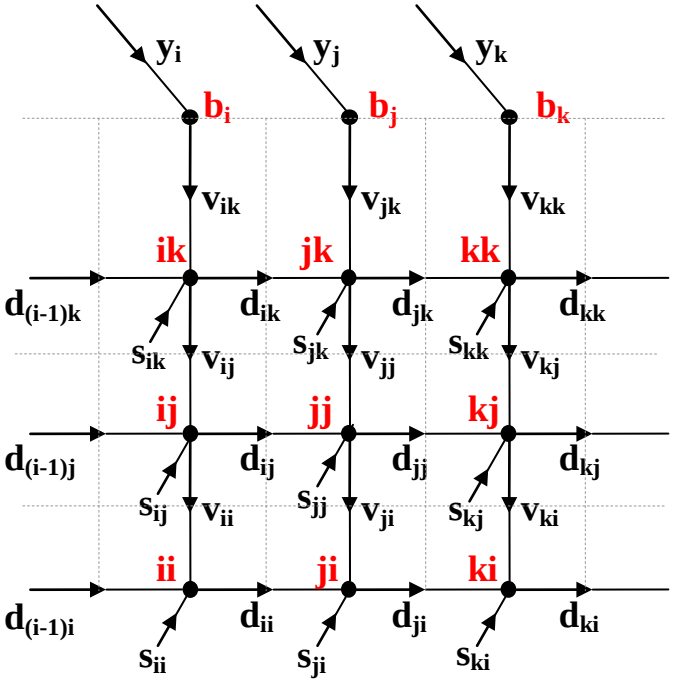
A two tolerance approach is used to control the number of iterations required before a solution is reached. The first tolerance is the sum of the nodal pressure change vector $\delta \boldsymbol{P}$ for the iteration. The second is the sum of the right hand side vector. Both of these tend towards zero as the solution improves. If either meets a user specified criteria, convergence is assumed:

$$TOL_P > \sum \delta P_i$$
$$\text{or :} \quad (3.1\text{-}49)$$
$$TOL_Q > \sum (C_i - A\ P_{old})$$

Partial derivatives are calculated using both analytic and numeric approaches. Analytic approaches are used when the matrix term is linear or is a simple polynomial. Numeric approaches are used for all other cases.

## 3.1.7 Unconfined and Partial Penetration

The unconfined and partial penetration cases are combined in this section, due to their interrelated embodiments. The equations for unconfined aquifers implemented in nSIGHTS assumes a two-dimensional, liquid, non-leaky, single porosity system with a flow dimension of 2. Other flow dimensions are possible, but are theoretically suspect.




where:
$d_{ii}$ represents the flow between nodes ii and ji,
$v_{ij}$ represents the flow between nodes ij and ii
$si_i$ represents the flow due to storage within the volume represented by node ii
$y_i$ represents the flow caused by the changing water table level within the i node column
$b_i$ represents the water table elevation at node column i (this node is in addition to the number of vertical nodes)

**Figure 3.8 System graph for unconfined system.**

Constitutive relationships hold for each node, with an additional relationship required for the water table node. For example, at any given time t and n vertical nodes, equation (3.1-50) represents the constitutive relationship at node jj and equation (3.1-51) represents the relationship at the water table associated with node column j:

$$d_{ij} - d_{jj} + v_{jk} - v_{jj} + s_{jj} = 0 \qquad (3.1\text{-}50)$$

$$-v_{jn} + y_j = 0 \qquad (3.1\text{-}51)$$

When expanded with the edge equations, the constitutive relationships for each node are used to create a set of algebraic equations describing the system. Solving the equations yields nodal pressure and water table elevations at each node.

In the development of the edge equations, the thickness of the formation is a variable, and is consequently kept separate from the area term. Assuming a flow dimension of 2, the area term used in the unconfined edge equations is equal to the area term described in equation 3.1-9

divided by the thickness of the formation:

$$A_i = 2\pi r_{a_i} \quad (3.1\text{-}52)$$

Flow dimensions other than 2 can be used, but are theoretically suspect.

As the volume term contains the area term, it is likewise divided by the formation thickness:

$$V_i = \pi r_{a_i}^2 - \pi r_{a_{i-1}}^2 \quad (3.1\text{-}53)$$

The vertical node spacing is equal along the height of the node column, with the first node located at the bottom of the column, and a b node at the top of the column. The b node is in addition to the number of vertical nodes. As the height of the node column changes with time (as the water table lowers or raises), the node spacing is re-calculated at each time step.

The area and volume terms are the same for each node in a column, with the exception of the bottom and top node. The bottom node has only half the volume of an interior vertical node, and the top node has 1.5 times the volume of an interior node.

Edge equations describe the volumetric flow ($L^3t^{-1}$) for each type of graph edge. Four edge equations are developed for the unconfined system: horizontal flow, vertical flow, storage, and water table yield. For convenience, the edge equations are defined here in terms of head, instead of pressure. The actual implementation is in terms of pressure.

Equation 3.1-54 presents the governing equation for horizontal flow:

$$d_{ij} = K_H A_i b_{ij} \frac{\Delta h}{\Delta r} \quad (3.1\text{-}54)$$

where:

$K_H$ = horizontal hydraulic conductivity, $Lt^{-1}$
$b_{ij}$ = block height, L
$\Delta h$ = change in head between nodes ij and (i+1)j, L
$\Delta r$ = distance between nodes ij and (i+1)j, L

Equation 3.1-55 presents the governing equation for vertical flow:

$$v_{ij} = K_v V_i \frac{\Delta h}{b_{ij}} \quad (3.1\text{-}55)$$

where:

$K_V$ = vertical hydraulic conductivity, $Lt^{-1}$
$\Delta h$ = change in head between nodes ij and i(j+1), L

Note that for the top vertical node, $\Delta h = h - b$.

Equation 3.1-56 presents the equation for storage:

$$s_{ij} = SV_i b_{ij} \frac{\Delta h}{\Delta t} \quad (3.1\text{-}56)$$

where:

S = formation specific storage, $L^{-1}$

$\Delta h$ = change in head at node ij between time t and time t-$\Delta t$, L

$\Delta t$ = time step size, t

Equation 3.1-57 presents the storage implementation for water table yield:

$$y_i = S_y V_i \frac{\Delta b}{\Delta t} \quad (3.1\text{-}57)$$

where:

$S_y$ = specific yield

$\Delta b$ = change in water table height between time t and time t-$\Delta t$, L

### 3.1.7.1 Matrix Equations for Unconfined Case

The edge equations can be reduced to the following form:

$$d_{ij} = D_{ij}(h_{(i+1)j} - h_{ij}) \quad (3.1\text{-}58)$$
$$v_{ij} = W_{ij}(h_{ij} - h_{i(j+1)})$$
$$s_{ij} = S_{ij}(h_{ij} - h_{ij(t-\Delta t)})$$
$$y_i = Y_i(b_i - b_{i(t-\Delta t)})$$

The $b_i$ terms within $D_{ij}$, $W_{ij}$ and $S_{ij}$ are taken from the previous time step (i.e. $b_{i(t-\square t)}$), in order to produce a linear set of equations.

Applying constitutive relationships at node i (assuming n vertical nodes) yields:

$$D_{(i-1)j}(h_{ij} - h_{(i-1)j}) - D_{ij}(h_{(i+1)j} - h_{ij}) + W_{ij}(h_{ij} - h_{i(j+1)}) - W_{i(j-1)}(h_{i(j-1)} - h_{ij}) + S_{ij}(h_{ij} - h_{ij(t-\Delta t)}) = 0 \quad (3.1\text{-}59)$$

and

$$-W_{in}(h_{in} - b_i) + Y_i(b_i - b_{i(t-\Delta t)}) = 0 \quad (3.1\text{-}60)$$

Rearranging equations (3.1-59) and (3.1-60) yields:

$$-D_{(i-1)j}h_{(i-1)j} - W_{ij}h_{i(j+1)} + (D_{(i-1)j} + D_{ij} + W_{ij} + W_{i(j-1)} + S_{ij})h_{ij} - W_{i(j-1)}h_{i(j-1)} - D_{ji}h_{(i+1)j} = S_{ij}h_{ij(t-\Delta t)} \quad (3.1\text{-}61)$$

and

$$-W_{in}h_{in} + (W_{in} + Y_i)b_i = Y_i b_{i(t-\Delta t)} \quad (3.1\text{-}62)$$

Using the boundary conditions described in the nSIGHTS technical documentation and applying

equations (3.1-61) and (3.1-62) to the entire system graph (assuming 3 vertical nodes and m radial nodes) leads to the following matrix equation:

$$\begin{bmatrix}
A_{11} & -W_{11} & & -D_{11} & & & & & & & & \\
-W_{11} & A_{12} & -W_{12} & & -D_{12} & & & & & & & \\
 & -W_{12} & A_{13} & -W_{13} & & -D_{13} & & & & & & \\
 & & -W_{13} & B_1 & & & & & & & & \\
-D_{11} & & & & A_{21} & -W_{21} & & -D_{21} & & & & \\
 & -D_{12} & & & -W_{21} & A_{22} & -W_{22} & & -D_{22} & & & \\
 & & -D_{13} & & & -W_{22} & A_{23} & -W_{23} & & -D_{23} & & \\
 & & & & & & -W_{23} & B_2 & & & & \\
 & & & & \ddots & & & \ddots & \ddots & \ddots & & \\
 & & & & & -D_{m1} & & & -W_{m1} & A_{m2} & -W_{m2} & \\
 & & & & & & -D_{m2} & & & -W_{m2} & A_{m3} & -W_{m3} \\
 & & & & & & & & & & -W_{m3} & B_m
\end{bmatrix} \cdot \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ b_1 \\ h_{21} \\ h_{22} \\ h_{23} \\ b_2 \\ \vdots \\ h_{m2} \\ h_{m3} \\ b_m \end{bmatrix} = \begin{bmatrix} -Q_w + (B_w + S_{11})h_{11(t-\Delta t)} \\ S_{12}h_{12(t-\Delta t)} \\ S_{13}h_{13(t-\Delta t)} \\ Y_1 b_{1(t-\Delta t)} \\ S_{21}h_{21(t-\Delta t)} \\ S_{22}h_{22(t-\Delta t)} \\ S_{23}h_{23(t-\Delta t)} \\ Y_2 b_{2(t-\Delta t)} \\ \vdots \\ Q_o + S_{m2}h_{m2(t-\Delta t)} \\ Q_o + S_{m3}h_{m3(t-\Delta t)} \\ Y_m b_{m(t-\Delta t)} \end{bmatrix} \quad (3.1\text{-}63)$$

where:

$$A_{11} = S_{11} + W_{11} + D_{11} + B_w$$

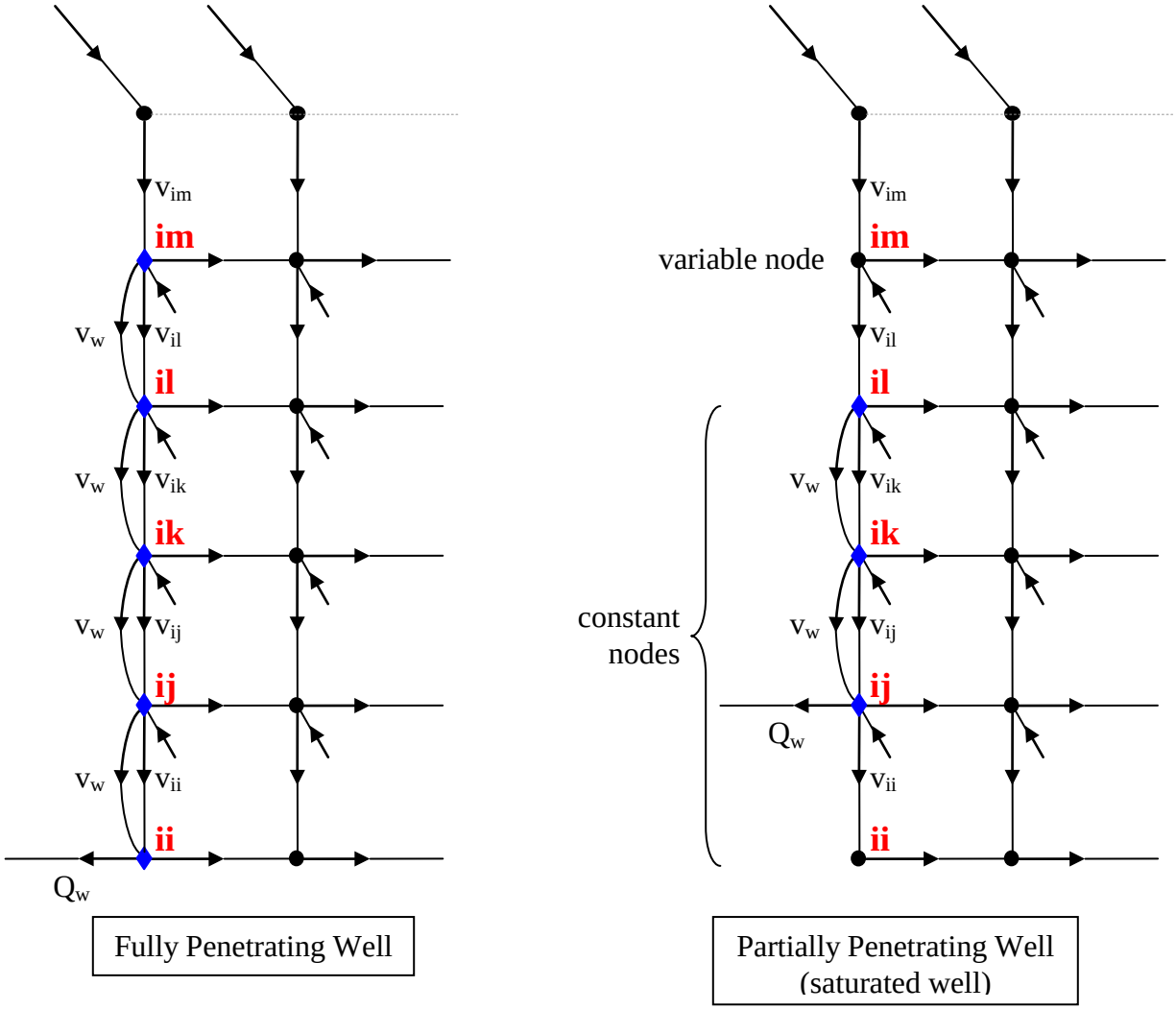
$$A_{ij} = D_{(i-1)j} + D_{ij} + W_{i(j-1)} + W_{ij} + S_{ij}$$

$$B_i = W_{i3} + Y_i$$

The existing linear matrix solver in nSIGHTS assumes a tri-diagonal form. In order to solve equation (3.1-63), which has two additional diagonal bands, a banded matrix solver was implemented.

### 3.1.7.2 Well Boundary Condition for Fully Penetrating and Partially Penetrating Wells

Both fully penetrating and partially penetrating wells are implemented for the unconfined solution in nSIGHTS. In both cases, flow is extracted from the bottom node of the well, and an additional high-conductivity flow edge is added between well nodes. The figure below shows a segment of the directed graph near the well, for a case with five vertical nodes. The location of the extraction node, and the additional vertical flow edges for both the fully penetrating and partially penetrating well cases are shown. Only edges relevant to the discussion are labeled.




where:

blue diamonds represent well nodes

$v_w$ represents the high-conductivity vertical well flow between well nodes

$Q_w$ represents the well flow rate

**Figure 3.9 System graph for unconfined system; Fully and Partially Penetrating Wells.**

The high conductivity vertical flow term is defined as follows:

$$v_w = K_w A_w \frac{\Delta h}{b_{ij}} \quad (3.1\text{-}64)$$

where:

$$K_w = \frac{r_w^2 \rho g}{8\mu} \quad (3.1\text{-}65)$$

$$A_w = \pi r_w^2 \quad (3.1\text{-}66)$$

$r_w^2$ is the well radius

This high conductivity term is added the vertical flow edge equation, such that for well nodes, $W_{ij}$ is redefined as:

$$W_{ij} = (K_v V_i + K_w A_w)\frac{1}{b_{ij}} \quad (3.1\text{-}67)$$

The matrix equations are then solved as described above.

In the full penetration well case, the node spacing at the well changes with time. As the location of the well nodes cannot change with time in the partial penetration case without causing numerical instabilities, two sets of vertical nodes are defined in the partial penetration case: a set of variable vertical nodes and a set of constant vertical nodes. In the case where the partially penetrating well screen extends to surface (a water table well), the variable vertical nodes are defined as well nodes, and the constant vertical nodes are defined as formation nodes below the well. If the partially penetrating well does not extend to surface (saturated well), the variable vertical nodes are defined as formation nodes, and the constant vertical nodes are split into well nodes and formation non-well nodes. The partially penetrating well shown in the above figure is a saturated well. The case of the saturated well does not allow the water table to drop below the top elevation of the well screen.

The well nodes of a partially penetrating well are defined as all nodes between two elevation values, defined by the offset from the bottom of the formation and the well screen length. For a water table well, the well screen length is assumed to be the saturated formation thickness minus the bottom offset.

## *3.2 Functional Approximations*

nSIGHTS uses functional approximations to represent:

1) well-bore boundary conditions varying as a function of time (e.g., variable pumping rates),

2) radial heterogeneity due to formation properties varying as a function of distance from the well-bore, and

3) pressure-dependent non-linear parameters where parameter values vary as a function of pressure.

nSIGHTS uses several methods for expressing these functional approximations. Most methods require that XY data representing the desired function be available. X values are also usually required to be in ascending order. Available methods are:

1) polynomial of order 1 to 10 - a polynomial of specified order is calculated using a linear regression on the entered XY data. The polynomial will not represent the input data exactly unless the order of the polynomial is one less than the number of points in the input data.

2) cubic spline - a piece-wise polynomial approximation of the XY data is calculated, continuous in first and second derivatives. Function slopes at the extremes must be specified. A tension factor can be used to modify the shape of the function. The cubic spline will represent each point in the input data exactly.

3) linear - the functional representation is a series of straight lines joining consecutive XY data points. Derivatives are not continuos.

4) step - function values are constant at XY data point values. Values change at the linear mid-point between adjacent data points.

Figure 3.10 illustrates the functional approximations for a single XY data set.

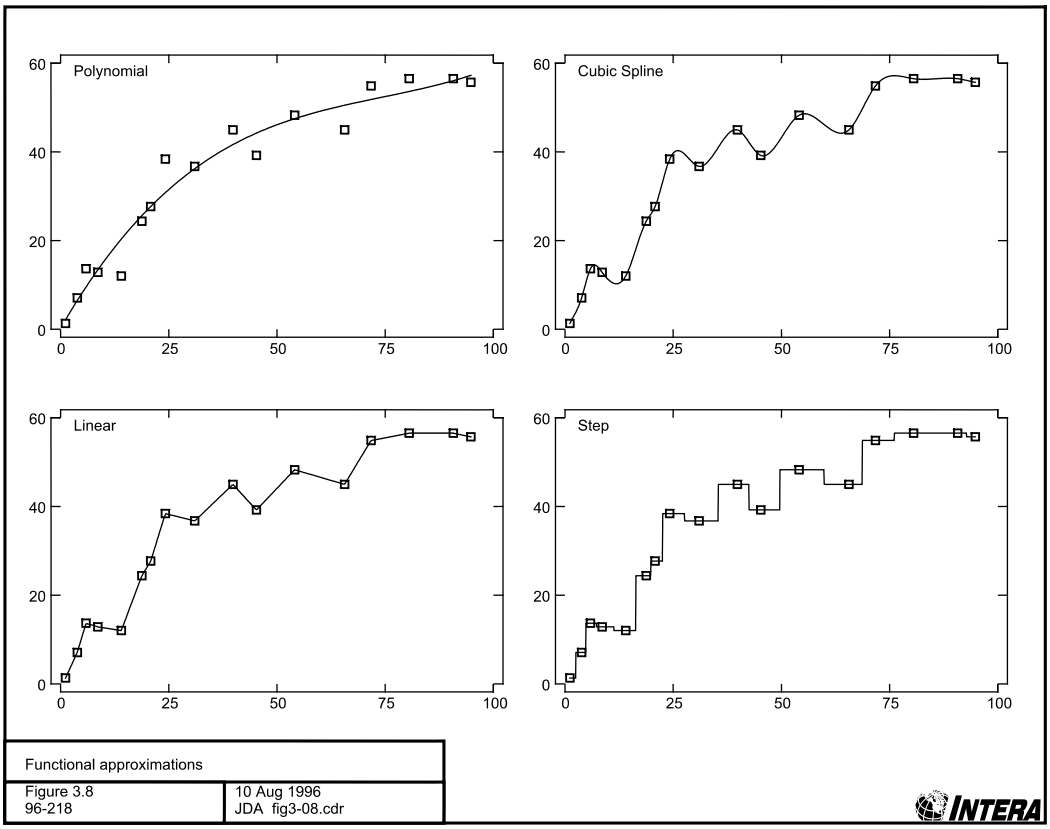



**Figure 3.10 Functional approximations**

## *3.3 Data Analysis*

Apart from its simulation capabilities, nSIGHTS contains facilities for performing various transformations and processing on user-supplied data and on simulation results. User-supplied data are usually well-test results acquired in the field. These data are typically available in the form of a pressure and/or flow-rate versus time record, describing the response of the formation to the testing procedures over the duration of the test. Ancillary data such as test-zone temperature versus time may also be available. nSIGHTS produces simulation results in a similar form. The predicted pressure and/or flow rates in the test-zone (or pressure at specified radii from the test-zone) are produced for each time step in the simulation. The time record starts at the specified test-start time.

In many cases, the available form of the user-supplied and/or the nSIGHTS predicted data is not appropriate for further analysis. nSIGHTS provides the tools required to further process both user and simulation data. Additionally, nSIGHTS provides a number of standard analytic tools used in well-test interpretation. These tools can be categorized as scaling/transformations, derivative calculations, and time/superposition functions.

### 3.3.1 Scaling /Transformations

User-supplied and nSIGHTS predicted data contain X and Y components. Scaling/transformation procedures are applied to each component independently. Available scaling procedures include: multiplying or dividing data by constants, and/or adding or subtracting constants from the data. Scaling is useful for normalizing, converting units, and offsetting data. In certain cases, scaling constants can be replaced by variables associated with each simulation, such as static formation pressure.

Transformations can also be applied, either before or after scaling is applied. There are 11 available transformations: natural log, log base 10, square root, inverse, exponent, raise 10 to the power, square, absolute value, square root of the square root, inverse square root, and log base 10 of the absolute value.

### 3.3.2 Derivative Calculations

Pressure derivatives are frequently used in well-test interpretation to determine reservoir properties. nSIGHTS can calculate the derivative of any XY data set with respect to several time functions. Available time functions include: normal, superposition, Horner, and Argawal time. These are described in further detail in the next sub-section. When computing the derivative, the nSIGHTS derivative calculation procedures may also perform smoothing on the input data set. Smoothing is necessary to produce a useful derivative if the input data is noisy.

All derivative procedures select a subset of the available data on either side of a data point and use the subset as the basis for calculating the derivative value at the data point. There are six methods for selecting data subsets and calculating derivatives:

| | |
|---|---|
| single point | derivative is calculated as slope of line joining adjacent data points. The X value for the derivative is set as the linear average of the two data values used. |
| 2 slope average | the derivative at a point is calculated as the average of the slopes of the lines joining the point to adjacent points. |
| Window | a specified number of data values on either side of the point are used in the calculation. |
| % lin span | all points that have X values within a specified distance of the point at which the derivative is being calculated are used in the calculation. The distance is expressed as a percentage of the linear range of the entire data set ($X_{max}$ - $X_{min}$). |
| % log span | all points for which the log of the X value are within a specified distance of the log of the X value of the point at which the derivative is being calculated are used in the calculation. The distance is expressed as a percentage of the log range of the entire data set ($\log(X_{max})$ - $\log(X_{min})$). |
| full polynomial | The entire data set is used. A best-fit polynomial of order n is determined using a linear regression algorithm. Coefficients of the polynomial describing the derivative of the best-fit polynomial are determined and the derivative calculated for each X value in the input data set. |

The “single point” and “two-slope average” methods are useful only for very smooth field data sets or for synthetic data generated by nSIGHTS or by another method. The three windowed methods provide sufficient smoothing to render usable derivatives. The degree of smoothing is increased by increasing the number of points in the window, either directly for the “window” method, or by increasing the percentage of the data span. The full polynomial method, which yields very smooth derivatives, is unfortunately almost useless for normal test-interpretation as typical well-test responses are not well described by polynomials.

Further options for derivative calculation are available for the three windowed methods (described above as window, % lin span and % log span). These options determine how the derivative is calculated using the subset of the data which falls within the window:

1) regression - a best-fit straight-line for the windowed subset is determined using a linear regression algorithm. The slope of the line is the derivative at the selected point.

2) Clark - the derivative is calculated using an algorithm developed by Clark and von Golf-Racht, 1985:

$$\frac{dy_i}{dx_i} = \frac{\frac{y_i - y_s}{x_i - x_s}(x_e - x_i) + \frac{y_e - y_i}{x_e - x_i}(x_i - x_s)}{x_e - x_s} \qquad (3.3\text{-}1)$$

where :

$\frac{dy_i}{dx_i}$ = derivative at point i

$x_s$, $y_s$ = X and Y value of point at start of window

$x_e$, $y_e$ = X and Y value of point at end of window

3) Simmons - the algorithm developed by Simmons is used to calculate the derivative at the data point:

for i = 1

$$\frac{dy_1}{dx_1} = \frac{\left[\left(1 - \frac{(x_e - x_1)^2}{(x_m - x_1)^2}\right) y_1 + \frac{(x_e - x_1)^2}{(x_m - x_1)^2} y_m - y_e\right]}{\frac{(x_e - x_1)^2}{x_m - x_s} - (x_e - x_1)}$$

for i = n

$$\frac{dy_n}{dx_n} = \frac{\left[\left(1 - \frac{(x_n - x_s)^2}{(x_n - x_m)^2}\right) y_n + \frac{(x_n - x_s)^2}{(x_n - x_m)^2} y_m - y_s\right]}{(x_n - x_s) - \frac{(x_n - x_s)^2}{x_n - x_m}}$$

otherwise :

$$\frac{dy_i}{dx_i} = \frac{(x_i - x_s)^2 y_e + ((x_e - x_i)^2 - (x_i - x_s)^2) y_i - (x_e - x_i)^2 y_s}{(x_i - x_s)^2 (x_e - x_i) + (x_e - x_i)^2 (x_i - x_s)}$$

where : (3.3-2)

$x_m$ = mid-point x value

$y_m$ = mid-point y value

### 3.3.3 Time/Superposition Functions

Time functions are used to adjust for non-ideal initial conditions such as multi-rate flow periods and to perform specialized interpretations such as Horner plots. nSIGHTS supports four time functions in addition to normal time.

Horner time (Horner, 1951) is described as:

$$t_h = \frac{t_p + \Delta t}{\Delta t} \qquad (3.3\text{-}3)$$

where :

$t_h$ = Horner time

$t_p$ = length of flow period before shut-in

$\Delta t$ = time since shut-in

Argawal equivalent time (Argawal, 1980) is:

$$t_e = \frac{t_p \Delta t}{t_p + \Delta t} \qquad (3.3\text{-}4)$$

where :

$t_e$ = Argawal equivalent time

Multiple rate Horner superposition time:

$$t_h = \frac{1}{q_n} \sum_{i=1}^{n} q_i \log\left(\frac{t_n - t_{i-1} + \Delta t}{t_n - t_i + \Delta t}\right) \qquad (3.3\text{-}5)$$

where :

n = number of flow periods

$t_i$ = end-time of flow period i

$q_i$ = flow rate during period i

$t_0 = 0.0$

Multiple rate Bourdet superposition time:

$$t_s = \ln(\Delta t) + \frac{1}{q_n - q_{n-1}} \sum_{i=1}^{n-1} (q_i - q_{i-1}) \ln\left( \Delta t + \sum_{j=i}^{n-1} (t_{j+1} - t_j) \right)$$

(3.3-6)

where :

$t_s$ = superposition time

$q_0 = 0.0$

## *3.4 Optimization*

One of nSIGHTS's most useful capabilities is the ability to perform optimizations or inverse modelling. These procedures search for the combination of parameter values which minimize the difference between two data sets: a field data set, and simulation results. Two optimization algorithms are available within nSIGHTS: the downhill Simplex method and Levenberg-Marquardt. Both are described in some detail in Press et al (1992): Simplex in Section 10.4; Levenberg-Marquardt in Section 15.5. Both algorithms have the same goal: to minimize a function. The following sub-sections describe: the functions to be minimized; parameter normalizations used in optimizations; the optimization procedures; the calculation of fit statistics; the calculation of covariance matrices and Jacobian data; and the determination of single- and joint-parameter confidence intervals.

### 3.4.1 Minimization Functions

In nSIGHTS, the function to be minimized is the sum of one or more fit component functions:

$$f(x\,;\,a) = \sum_{i=1}^{n_f} f_i(x\,;\,a)$$

where:

(3.4-1)

$a$ = vector of parameters to be optimized

$f(x\,;\,a)$ = function to be minimized

$f_i(x\,;\,a)$ = fit component function

$n_f$ = number of fit components

each of which represents some function comparing field or processed data and analogous simulation results. For example, one fit component may compare flow rates, another pressures in

an observation well, while a third may compare pressure derivatives. Typically the $X^2$ (Chi-squared) function is used:

$$f_i(x;\ a) = \frac{1}{\sigma_i^2} \sum_{j=1}^{N_i} \left[ y_j - y(x_j;\ a) \right]^2$$

where :

(3.4-2)

$N_i$ = number of points to be fitted in data set i

$y_j$ = data to be fitted

$y_j(x_j; a)$ = nSIGHTS calculated result at time $x_j$

$\sigma_i$ = measurement error of data set

The data to be fitted, $y_i$, may be actual field data, processed field data (e.g. pressure derivatives), or an interpolated data set generated from actual/processed field data using one of the functional approximations described in Section 3.2. Interpolations can be used advantageously when the original field data are poorly spaced in time, however, the statistical validity of calculated confidence intervals may be questionable.

Equation 3.4-2 must be used when confidence limits are to be calculated or when the Levenberg-Marquardt algorithm is used. If more than one fit component is used, $\sigma_i$ may be replaced with the normalized ratio of the means of the fit component data:

$$\sigma_i = \frac{\text{mean}_i}{\text{mean}_{min}}$$

where : (3.4-3)

$\text{mean}_i$ = mean of data set i

$\text{mean}_{min}$ = minimum mean of all data sets to be fitted

This approach should only be used when the actual measurement error is not known. Confidence intervals calculated using equation 3-4.3 will not be correct and should be used for qualitative comparisons only.

### 3.4.2 Parameter Normalization

The parameter vector, **a**, used in the optimization consists of parameters selected by the user for optimization and normalized over a range 0 to 1. Both linear and logarithmic normalizations are available:

$$P_i = P_{i_{min}} + a_i (P_{i_{max}} - P_{i_{min}})$$

or :

$$P_i = 10^{\log_{10}(P_{i_{min}}) + a_i (\log_{10}(P_{i_{max}}) - \log_{10}(P_{i_{min}}))}$$

where : (3.4-5)

$a_i$ = parameter value used within optimizer

$P_i$ = parameter value used within simulator

$P_{i_{min}}$ = user specified minimum value for optimized parameter

$P_{i_{max}}$ = user specified maximum value for optimized parameter

### 3.4.3 Optimization Procedures

Two optimization procedures are described below.

#### 3.4.3.1 Simplex

This sub-section presents a brief overview of points relevant to using the Simplex procedure in nSIGHTS. The Simplex procedure is described in detail in Section 10.4 of Press et al (1992). The algorithm requires $n_p + 1$ initial evaluations of the fit function to form the vertices of the "simplex", a non-degenerate geometric structure in $n_p$ dimensional parameter space:

$$S_o = f(x ; a_0)$$

and :

$$S_i = f(x ; a_i)$$

where : (3.4-6)

$S_i$ = simplex vertices

$a_0$ = initial user-entered estimate of parameter values

$a_i = a_0 + \lambda e_i$

$\lambda$ = vertex span

$e_i$ = unit vector with element i = 1, all others = 0

The Simplex algorithm then examines the vertex values and successively adjusts parameter estimates ***a*** so that the values are reduced. The optimization is complete when the following criteria is reached:

$$TOL > 2\frac{|S_{max} - S_{min}|}{|S_{max}| + |S_{min}|}$$

where :

(3.4-7)

$S_{max} = MAX(S_i), i = 0..n_p$

$S_{min} = MIN(S_i), i = 0..n_p$

TOL = user specified tolerance

### 3.4.3.2 Levenberg-Marquardt

The Levenberg-Marquardt (L-M) algorithm is described in Section 15.5 of Press et al (1992). The L-M method is a gradient method which requires evaluation of the Hessian matrix (partial derivative of the function with respect to its parameters) at each successful iteration. New estimates of the parameters are determined based on the gradient and a step parameter, $\lambda$. The smaller the value of $\lambda$, the greater the step change. If the new estimate improves the fit, the fit-point is updated and the gradient calculated again. Otherwise, $\lambda$ is increased and a new estimate point recalculated. If a previous iteration improved the fit without increasing $\lambda$, the $\lambda$ value is reduced by an adjustment factor to accelerate the convergence to a fit point. The fit-point is reached when either of the following conditions are met:

$$P_{TOL} > 2\frac{|f(x;\ a_{min}) - f(x;\ a_{last})|}{f(x;\ a_{min}) - f(x;\ a_{last})}$$

or :

$$R_{TOL} > 1 - \frac{f(x;\ a_{last-5})}{f(x;\ a_{last})}$$

where :

(3.4-8)

$a_{min}$ = parameters for previous minimum $X^2$ value

$a_{last}$ = parameters for current iteration

$P_{TOL}$ = user specified parameter tolerance

$R_{TOL}$ = user specified relative change tolerance

### 3.4.4 Fit Statistics

After a Chi-squared optimization has reached a fit estimate the following fit statistics are calculated:

$$SSE_i = \sum_{j=1}^{N_i} [y_j - y(x_j; a_f)]^2$$

$$SSE = \sum_{i=1}^{n} SSE_i$$

$$N = \sum_{i=1}^{n} N_i$$

$$MSE = \frac{SSE}{N}$$

$$\text{est } \sigma_i = \frac{SSE_i}{N_i - n_p}$$

$$\text{est } \sigma = \frac{SSE}{N - n\ n_p}$$

where :

$SSE_i$ = sum of squared errors for fit component i

SSE = sum of squared errors for full fit

N = number of points in full fit

MSE = mean squared errors for full fit

est $\sigma_i$ = estimated variance for fit component i

est $\sigma$ = estimated variance for full fit

(3.4-9)

### 3.4.5 Covariance Matrices

nSIGHTS calculates several covariance matrices at the optimization solution point. $\boldsymbol{C_a}$, the "actual covariance matrix", is the estimated covariance matrix of the standard errors in the fitted normalized parameters **a**. It is calculated as:

$$C_a = \left[ \frac{1}{2} \frac{\partial^2 X^2}{\partial a_k \partial a_l} \right]^{-1}$$

where :

(3.4-10)

$C_a$ = covariance matrix

$\frac{\partial^2 X^2}{\partial a_k \partial a_l}$ = Hessian matrix

Two formulations are used in calculating the Hessian:

Half (1st order):

$$\frac{\partial^2 X^2}{\partial a_k \partial a_l} = 2 \sum_{i=1}^{n_f} \frac{1}{\sigma_i^2} \sum_{j=1}^{N_i} \left[ \frac{\partial y(x_j;a)}{\partial a_k} \frac{\partial y(x_j;a)}{\partial a_l} \right] \qquad (3.4\text{-}11)$$

or, Full (2nd order):

$$\frac{\partial^2 X^2}{\partial a_k \partial a_l} = 2 \sum_{i=1}^{n_f} \frac{1}{\sigma_i^2} \sum_{j=1}^{N_i} \left[ \frac{\partial y(x_j;a)}{\partial a_k} \frac{\partial y(x_j;a)}{\partial a_l} - [y_j - y(x_j;a)] \frac{\partial^2 y(x_j;a)}{\partial a_l \partial a_k} \right]$$

The first (Half) calculation ignores the second derivative terms and has the advantage of requiring less simulations. It is also guaranteed to result in a positive definite Hessian matrix. The second (Full) equation is more strictly correct, but may be singular at the fit point if the fit is poor. Partial derivatives are calculated numerically:

Half:

$$\frac{\partial y(x_j;a)}{\partial a_k} = \frac{y(x_j; a_{+\Delta_k}) - y(x_j; a_f)}{\Delta a}$$

Full:

$$\frac{\partial y(x_j;a)}{\partial a_k} = \frac{y(x_j; a_{+\Delta_k}) - y(x_j; a_{-\Delta_k})}{2\,\Delta a}$$

$$\frac{\partial^2 y(x_j;a)}{\partial a_k \partial a_k} = \frac{y(x_j; a_{+\Delta_k}) - 2\ y(x_j; a_f) - y(x_j; a_{-\Delta_k})}{\Delta a^2}$$

$$\frac{\partial^2 y(x_j;a)}{\partial a_k \partial a_l} = \frac{y(x_j; a_{+\Delta_{k,l}}) + y(x_j; a_f) - y(x_j; a_{+\Delta_k}) - y(x_j; a_{+\Delta_l})}{\Delta a^2}$$

where:

$a_f$ = parameter values at fit point

$\Delta a$ = derivative span

$a_{+\Delta_k} = a_f$, except element $a_{\Delta_k} = a_{f_k} + \Delta a$

$a_{-\Delta_k} = a_f$, except element $a_{\Delta_k} = a_{f_k} - \Delta a$

$a_{+\Delta_{k,l}} = a_f$, except element $a_{\Delta_k} = a_{f_k} + \Delta a$ and element $a_{\Delta_l} = a_{f_l} + \Delta a$ (3.4-12)

The derivative span, *Δa*, is calculated using an iterative procedure:

$$Z_{new} = \frac{f(a + \Delta a) - f(a - \Delta a)}{1.5\Delta a} + \frac{f(a - 2\Delta a) - f(a + 2\Delta a)}{12\Delta a}$$

REPEAT

$$Z_{old} = Z_{new}$$

$$\Delta a = \frac{\Delta a}{2}$$

$$Z_{new} = \frac{f(x + \Delta a) - f(x - \Delta a)}{1.5\Delta a} + \frac{f(x - 2\Delta a) - f(x + 2\Delta a)}{12\Delta a}$$

$$\text{criteria} = \text{MAX}(\text{TOL}, \text{TOL}\, Z_{new})$$

$$\text{UNTIL } \frac{|Z_{new} - Z_{old}|}{15} > \text{criteria}$$

where :

$f(a + \Delta a) = f(x ; a_{+\Delta k})$, where k is parameter with largest $\Delta f$

TOL = user specified tolerance (3.4-13)

$\boldsymbol{C_e}$, the "estimated covariance" is also calculated using equations 3.4-10 and 3.4-11 except that $\sigma i$ in 3.4-11 is replaced with *est* $\sigma_i$. Component estimated covariances $\boldsymbol{C_{ei}}$ are also calculated for each separate fit component.

The covariance matrices $\boldsymbol{C_a}$, $\boldsymbol{C_e}$, $\boldsymbol{C_{ei}}$ are all with respect to the normalized parameters $\boldsymbol{a}$. De-normalization procedures can be applied so that covariance matrices reflect the ranges of the parameter values used in the simulator:

$$C_D = T\, C\, T$$

where :

$C_D$ = denormalized covariance (3.4-14)

$T = T_{i,i} = P_{i_{max}} - P_{i_{min}}$ linear normalization, or,

$T_{i,i} = \log(P_{i_{max}}) - \log(P_{i_{min}})$ logarithmic

$T_{i,j} = 0, \quad i \neq j$

Jacobian data are also calculated:

$$J_{k,j} = \frac{\sigma_{p_k}}{\Delta a \, \sigma_j^2} \left( y(x_j; a_\Delta) - y(x_j; a_f) \right)$$

where : (3.4-15)

$J_{k,j}$ = Jacobian value for parameter k, point j

$\sigma_{p_k}$ = std. deviation of parameter k

### 3.4.6 Confidence Limits

Single- and joint-parameter confidence limits can be calculated for any covariance matrix ***C***. The single parameter confidence limits are simply:

$$\delta a_k = \pm \sqrt{\Delta X_1^2}\sqrt{C_{kk}}$$

where :

(3.4-16)

$\Delta X_1^2$ = Chi-squared value for 1 degree of freedom corresponding to confidence limit

The joint confidence area for two parameters takes the form of an ellipse in parameter space centered on the fit point:

$$\frac{\delta a_1}{b_1^2} + \frac{\delta a_2}{b_2^2} = 1$$

where :

(3.4-17)

$b_i$ = lengths of ellipse semi-major axes

The lengths of the semi-major axes are calculated using the eigenvalues of the inverse of the projected covariance matrix:

$$b_i = \sqrt{\frac{\Delta X_2^2}{D_i}}$$

where :

(3.4-18)

$D_i$ = eigenvalues of $C_{proj}^{-1}$

The rotation of the ellipse is defined by the eigenvectors of $\boldsymbol{C}_{proj}^{-1}$. Eigenvectors and eigenvalues are calculated using a Jacobi transformation routine presented in Press et. al. The values of the Chi-squared function corresponding to different confidence intervals and degrees of freedom are presented in the table below:

**Table 3.4-1 Chi-squared values**

| confidence interval % | degree of freedom 1 | 2 | 3 |
|---|---|---|---|
| 68.3 | 1.00 | 2.41a | 3.53 |
| 90 | 2.71 | 4.61 | 6.25 |
| 95.4 | 4.00 | 6.17 | 8.02 |
| 99 | 6.63 | 9.21 | 11.3 |

Equations 3.4-13 and 3.4-14 extend naturally to three dimensions to provide confidence ellipsoids for three parameters.

## *3.5 Sampling*

In sampling mode, nSIGHTS combines optimizations of fitting-parameters with sampled values of non-fitting parameters and/or sequence boundary conditions to create a statistical description of the range of possible fitting parameter values.

nSIGHTS uses sampling routines translated from the Fortran routines described in Iman and Shortencarier (1984). Two sampling modes are supported: Monte-Carlo and Latin-Hypercube (LHS). Up to 10,000 samples of a maximum of fifty variables can be generated. nSIGHTS uses a seed-based random number generator. The seed is user input so that sample realizations are reproducible.

The LHS sampler allows correlations between variables to be specified or generated randomly. Figure 3.10 shows a plot of two strongly correlated (0.90) uniform distributions.

nSIGHTS allows selected variables to be described by any one of six distributions: normal, log-normal, uniform, log-uniform, triangular, or log-triangular. Figure 3.11 shows histograms for example normal, triangular, and uniform distributions.

In post-processing, nSIGHTS calculates various summary statistics for both sampled and optimized variables. Statistics include: minimum, maximum, mean, variance and standard deviation. Note that if the sampled variable or optimized variable is logarithmic (loguniform, lognormal, or logtriangular distributions for sampled variables, log optimization stepping for optimized variables) the mean, variance, and standard deviation are calculated using the $\log_{10}$ of the data. The mean is then raised to the power of 10 for display. Variance and standard deviation remain in log form.

Correlation matrices are also calculated for all sampled and optimized parameters. The linear correlation coefficient (Pearson's *r*) and a rank correlation coefficient (Spearman's r) are

calculated for every combination of variable. For logarithmic data, the $\log_{10}$ of the data is taken before correlations are calculated. Pearson's *r* is given as:

$$r_p = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}} \qquad (3.5\text{-}1)$$

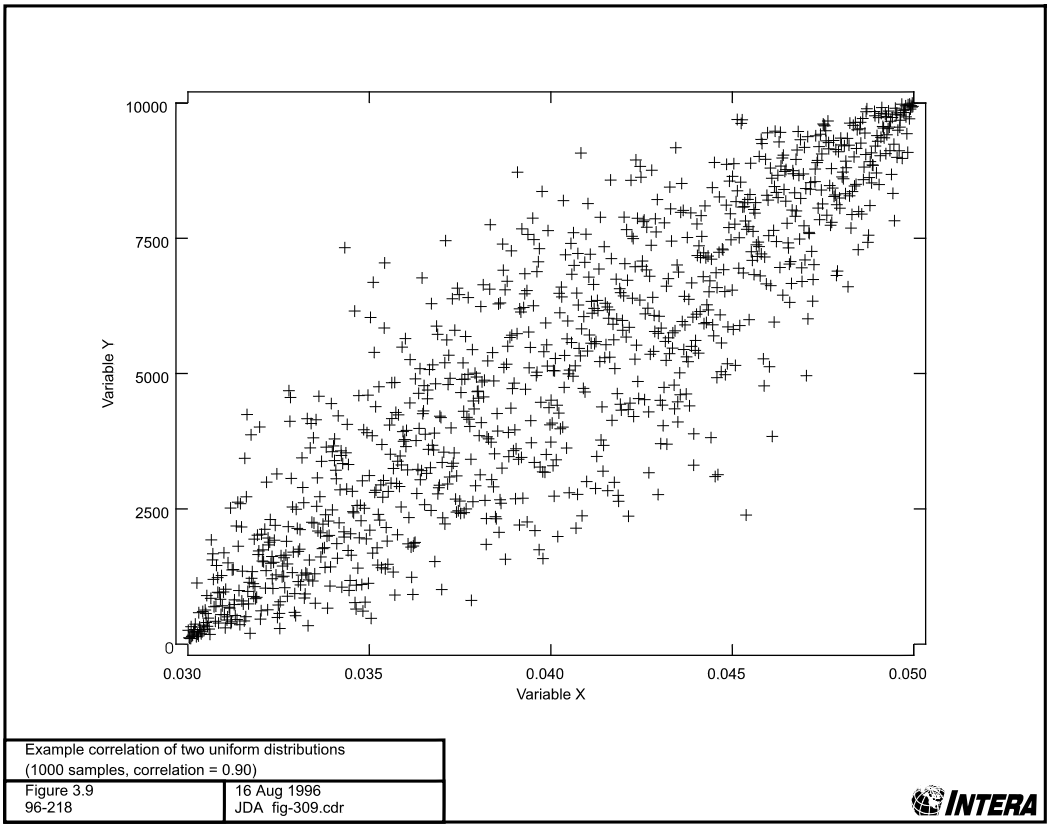



**Figure 3.11 Example correlation of two uniform distributions (1000 samples, correlation = 0.90)**

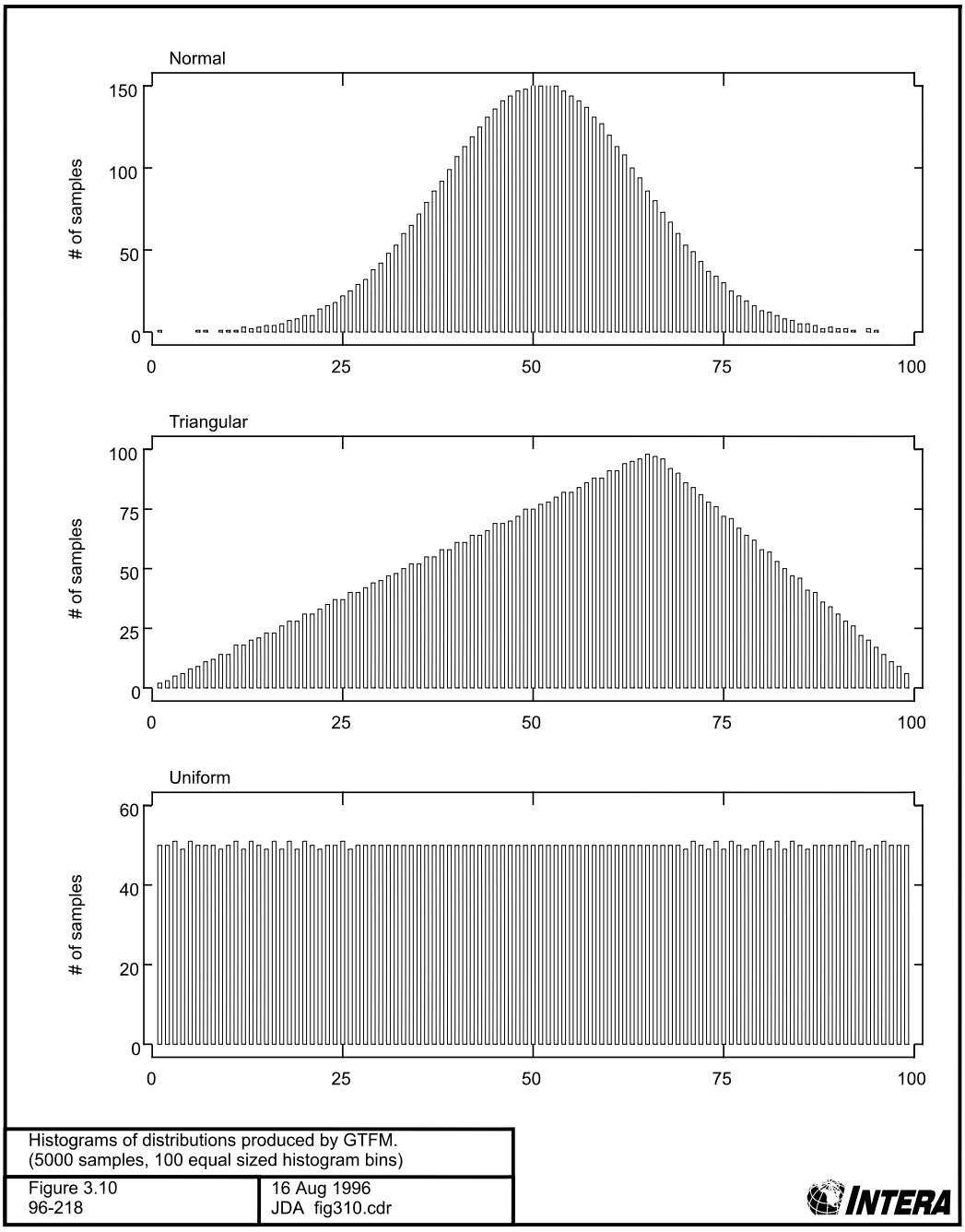



**Figure 3.12 Histograms of distributions produced by nSIGHTS (5000 samples, 100 equal sized histogram bins).**

Spearman's r is:

$$r_s = \frac{\sum_i (R_i - \overline{R})(S_i - \overline{S})}{\sqrt{\sum_i (R_i - \overline{R})^2}\sqrt{\sum_i (S_i - \overline{S})^2}} \qquad (3.5\text{-}2)$$

where :

$R_i$ = rank of $x_i$ among all x

$S_i$ = rank of $y_i$ among all y

## *3.6 Barometric Compensation*

The barometric compensation feature employs a deconvolution method to compensate for the effects of natural barometric fluctuations upon observed water levels. Multiple regression coefficients are derived from paired values of water levels and barometric pressure readings, which have been configured at matching regular time intervals.

The time of measurement, barometric pressure, and water level are all fed into the feature, along with two user settings; Lag and Memory, of the system. Based on these entries, the code produces a unit response function that fits the response model.

In this section, the equations for the regression deconvolution and pressure head adjustment are presented. The method for barometric response removal is presented followed by the modification to simultaneously remove Earth tide responses.

### 3.6.1 Barometric Response Function

To remove barometric effects, a barometric response function is required. Estimating the time-lag response between barometric pressure changes and water level responses in a well can be accomplished using regression deconvolution. A linear set of equations is established to estimate the unknown barometric response function:

$$\Delta WL(t) = u(0)\Delta BP(t) + u(1)\Delta BP(t-1) + u(2)\Delta BP(t-2) + \ldots + \Delta u(n)BP(t-n), \qquad (3.6\text{-}1)$$

where $\Delta WL(t)$ is the total head at time step $t$, $\Delta BP(t)$ is the change in barometric pressure at time $t$, $\Delta BP(t–i)$ is the change in barometric pressure $i$ time steps before t, $u(t)$ is the barometric pressure impulse function at lag $i$, and $n$ is the maximum lag.

For the instantaneous response, only the first term, $u(0)$, is used and all the other terms are zero. In general, the barometric response function has more than one lag term due to borehole storage effects and delays between the change in barometric pressure and the observed water level response. The maximum lag should be set to a large enough number so that all long-term

responses are included.

The response function is found using ordinary least squares linear regression. Once the values of $u(i)$ are found, then the cumulative response function $U(i)$ is calculated by summing the impulse responses:

$$U(i) = \sum_{j=1}^{i} u(j).$$

The cumulative response function is useful for diagnosing the aquifer type (confined or unconfined), borehole storage effects, well skin effects, and even for estimating aquifer hydraulic properties. Rasmussen and Crawford (1997) document the regression deconvolution method and its use as a diagnostic tool.

Using an extension of the regression deconvolution method, the fluctuations in the water level data caused by both Earth tides and barometric pressure effects can be removed by simultaneously using both observed barometric pressures and Earth tide gravity potential data. Earth tide potentials can be predicted for any location for any time using programs available on the Internet. They may also be measured directly at the site using appropriate sensors.

### 3.6.2 Data Preprocessing:

nSights defines a standard time increment, $\Delta t$, for both the barometric pressure, $BP$, and water level/pressure head observations (typically between 0.5 and 2.0 hours). A set of data differences are used to build the response function:

$$\Delta BP(t) = BP(t + \Delta t) - BP(t), \tag{3.6-2}$$

$$\Delta WL(t) = WL(t + \Delta t) - WL(t). \tag{3.6-3}$$

### 3.6.3 Regression Deconvolution:

1. Define a one dimensional vector, **y**, with dimension [$n$–$m$] using:

$$\mathbf{y} = \left[\Delta WL_{m+1},\ \Delta WL_{m+2},\ \Delta WL_{m+3},\ ...,\ \Delta WL_{n}\right], \tag{3.6-4}$$

   where $n$ is the total number of elements in the $\Delta WL$ and $m$ is the memory or maximum time delay of the system, which will not be known *a priori*. The value of $m$ should initially be set equal to 12, but may be selected by the user.

2. Define the two-dimensional matrix, **X**, with dimensions [$n$–$m$, $m$+1], using:

$$\mathbf{X} = \begin{bmatrix} \Delta BP_{m+1} & \Delta BP_{m+2} & \Delta BP_{m+3} & \cdots & \Delta BP_{n} \\ \Delta BP_{m+1-1} & \Delta BP_{m+2-1} & \Delta BP_{m+3-1} & \cdots & \Delta BP_{n-1} \\ \Delta BP_{m+1-2} & \Delta BP_{m+2-2} & \Delta BP_{m+3-2} & \cdots & \Delta BP_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta BP_{m+1-m} & \Delta BP_{m+2-m} & \Delta BP_{m+2-m} & \cdots & \Delta BP_{n-m} \end{bmatrix}. \tag{3.6-5}$$

3. Create a new matrix, **A**, with dimensions [$m$+1, $m$+1] such that

$$\mathbf{A} = \mathbf{X}^T \mathbf{X}, \tag{3.6-6}$$

where the superscript *T* indicates the transpose.

4. Create a new vector, **b**, with dimension [*m*+1] such that

$$\mathbf{b} = \mathbf{X}^T \mathbf{y}. \tag{3.6-7}$$

5. Solve for the one-dimensional *impulse response* vector, **u**, with dimension [*m*+1], using Gaussian elimination with matrix **A** and vector **b** as input.

   The Gaussian elimination function solves **A·xu = b** using scaled column pivoting. The inputs are a matrix **A** with dimensions [*m*+1, *m*+1] and vector **b** with dimension [*m*+1]. The Gaussian elimination function returns the unit response vector **u** with dimension [*m*+1] as well as the determinant of the **A** matrix.

6. Calculate the step (or cumulative) response function, $\underline{U}$, using:

$$\underline{U}(\tau) = -\sum_{j=0}^{i} u(j), \tag{3.6-8}$$

where $\tau = i\Delta t$ is the time-lag or delay in the barometric response.

### 3.6.4 Barometric Response Function Selection:

Plot the step response function as $\underline{U}$ vs. $\tau$ , on the screen. Regression deconvolution is repeated and a new cumulative response function is displayed on the screen each time a new value for *m* or a new time interval, $\Delta t$, is selected. Two general types of step response functions should be observed, one for confined and another for unconfined or semi-confined aquifers, shown in Figure 4-1. The plot is used to select the ideal response function for the data set. In confined aquifers, this is a response that has a maximum value at the *m* value selected by the user. In unconfined aquifers, the minimum is at the *m* value selected by the user (Figure 3.6-1). For a detailed description of the response functions, refer to the work of Rasmussen and Crawford (1997) or Spane (2002).

### 3.6.5 Pressure Head Adjustment

1. Once the desired response function has been obtained, calculate the water-level adjustment variable, $\Delta WL^*$, using:

$$\Delta WL^* = X \cdot u \tag{3.6-9}$$

2. Calculate the adjustment times series, $WL^*$, using:

$$WL^*(i) = \sum_{j=1}^{i} \Delta WL^*(j) \tag{3.6-10}$$

The result is a vector of corrections $WL^*$ with dimension [$n-m$].

3. Calculate the adjusted pressure heads, *AWL*, using:

$$WL(i) = \sum_{j=1}^{i} \Delta WL(j) \tag{3.6-11}$$

$$AWL(i) = WL(i) - WL^*(i+m) \tag{3.6-12}$$

### 3.6.6 Earth Tide Modification

The regression deconvolution technique was adapted to simultaneously remove the Earth tide response from water level data. The modification requires an Earth tide signal at the same interval as the pressure head measurements. For the purpose of this document the generation of the Earth tide signal is not addressed. It can be obtained from sensors installed in the vicinity of a site or generated synthetically. Typical units of the Earth tide signal are $L/T^2$ or acceleration. Resuming the analysis from Step 4 above, the Earth tide signal is defined as:

$$\Delta ET(t) = ET(t+\Delta t) - ET(t). \tag{3.6-13}$$

### 3.6.7 Earth Tide and Barometric Regression deconvolution

1. Define a one dimensional vector, **y**, with dimension [*n–m*] using:

$$\mathbf{y} = \left[\Delta WL_{m+1},\ \Delta WL_{m+2},\ \Delta WL_{m+3},\ ...,\ \Delta WL_{n}\right]. \tag{3.6-14}$$

2. The **y** vector remains the same as the barometric technique and the **X** matrix is expanded to the dimensions of [*n–m*, 2*m*+2] to include the $\Delta ET(t)$ array with form::

$$\mathbf{X} = \begin{bmatrix} \Delta BP_{m+1} & \Delta BP_{m+2} & \Delta BP_{m+3} & \cdots & \Delta BP_{n} \\ \Delta BP_{m+1-1} & \Delta BP_{m+2-1} & \Delta BP_{m+3-1} & \cdots & \Delta BP_{n-1} \\ \Delta BP_{m+1-2} & \Delta BP_{m+2-2} & \Delta BP_{m+3-2} & \cdots & \Delta BP_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta BP_{m+1-m} & \Delta BP_{m+2-m} & \Delta BP_{m+3-m} & \cdots & \Delta BP_{n-m} \\ \Delta ET_{m+1} & \Delta ET_{m+2} & \Delta ET_{m+3} & \cdots & \Delta ET_{n} \\ \Delta ET_{m+1-1} & \Delta ET_{m+2-1} & \Delta ET_{m+3-1} & \cdots & \Delta ET_{n-1} \\ \Delta ET_{m+1-2} & \Delta ET_{m+2-2} & \Delta ET_{m+3-2} & \cdots & \Delta ET_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta ET_{m+1-m} & \Delta ET_{m+2-m} & \Delta ET_{m+3-m} & \cdots & \Delta ET_{n-m} \end{bmatrix}. \tag{3.6-15}$$

3. Create a new matrix, **A**, with dimensions [2*m*+2, 2*m*+2] such that

$$\mathbf{A} = \mathbf{X}^T\mathbf{X}. \tag{3.6-16}$$

4. Create a new vector, **b**, with dimension [2*m*+2] such that

$$\mathbf{b} = \mathbf{X}^T\mathbf{y}. \tag{3.6-17}$$

5. Solve for the one-dimensional *impulse response* vector, **u**, with dimension [2*m*+2], using Gaussian elimination with matrix **A** and vector **b** as input.

   The Gaussian elimination function solves **A·xu = b** using scaled column pivoting. The inputs are the matrix **A** with dimensions [2*m*+2, 2*m*+2] and vector **b** with dimension [2*m*+2]. The Gaussian elimination function returns the unit response vector **u** with dimension [2m+2] and the determinant of the **A** matrix.

6. Calculate the step (or cumulative) response function, $\underline{U}$, using:

$$\underline{U}(\tau) = -\sum_{j=0}^{i} u(j) \tag{3.6-18}$$

   where $\tau = i\Delta t$ is the time-lag or delay in the barometric response.

### 3.6.8 Barometric Response Function Selection:

Plot the step response function as $\underline{U}$ vs. $\tau$ to the screen. Regression deconvolution is repeated and a new cumulative response function is displayed on the screen each time a new value for *m* or a new time interval, $\Delta t$, is selected. In general, the cumulative response function for the simultaneous deconvolution of *ET* and *BP* will be similar in shape to the barometric only response function of a well. Exceptions occur in wells where the Earth tide response is much larger than the barometric pressure effect. The principle Earth tide components which cause forces that affect water levels occur in periods shorter then 12 hours. Accordingly a maximum lag time of 12 hours tends to be ideal in removing barometric and Earth tide components.

**Pressure Head Adjustment**

1. Once the desired response function has been obtained, calculate the water-level adjustment variables for *ET* and *BP*, $\Delta WL^{BP*}$ and $\Delta WL^{ET*,}$ using:

$$\text{vector } \mathbf{u}_{BP} = \left[u_1 \cdots u_{m+1}\right], \text{ with dimension } [m+1] \tag{3.6-19}$$

$$\text{vector } \mathbf{u}_{ET} = \left[u_{m+2} \cdots u_{2m+2}\right], \text{ with dimension } [m+1] \tag{3.6-20}$$

$$\mathbf{X}_{BP} = \begin{bmatrix} \Delta BP_{m+1} & \Delta BP_{m+2} & \Delta BP_{m+3} & \cdots & \Delta BP_{n} \\ \Delta BP_{m+1-1} & \Delta BP_{m+2-1} & \Delta BP_{m+3-1} & \cdots & \Delta BP_{n-1} \\ \Delta BP_{m+1-2} & \Delta BP_{m+2-2} & \Delta BP_{m+3-2} & \cdots & \Delta BP_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta BP_{m+1-m} & \Delta BP_{m+2-m} & \Delta BP_{m+2-m} & \cdots & \Delta BP_{n-m} \end{bmatrix}, \text{ dimensions } [n-m,\ m+1] \quad (3.6\text{-}21)$$

$$\mathbf{X}_{ET} = \begin{bmatrix} \Delta ET_{m+1} & \Delta ET_{m+2} & \Delta ET_{m+3} & \cdots & \Delta ET_{n} \\ \Delta ET_{m+1-1} & \Delta ET_{m+2-1} & \Delta ET_{m+3-1} & \cdots & \Delta ET_{n-1} \\ \Delta ET_{m+1-2} & \Delta ET_{m+2-2} & \Delta ET_{m+3-2} & \cdots & \Delta ET_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta ET_{m+1-m} & \Delta ET_{m+2-m} & \Delta ET_{m+2-m} & \cdots & \Delta ET_{n-m} \end{bmatrix}, \text{ dimensions } [n-m,\ m+1] \quad (3.6\text{-}22)$$

$$\Delta WL^{BP*} = \mathbf{u}_{BP} X_{BP} \quad (3.6\text{-}23)$$

$$\Delta WL^{ET*} = \mathbf{u}_{ET} X_{ET} \quad (3.6\text{-}24)$$

2. Calculate the adjustment times series, $WL^{*}$, using:

$$WL^{BP*}(i) = \sum_{j=1}^{i} \Delta WL^{BP*}(j) \quad (3.6\text{-}25)$$

$$WL^{ET*}(i) = \sum_{j=1}^{i} \Delta WL^{ET*}(j) \quad (3.6\text{-}26)$$

$$WL^{*}(i) = WL^{BP*}(i) + WL^{ET*}(i) \quad (3.6\text{-}27)$$

3. Calculate the adjusted pressure heads, *AWL*, using:

$$WL(i) = \sum_{j=1}^{i} \Delta WL(j) \quad (3.6\text{-}28)$$

$$AWL(i) = WL(i) - WL^{*}(i+m) \quad (3.6\text{-}29)$$

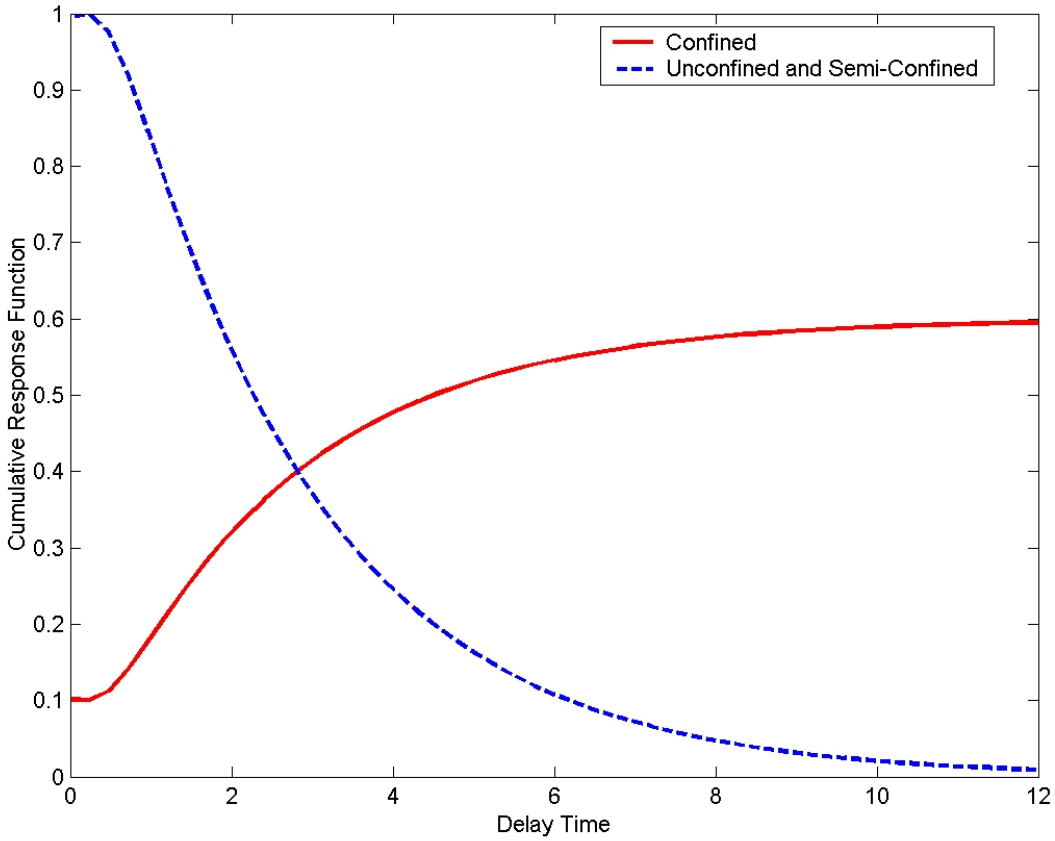



Figure 3.6-1. Response Functions.

# REFERENCES FOR APPENDIX A

Argawal, R.G. 1980. A new approach to account for producing time effects when drawdown curves are used to analyze pressure buildup and other test data. SPE 9289

Barker, J. A., 1988. A generalized radial flow model for hydraulic tests in fractured rock. Water Resources Research, Vol 24, No 10, pg 1796-1804

Clark, D.G. and T.D. van Golf-Racht, 1985. Pressure derivative approach to transient test analysis: A high-permeability North Sea reservoir example. J. Pet. Tech., Nov 1985.

Horner, D.R., 1951. Pressure build-up in wells. Proc. Third World Pet. Cong. The Hague II, pp 503 - 521

Iman, R.L. and Shortencarier M.J., A FORTRAN 77 Program and User's Guide for the Generation of Latin Hypercube and Random Samples for Use with Computer Models. SAND83-2365, NUREG/CR-3624, Sandia National Laboratories, Albuquerque, NM, 1984.

Press, W.H, S.A. Teukolsky, W.T.Vetterling, B.P.Flannery, Numerical Recipes, Second Edition, Cambridge University Press, 1992.

Rasmussen, T. C. and L. A. Crawford, 1997, Identifying and removing barometric pressure effects in confined and unconfined Aquifers, *Ground Water*, **35**(*3*):502–511.