N90 New Mexico Bureau
of
Geology and Mineral Resources

# CAPTURE ZONES IN TRANSIENT FLOW FIELDS: SIMULATIONS AND ANALYSIS

*by*

*Shawn C. Leppert*

New Mexico Institute of Mining and Technology
Socorro, New Mexico

January, 1990

Submited in Partial Fulfillment
of the Requirements for the Degree
Master of Science in Hydrology

## ABSTRACT

An accurate particle tracking algorithm is developed and coupled with a dynamic front tracker to delineate capture zones in transient flow systems. Pumping and recharge rates are the transient parameters. A sensitivity analysis is performed, implementing variations in wave form, amplitude and period. Period capture zones, extraction probabilities, and effective stagnation points are determine to describe the transient capture zones. Simulated 0% boundaries are compared to steady state ultimate capture zones. A method using the effective stagnation point is developed to predict front patterns and qualitative estimates of extraction probabilities. This method is used to adequately determine front patterns. There are three front patterns; connected, intermediate connected and disconnected. When a sinusoidal variation in a parameter is simulated the fronts mimics the size and shape of the fronts generated by the mean parameter under steady state conditions. Transients complicate the system enough to require at least a quasi-steady state conceptualization. The algorithm applied is a useful tool when simple numerical models are all that is required. It produces accurate fronts and is extremely versatile.

## ACKNOWLEDGEMENTS

This author wishes to extend his sincere thanks to his advisor, Dr. John Wilson, for his patience, expertise, and especially his willingness to give me a chance at New Mexico Tech. John's intensity and intelligence inspired me to be persistent in achieving the goals I set for myself. Bill Linderfelt deserves special thanks for being an excellent research partner and comrade. He contributed extensively to the ideas presented and was always willing to discuss theoretical problems, whether at two o'clock in the morning or afternoon. Annette Scafer-Perini's assistance, especially in the early stages of the research gave me an excellent approach to modeling. She was always willing to help me to continue my growth in the computer age. The person I credit my success to is my wife, Miki. She has been there through the toughest of times, we have weathered the storms together. Without her urging me on this research would probably never have ended, thank you all.

# CAPTURE ZONES IN TRANSIENT FLOW FIELDS: SIMULATIONS AND ANALYSIS

*by*
*Shawn C. Leppert*

New Mexico Institute of Mining and Technology
Socorro, New Mexico
January, 1990

## Table of Contents

# LIST OF FIGURES

24.1 18C Probabilities associated with the simulation, time on $t_P = 0.04\tau_{dim}$. The lightly dotted zone is the passed zone. There is no 100%. The 0% boundary is represented by the dotted curve and the mean UCZ by the chain-dotted curve. The extraction probability cross-section are for the transects A-B and C-D, where H is the probability associated with the x-position (right) and y-position (left), the graph and is determined by the % of particles that will reach the well given an infinite amount of time.

24.2 18D Ultimate capture zone comparison graph for the simulation time on $t_P = 0.04\tau_{dim}$. The lightly dotted zone is the passed zone. The 0% boundary is represented by the dotted curve. The chain-dotted curves represent the ultimate capture zones for the mean pumping rate (top) and the effective stagnation point mean (bottom).

24.3 19A Period capture zones. This simulation is the reduced pumping rate with a time on $t_P = 0.8\tau_{dim}$. The pumping schedule is presented in dimensionless parameters $\tau = \dfrac{\textit{dimensionless current time}}{1.3\tau_{dim}}$ and the pumping rate ($Q_w$) normalized by the mean pumping rate ($\overline{Q_w}$) for the first example (0.0603).

24.4 19B Diagnostic front positions for a reduced pumping rate simulation with a time on $t_P = 0.8\tau_{dim}$. Inset A shows the position of the front at the time the pump is reduced and at the end of the period (they do not intersect). Inset B illustrates the front pattern.

24.5 19C Probabilities associated with a reduced pumping simulation, time on $t_P = 0.8\tau_{dim}$. The lightly dotted zone is the passed zone. 100% boundary is represented by the dashed curve, the 0% boundary by the dotted curve, and the mean UCZ by the chain-dotted curve. The extraction probability cross-section is for the transect A-B, where H is the probability associated with the x position on the graph and is determined by the % of particles that will reach the well given an infinite amount of time.

24.6 19D Ultimate capture zone comparison graph for the reduced pumping simulation time on $t_P = 0.8\tau_{dim}$. The lightly dotted zone is the passed zone. 100% boundary is represented by the dashed curve and the 0% boundary by the dotted curve. The chain-dotted curves represent the ultimate capture zones for the mean pumping rate (top) and the effective stagnation point mean (bottom)

25.1 20A Period capture zones. This simulation is the significantly reduced pumping rate with a time on $t_P = 0.5\tau_{dim}$. The pumping schedule is presented in dimensionless parameters $\tau = \dfrac{\textit{dimensionless current time}}{1.3\tau_{dim}}$ and the pumping rate ($Q_w$) normalized by the mean pumping rate ($\overline{Q_w}$) for the first example (0.0603).

25.2 20B Diagnostic front positions for a significantly reduced pumping rate simulation with a time on $t_P = 0.5\tau_{dim}$. Inset A shows the position of the front at the time the pump is reduced and at the end of the period (they do not intersect). Inset B illustrates the front pattern.

25.3 20C Probabilities associated with a reduced pumping simulation, time on $t_P = 0.5\tau_{dim}$. The lightly dotted zone is the passed zone. 100% boundary is represented by the dashed curve, the 0% boundary by the dotted curve, and the mean UCZ by the chain-dotted curve. The extraction probability cross-sections are for the transects A-B and C-D, where H is the probability associated with the x-position (right) and y-position (left) on the graph and is determined by the % of particles that will reach the well given an infinite amount of time.

25.4 20D Ultimate capture zone comparison graph for the significantly reduced pumping simulation time on $t_P = 0.5\tau_{dim}$. The lightly dotted zone is the passed zone. 100% boundary is represented by the dashed curve and the 0% boundary by the dotted curve. The chain-dotted curves represent the ultimate capture zones for the mean pumping rate (top) and the effective stagnation point mean (bottom).

25.5 21A Period capture zones. This simulation is the randomly generated pumping rate with a time of change = $0.5\tau_{dim}$. The pumping schedule is presented in dimensionless parameters $\tau = \frac{dimensionless\ current\ time}{1.3\tau_{dim}}$ and the pumping rate ($Q_w$) normalized by the mean pumping rate ($\overline{Q_w}$) for the first example (0.0603).

25.6 21B Diagnostic front positions for a randomly generated pumping rate simulation with a time of change = $0.5\tau_{dim}$. Inset A shows the position of the front at the time the pump is varied and at the end of the period (they do not intersect). Inset B illustrates the front pattern.

25.7 21C Probabilities associated with a randomly generated pumping rate simulation, time changed = $0.5\tau_{dim}$. The lightly dotted zone is the passed zone. 100% boundary is represented by the dashed curve, the 0% boundary by the dotted curve, and the mean UCZ by the chain-dotted curve. The extraction probability cross-sections are for the transects A-B and C-D, where H is the probability associated with the x-position (right) and y-position (left) on the graph and is determined by the % of particles that will reach the well given an infinite amount of time.

25.8 21D Ultimate capture zone comparison graph for randomly generated pumping rate simulation time change = $0.5\tau_{dim}$. The lightly dotted zone is the passed zone. 100% boundary is represented by the dashed curve and the 0% boundary by the dotted curve. The chain-dotted curves represent the ultimate capture zones for the mean pumping rate (top) and the effective stagnation point mean (bottom).

25.9 22A Period capture zones. This simulation is the continuous sine wave pumping rate. The pumping schedule is presented in dimensionless parameters $\tau = \frac{dimensionless\ current\ time}{1.3\tau_{dim}}$ and the pumping rate ($Q_w$) normalized by the mean pumping rate ($\overline{Q_w}$) for the first example (0.0603).

25.10 22B Diagnostic front positions for a continuous sine wave pumping rate simulation. Inset A shows the position of the front at the time the pump is varied and at the end of the period (they do not intersect). Inset B illustrates the front pattern.

25.11 22C Ultimate capture zone comparison graph for a continuous sine wave pumping rate simulation. 100% boundary is represented by the dashed curve. The chain-dotted curve represent the ultimate capture zones for the mean pumping rate.

26.1 23 Effective stagnation point as a function of the dimensionless off time. The effective stagnation points are generated from a flow field with a pumping rate $Q_w = 0.157$ and an dimensionless on time $t_P = 0.5$. y-coordinate is 0. The triangle represents the off time that creates an effective stagnation point on the boundary of the first period capture zone at t = 0. The square is the off time when the effective stagnation point ceases to exist. The stagnation point for a constant pumping rate of 0.157.

27.1 24A Period capture zones. This simulation a pumping rate of $Q_w = 0.157$, time on $t_P = 0.5\tau_{dim}$, P=1.14. The pumping schedule is presented in dimensionless parameters

$\tau = \dfrac{\textit{dimensionless current time}}{1.3\tau_{dim}}$ and the pumping rate ($Q_w$) normalized by the mean pumping rate ($\overline{Q_w}$) for the first example (0.0603).

27.2 24B Diagnostic front positions for a simulation with a pumping rate of $Q_w$ = 0.157, time on $t_P = 0.5\tau_{dim}$, P=1.14. Inset A shows the position of the front at the time the pump is the pump is turned on and at the end of the period. Inset B illustrates the front pattern.

27.3 25A Period capture zones. This simulates a pumping rate of $Q_w$ = 0.157, time on $t_P = 0.5\tau_{dim}$, P=1.24. The pumping schedule is presented in dimensionless parameters $\tau = \dfrac{\textit{dimensionless current time}}{1.3\tau_{dim}}$ and the pumping rate ($Q_w$) normalized by the mean pumping rate ($\overline{Q_w}$) for the first example (0.0603).

27.4 25B Diagnostic front positions for a simulation with a pumping rate of $Q_w$ = 0.157, time on $t_P = 0.5\tau_{dim}$, P=1.24. Inset A shows the position of the front at the time the pump is the pump is turned on and at the end of the period. Inset B illustrates the front pattern.

28.1 26A Period capture zones. This simulates a pumping rate of $Q_w$ = 0.157, time on $t_P = 0.5\tau_{dim}$, P=1.46. The pumping schedule is presented in dimensionless parameters $\tau = \dfrac{\textit{dimensionless current time}}{1.3\tau_{dim}}$ and the pumping rate ($Q_w$) normalized by the mean pumping rate ($\overline{Q_w}$) for the first example (0.0603).

28.2 26B Diagnostic front positions for a simulation with a pumping rate of $Q_w$ = 0.157, time on $t_P = 0.5\tau_{dim}$, P=1.46. Inset A shows the position of the front at the time the pump is the pump is turned on and at the end of the period. Inset B illustrates the front pattern.

28.3 26C Probabilities associated with the pumping simulation, $Q_w$ = 0.157, time on $t_P = 0.5\tau_{dim}$, P=1.46. The lightly dotted zone is the passed zone. The 0% boundary is represented by the dotted curve, and the mean UCZ by the chain-dotted curve. The extraction probability cross-sections are for the transects A-B and C-D, where H is the probability associated with the x position on the graph and is determined by the % of particles that will reach the well given an infinite amount of time.

28.4 27A Period capture zones. This simulates a pumping rate of $Q_w$ = 0.157, time on $t_P = 0.5\tau_{dim}$, P=1.9. The pumping schedule is presented in dimensionless parameters $\tau = \dfrac{\textit{dimensionless current time}}{1.3\tau_{dim}}$ and the pumping rate ($Q_w$) normalized by the mean pumping rate ($\overline{Q_w}$) for the first example (0.0603).

28.5 27B Diagnostic front positions for a simulation with a pumping rate of $Q_w$ = 0.157, time on $t_P = 0.5\tau_{dim}$, P=1.9. Inset A shows the position of the front at the time the pump is the pump is turned on and at the end of the period. Inset B illustrates the front pattern.

28.6 27C Probabilities associated with the pumping simulation, $Q_w$ = 0.157, time on $t_P = 0.5\tau_{dim}$, P=1.9. The lightly dotted zone is the passed zone. The 0% boundary is represented by the dotted curve, and the mean UCZ by the chain-dotted curve. The extraction probability cross-sections are for the transects A-B and C-D, where H is the probability associated with the x position on the graph and is determined by the % of particles that will reach the well given an infinite amount of time.

28.7 28A Period capture zones. This simulates a pumping rate of $Q_w$ = 0.157, time on $t_P = 0.5\tau_{dim}$, P=4.0. The pumping schedule is presented in dimensionless parameters $\tau = \dfrac{\textit{dimensionless current time}}{1.3\tau_{dim}}$

and

the pumping rate ($Q_w$) normalized by the mean pumping rate ($\overline{Q_w}$) for the first example (0.0603).

28.8 28B Probabilities associated with the pumping simulation, $Q_w = 0.157$, time on $t_P = 0.5\tau_{dim}$, P=4.0. The lightly dotted zone is the passed zone. The 0% boundary is represented by the dotted curve, and the mean UCZ by the chain-dotted curve. The extraction probability cross-sections are for the transects A-B and C-D, where H is the probability associated with the x position on the graph and is determined by the % of particles that will reach the well given an infinite amount of time.

29.1 29 Period capture zones. This simulates a sinusoidal ambient flow rate where $Q_w = 0.157$, time on $t_P = 0.5\tau_{dim}$, P=1.3. The ambient flow fluctuations are presented in dimensionless parameters $\tau = \dfrac{\textit{dimensionless current time}}{1.3\tau_{dim}}$ and the ambient flow rate ($q_a$) normalized by the mean ambient flow rate ($\overline{q_a}$).

# INTRODUCTION

Many supply wells throughout the United States have transient pumping schedules. There are many reasons for varying the rate at which water is withdrawn from an aquifer. It may be defined by the temporal demands of a community or industry that uses the water, or surface storage limitations. Supply wells are often operated in a way that is cost effective, such as pumping only at periods when there is reduced demands on public utilities. Other parameters may also exhibit variations in time. For example vertical recharge and ambient flow rate. Flucuations are often climatically controlled. Variations may occur on the order of minutes, days, or months. A pumping well may be turned on and off, or the rate may simply reduced by half during periods of reduction. Natural variations may be a continuous function of time. There are an infinite number of possibilities.

Recently attitudes concerning contamination of groundwater supplies has switched from remediation to protection. In 1986 the United States Congress passed the Safe Drinking Water Act which provides for well-head protection. Communities are beginning to implement controls on the types of industry they will allow up-gradient of their drinking water supply. In the last few years quite a bit of research (much of it conducted by NMIMT) has been directed towards creating tools to assist in the designation of well-head protection areas. These zones define the region near the well that contributes to the drinking water supply. Most of the models used to delineate these areas are based on steady state conceptualizations. They ignore transient parameters. The purpose of this independent study is to determine if this steady state approach is a viable one when certain parameters influencing the flow field are time dependent.

Transients can be very complex and often require in-depth modeling if accurate results are required. On the other hand, simple hybrid analytical-numerical approaches can be a cost effective approach when used as a screening tool to decide whether a large scale numerical model should be employed. The hybrid approaches can be very useful in getting at gross flow behavior such as deciding where the fluid is being drawn from and the path it takes to the well. The primary goals of this study are to develop a generic 2-dimensional computer model that can simulate simple transient conceptualizations, and to apply it to determining how sensitive well-head protection areas are to transient conditions. The approach taken is a graphical technique that describes the position of these well-head protection areas or capture zones. Therefore, plan views of flow domains and their associated capture zones are used to demonstrate the results of the simulations. An infinite flow domain is used. This is the most generic conceptualization possible. Therefore, the results of this study could be applied to more than well-head protection areas. Contaminant recovery, responsible party investigations, and resource management interests may all apply the techniques utilized in this paper.

Particle and dynamic front tracking are the primary tools used to produce the capture zones. The velocities are derived from simple analytical solutions. Storage is neglected. It is assumed that the flow field responds instantaneously to changes within the domain. The aquifer is assumed to be homogeneous and isotropic, with a fully penetrating well, fluid flow is purely advection driven, and the flow is predominantly horizontal. The steady state capture zone is assumed to be exactly the ground water divide but this is often not the case in a transient system.

The sensitivity analysis will be presented in two basic parts, first an engineering approach where the volume of water extracted is constant. The information derived from those simulations will then be examined to determine the dynamics of the flow field. The conclusions attained by the first approach will be applied to further simulations.

## MATHEMATICAL DEVELOPMENT

The particle tracking algorithm applied in this paper needs only a velocity field to create individual particle pathlines. The algorithm is independent the analytical or numerical method used to calculate the point velocities , although only analytical velocities are used here. 2-dimensional pathlines have been simulated using the particle tracking routine (described below) for many different boundary conditions including: semi-infinite (infinite width, stream at x= 0, and infinite in the positive x direction), stream-barrier (infinite width strip, stream at x= 0, and a barrier at some x= L), semi-infinite barrier (same as the semi-infinite but using a barrier instead of a stream), and stream-stream (infinite width, stream at x= 0 and at x= L). The point velocities are derived using conformal mapping, image well theory, and superposition (Lee, 1986; J. Wilson, 1988, personal communication). For the purposes of this paper, infinite boundaries conditions coupled with an infinite flow field will be used. Infinite boundary conditions (domain is infinite in all directions) with the fewest degrees of freedom are implemented for a couple of reasons. First it is the most generic approach. Second the analytical solution is trivial lending itself for easy manipulation when trying to determine gross fluid behavior.

The 2-dimensional model applied here assumes an isotropic, homogeneous aquifer. The well is assumed to be fully penetrating and the flow is considered to be predominantly horizontal. There are two driving forces that influence the flow field in a region near a well under these conditions, those due to the pumping well and those due to the lateral ambient flow field. This paper uses the clssical analytical solution to the potential flow field, such as presented by Bear, 1972. The potential flow field is constructed using superposition theory. The potential due to the ambient flow ($\Phi_a$) is derived from the complex potential:

$$f(z) = -q_a z = (-q_a x)-(i\, q_a y) \quad (1)$$

where,

$q_a$ = ambient flow velocity, assumed to be uniform in the negative x-direction

x,y = position coordinates

$z = x + iy$

The equipotenials are defined by the real component of equation (1):

$$\Phi_a = -q_a x \quad (2)$$

and the stream function $\Psi_a$ is given by the imaginary component of equation (1):

$$\Psi_a = -q_a y \quad (3)$$

thus the x and y-diection velocities at any point in the flow field is derived by the partial differential equations:

$$q_x = -\frac{\partial \Phi}{\partial x} = q_a \quad (4)$$

$$q_y = -\frac{\partial \Phi}{\partial y} = 0 \quad (5)$$

The velocity at any point is simply the ambient flow velocity. Velocities due to a pumping well is simulated by the complex potential:

$$f(z) = M\, ln(z) = M\, (ln(r) + (i\, \theta)) \quad (6)$$

where,

$$M = \frac{Q_w}{2\pi}$$

$Q_w$ = pumping rate

r = radius from well

$\theta$ = the angle from 0

The equipotentials are circles radiating from the well:

$$\Phi_w = M^{-}\ln(r) \quad (7)$$

and the streamlines are rays given by:

$$\Psi_w = M\, \theta \quad (8)$$

Utilizing superposition the combined complex potentials $\varsigma$ adds equation (1) and (6):

$$\varsigma = (-q_a\, z) + (M\, ln(z)) \qquad M \geq 0 \quad (9)$$

the steady state equipotential for a source located at 0,0 are simulated from the equation:

$$\Phi = (-q_a\,x) + M\,ln\,(r) = (-q_a\,x) + (\frac{M}{2}\,ln\,(x^2 + y^2)) \quad (10)$$

and the streamlines are given by:

$$\Psi = (-q_a\,y) + (M\,\tan^{-1}(\frac{y}{x})) \quad (11)$$

Finally, the x and y direction velocities are calculated by the derivatives of (10) for the x and (11) for the y:

$$q_x = -\frac{\partial\Phi}{\partial x} = q_x = q_a\text{-}(M\,\frac{x}{x^2 + y^2}) \quad (12)$$

$$q_y = -\frac{\partial\Phi}{\partial y} = q_y = -(M\,\frac{y}{x^2 + y^2}) \quad (13)$$

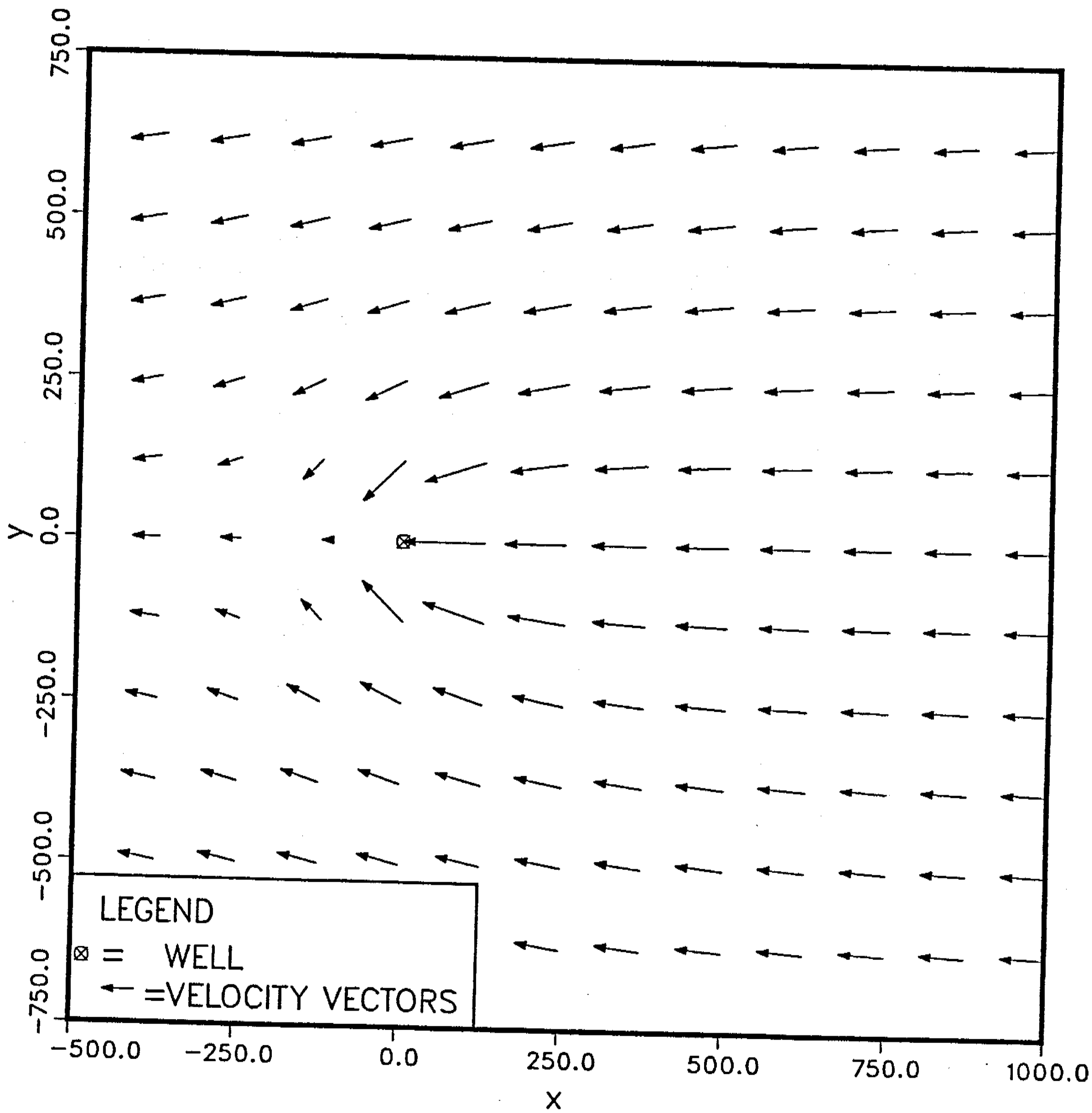
Figure (1) illustrates the magnitude of the velocities for an infinite flow field with a pumping well at the position 0,0. The pumping rate is $0.157\left[\frac{meters^3}{second}\right]$ and an ambient flow rate of $0.0002\left[\frac{meters^2}{second}\right]$ The x and y-axis is in meters relative to the well. The velocity vectors begin at the point where the velocity has been calculated, and the length of the vector is normalized by the maximum velocity in the flow field multiplied a constant equal to the distance between the positions.

As will be explained in later sections, the stagnation point(s) in the system must be delineated to inable capture zone delineation. For the infinite flow regime case, this is done by setting the x and y velocities to zero and solving for the x position (the y position will always be on the line y = 0). The equation for the stagnation point then becomes:

$$x_{stag} = \frac{M}{q_a} \quad (14)$$

$$y_{stag} = 0.0 \quad (15)$$

Using the equation above coupled with the particle tracking algorithm (discussed later) the capture zones for any pumping rate and ambient flow rate can be simulated. Of course, the accuracy of the capture zones is limited by the ability of the analytical solution to accurately describe the true flow field. The more realistic the analytical solution the better the particle tracking simulation. As will be


750.0
500.0
250.0
0.0
−250.0
−500.0
−750.0
Y
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
X
LEGEND
= WELL
=VELOCITY VECTORS

shown, the particle tracking scheme can be very precise.

## NUMERICAL METHODS

### Particle Tracking

A particle tracking scheme is implemented in this paper to define pathlines of particles within the flow field. It is also used to generate equal time front contours. Movement is assumed to be purely advective as a function of seepage velocity. The numerical approach used in the particle tracking is an Euler type approximation, very similar to the algorithm applied by Lee, 1986. This technique was chosen for its simplicity, and versatility. It handles vertical recharge very well. Although the infinite flow field has no vertical recharge the results of this study will be compared in subsequent studies to systems that do have a vertical recharge component.

Lee's particle tracking algorithm followed this flow chart:

A.) Input parameters

   1.) pumping rate, ambient flow rate, depth, and porosity

   2.) forward or reverse tracking flag

   3.) length constant

B.) Calculate the x and y-direction velocities at current position using the appropriate boundary conditions.

C.) Multiply the velocity by the length constant to calculate the time step

D.) Traject the particle from the current position to the new position a distance equal to the length constant

E.) Check to see if the particle is at a boundary or at the well

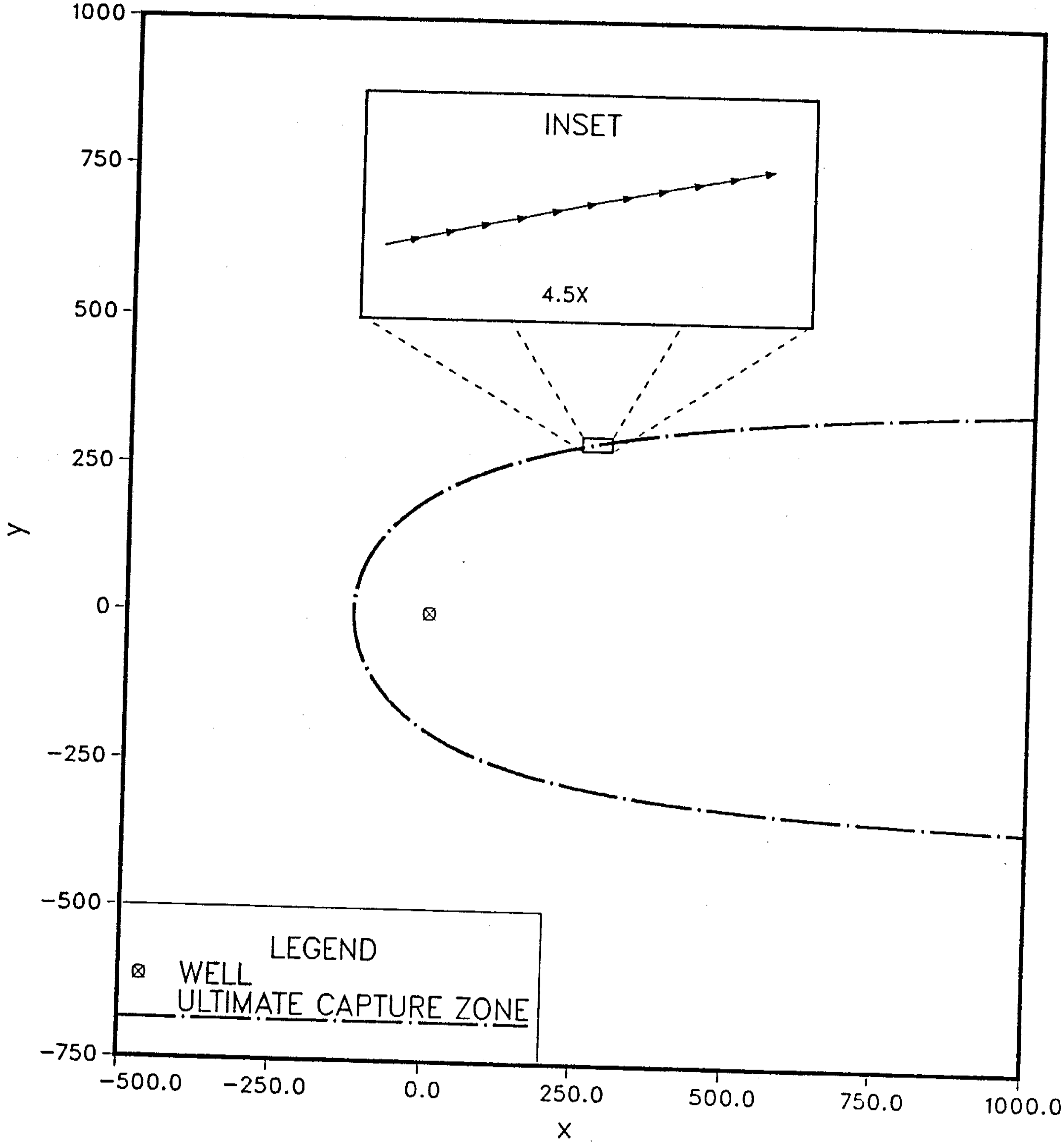
   1.) stop tracking if is sufficiently close

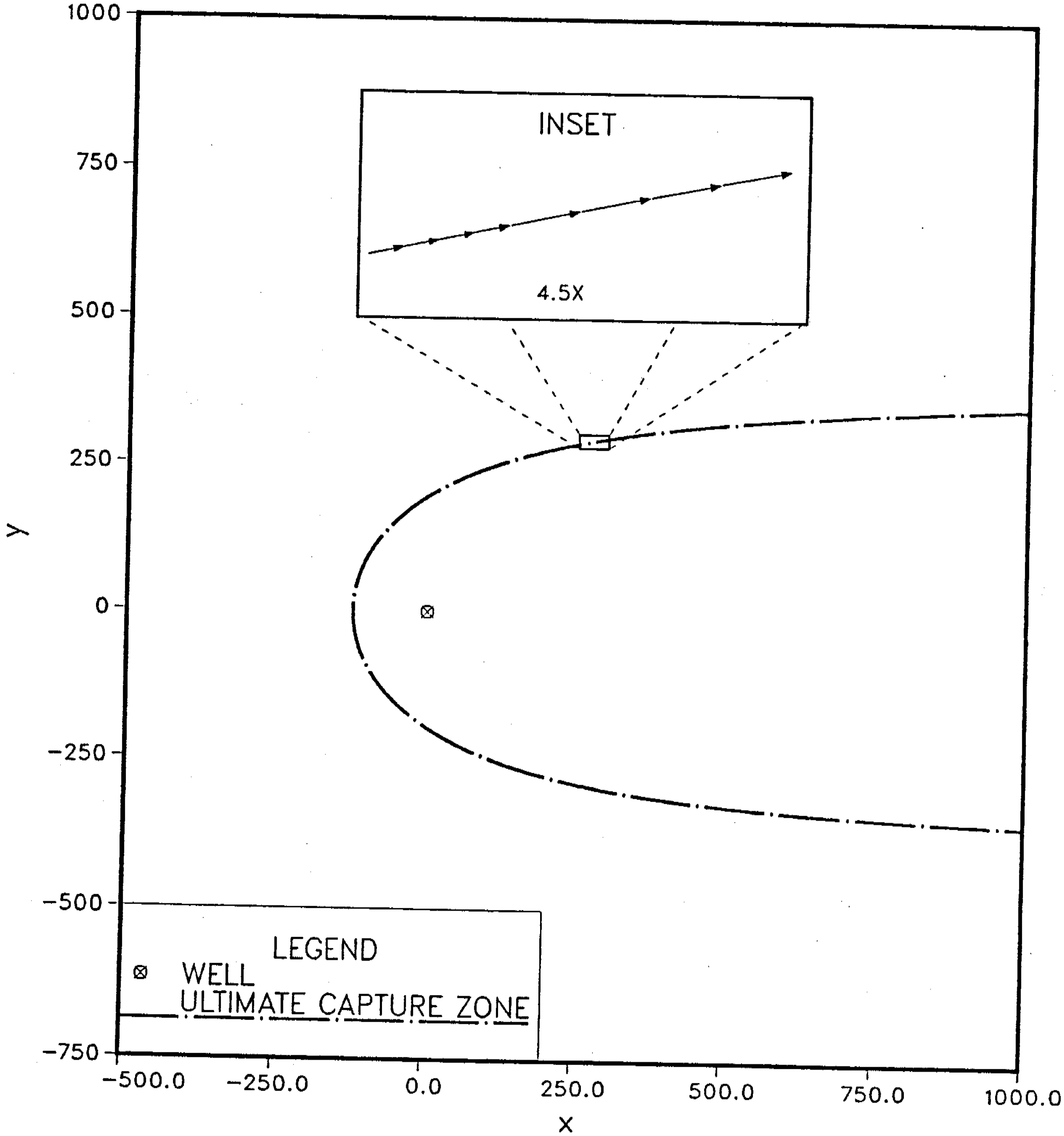
F.) Go to B

This technique is a self correcting approach when back-tracking is applied. That is, when trajecting away from the pumping well the streamlines converge. This helps to minimize the effects of drifting inherent in Euler type approximations.

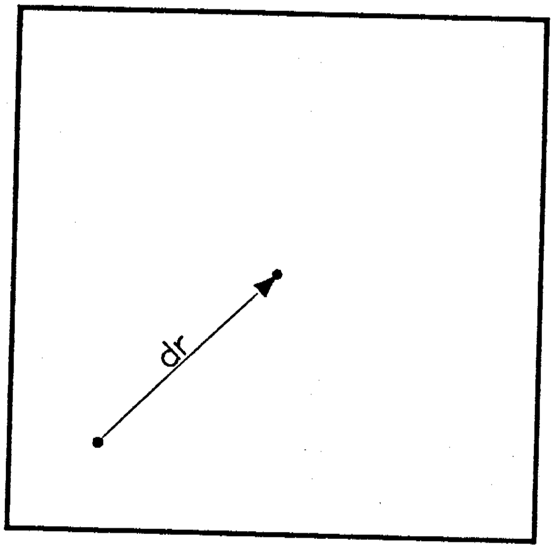
Transients complicate the tracking technique sufficiently to require a more complex particle tracking routine. Therefore, this algorithm was adapted to be compatible with the more complex transient flow regimes. Dividing the program into specific task subroutines increases its versatility and expands its application capabilities. A subroutine that maximizes/minimizes the length constant during each time step is coupled with the original program to increase precision. Figure (2) illustrates the constant length approach to particle tracking. In the inset the time step is noted by the arrowheads and the distance between the arrowheads is exactly equal to the length constant. The example is the same as explained in the mathematical development section. The specific pathline (streamline) example is the stagnation streamline (explained later). In this example back-tracking is used, explaining why the arrows point away from the well. This can be a v ery inefficient way to particle track. The approach taken in this paper is to adjust the length constant continuously so that size of the step reflects the level of divergence at the current position. Therefore, near the well where the velocity in the y-direction $v_y$ becomes significant the length "constant" decreases. Conversely, when the $v_y$ is very small the length "constant" is increased proportionally. Figure (3) illustrates the implementation of this technique using the same example.

Adaptive control over the time step is acheived by testing how sensitive the final position is to the size of the length "constant" using a step doubling procedure (Zheng, 1988). Figure (4) illustrates this technique. In figure (4A) the initial length constant is designated dr. First, the constant is minimized. The constant is divided in half and the position after two steps using the adjusted constant (figure 4B) is determined. The deviation distance between the two alternative positions is computed. If this deviation distance is within some tolerance the particle is tracked to the new position using the original length "constant". If not, the proceedure is repeated until the tolerance limits are met. If the deviation distance is within the tolerance in the first iteration, the constant is then maximized (figure 4C). This is realized in a similar manner. The particle is tracked two steps at the current constant, then the constant is doubled and the particle tracked from the current position. Again, if the deviation in the final positions is within the tolerance limit the tracking continues (figure 4D) using the original constant. If the limit is not met the constant is continually increased until it is. This approach produces more a

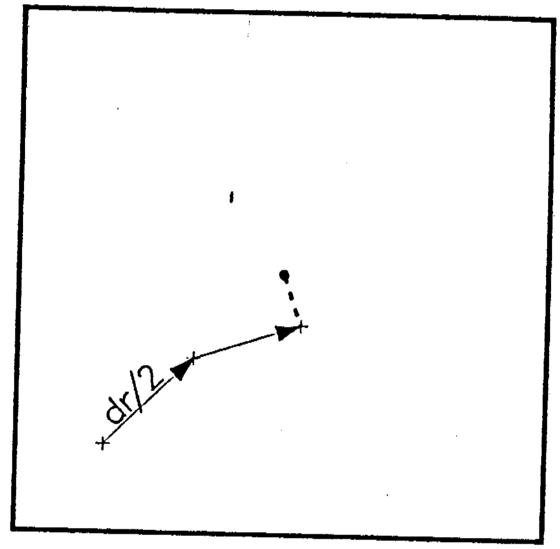

INSET
4.5X
LEGEND
WELL
ULTIMATE CAPTURE ZONE
1000
750
500
250
0
−250
−500
−750
y
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x


INSET
4.5X
LEGEND
WELL
ULTIMATE CAPTURE ZONE
1000
750
500
250
0
−250
−500
−750
y
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x

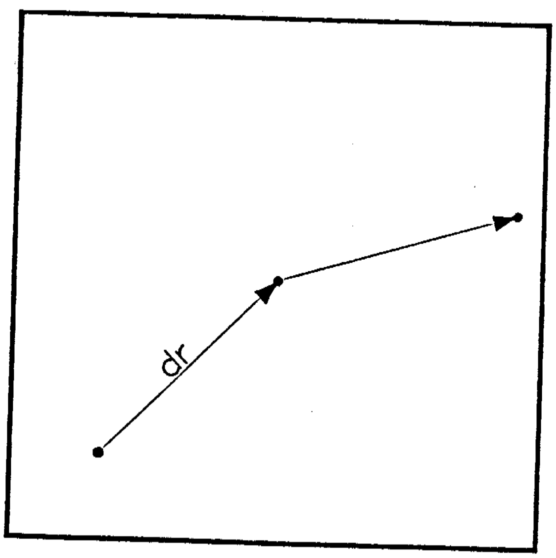
A : INITIAL STEP= dr



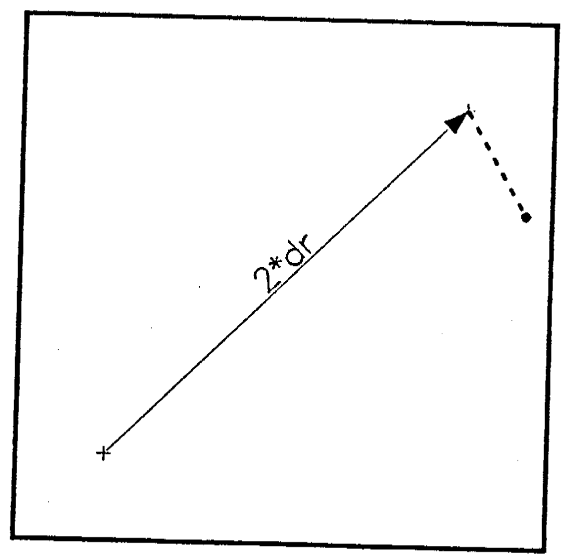

B : MINIMIZE STEP




C : IF STEP MINIMIZED, MOVE PARTICLE




D : MAXIMIZE STEP



precise simulation, often more efficiently.

Other refinements in Lee's algorithm include; adding a subroutine that can access any number of analytical solutions depending on which boundary conditions/analytical solution is desired (this allows one program to drive any number of different analytical solutions, remember the program is independant of the method used to generate the velocity field), a subroutine to calculate transient parameters as a function of the current time (explained later), and a subroutine that generates time dependant output (efficient when studying equal time front contours). All the subroutines that are implemented in this paper are included in appendix A. The updated flow chart follows:

A.) Input parameters

  1.) pumping rate, ambient flow rate, depth, and porosity

  2.) forward or reverse tracking flag

  3.) length constant

  4.) input time, maximum time to track

B.) Subroutine Taray - puts current position into array form

C.) Subroutine Tvari - decides which transient varible(s) to use

  1.) subroutines calculate the current transient variable(s) value using the current time

D.) Subroutine Tracker - determines appropriate length constant

E.) Subroutine Velocs - decides which analytical solution to use

  1.) subroutines calculate the current velocities for appropriate analytical solution

F.) Subroutine Tmove - trajects particle to next position using length constant and calculated velocities

  1.) calculates time step and updates the current time clock

G.) Subroutine Timary - plots the transient variables as afunction of time

H.) Subroutine Tbound - extracts particles near boundaries/well

1.) stop tracking if sufficiently close to boundary/well or if time has expired

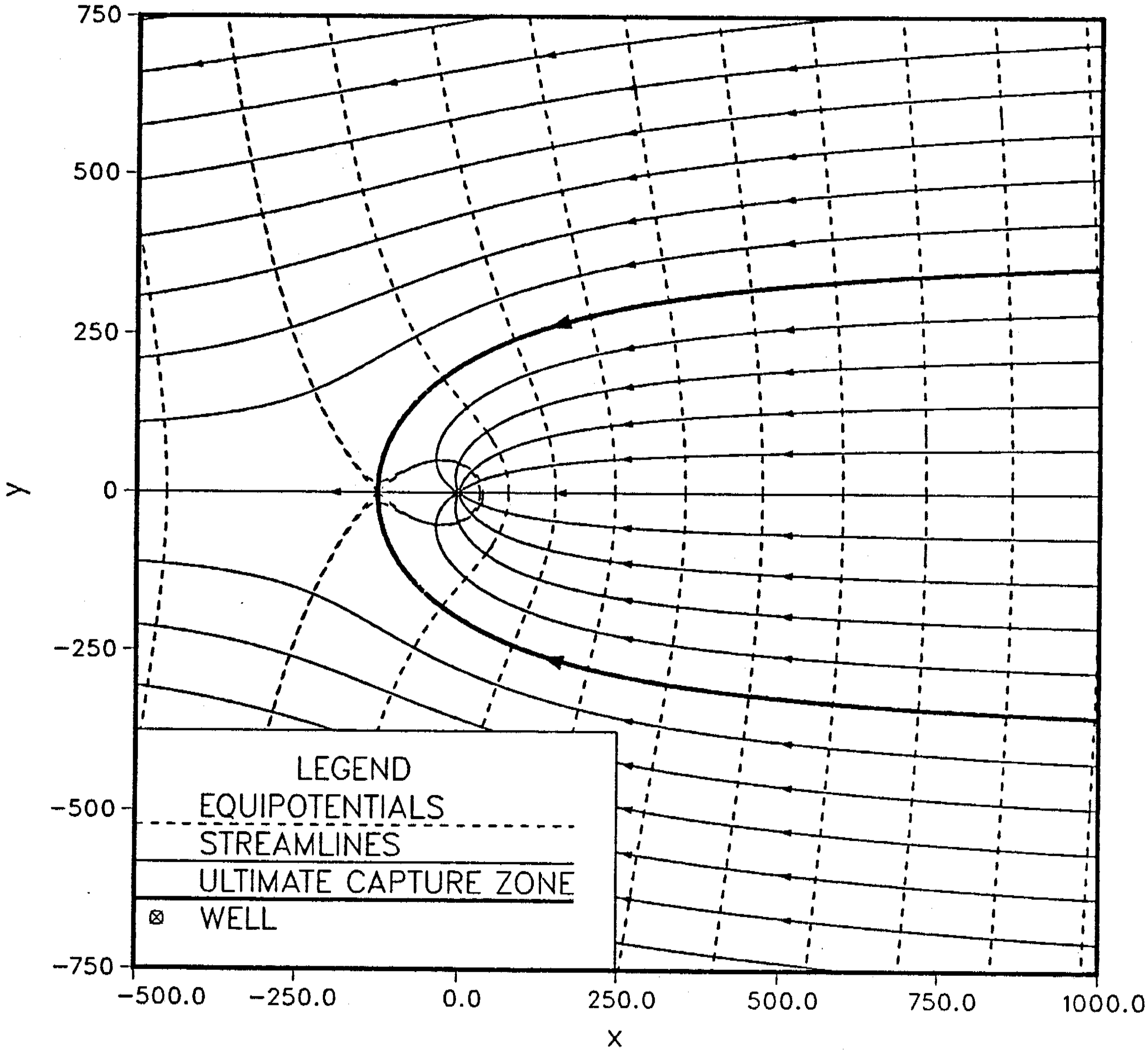
I.) Go to B

J.) Subroutine Twrite - write the appropriate arrays to output files

The particle tracking method implemented in this paper is a powerful tool for delineating capture zones in both transient and steady state. The precision, versatility, and reduced computer costs make this approach very useful in studying fliud movement for fairly simple conceptualizations. An example of the capabilities of the particle tracking algorithm is presented in figure (5). Both the equipotentials and the streamlines are generated from the particle tracking scheme. The equipotentials are generated simply by tracking particles perpendicular to the streamlines. This is acheived by substituting $v_y$ for $v_x$ and $-v_x$ for $v_y$ (from the theory that $\frac{\partial\Phi}{\partial x} = \frac{\partial\Psi}{\partial y}$ and $\frac{\partial\Phi}{\partial y} = -\frac{\partial\Psi}{\partial x}$; Newsom and Wilson, 1988). The method can be utilized as an approach to particle tracking within nodes of a finite element approach, for example to solve heterogenous conceptualizations.

**Dynamic Front Tracking (after Scafer-Perini et al, 1990)**

The dynamic front tracking method is the key to precise front tracking. Tracking fronts backward for a pumping well defines the time dependent capture zone. Tracking fronts that occur in a divergant flow field can create many difficulties, primarily filling the area near the stagnation point. (When back tracking towards the head of the front (points nearset the stagnation point) at intermediate times, when back tracking, the particles which previously had dominant velocities in the negative x-direction ($-v_x$) begin to be $v_y$ dominated.) This quickly evolves to a primarily positive x-direction velocity. The process is effectively a reversal of particle movement in a short period of time. This divergence creates relatively large void sections in the fronts when attempting to use the position of these particles at various times to represent equal time contours. Divergence significantly reduces the precision of the front realizations. Futhuremore, the longer the fronts are tracked the greater the effects of divergence. One method of minimizing these processes is to increase the number particles. If the particles are spaced evenly around the well the extra particles on the opposite side from the stagnation point are

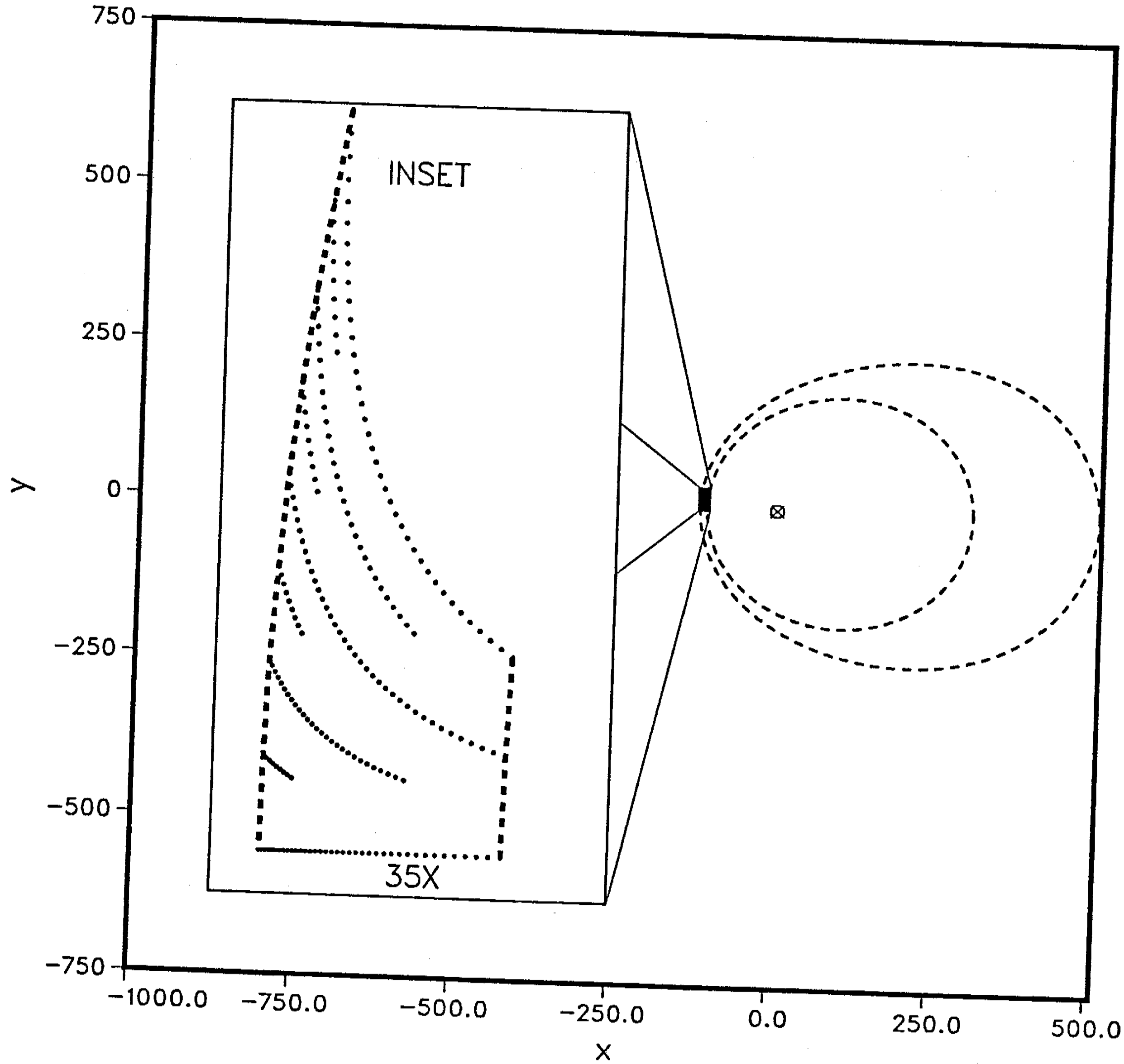


750
500
250
0
-250
-500
-750
y
-500.0
-250.0
0.0
250.0
500.0
750.0
1000.0
x
LEGEND
EQUIPOTENTIALS
STREAMLINES
ULTIMATE CAPTURE ZONE
WELL

unnessicary and effectively waste computer time. If the number of particles are increased on the stagnation side of the well the divergence can be reduced, but it is still dependant on the total time the particles are to be tracked. When the total tracking time is very large this approach becomes increasingly cumbersome and inefficient.

Scafer-Perini recently developed algorithms to solve the divergant flow problem. The methods are actually simple, but very effective. Shafer-Perini et al, 1990 describe two methods to numerically acheive the filled fronts. The algorithms are based on the premise that if particles are added when adjacent particles have seperated the divergant portion of the front will remain filled. One method they develope is the bumped array method (decribed below). The other is the linked list approach. The linked list method is similar to the bumped array method but uses two arrays to define a front. One array contains the points defining the current position of the front. The other defines in what order to move the particles to create a smooth front (pointer array). The primary difference is the bumped array method is storage efficient, the linked list method is CPU efficient. This paper implements a modified version of the bumped array algorithm. It would be a simple task to convert the approach applied to the link list routine (see Scafer-Perini, 1989).

The code that was developed for this research uses the particle tracker as the driving force for the dynamic front tracker. Dynamic front tracking manipulates the particle tracker to produce eqaul time front contours. This is achieved by moving the initial particles a small distance backward using the particle tracker, checking the separation distance between each particle, adding particles where the separation distance is greater than the desired tolerance, adding the number of new particles to the existing number of particles (bumping the array back for each particle) and repeat the process for the next time step. Figure (6) graphically demontrates this process. In the inset there are two time fronts represented by the dashed lines, the particles for each time step are represent by the dots. The section of the first front shown is represented by three particles at the same point in time. At a later time (not connected into a front), when the separation distance between the particles exceeds the limit, five particles would define the front. The second front shown is defined by nine particles. The inserted parti-


INSET
35X
750
500
250
0
−250
−500
−750
y
−1000.0
−750.0
−500.0
−250.0
0.0
250.0
500.0
x

cles are place at the midpoint of the separation distance during the current time step. This can be a poor approximation if the time between c hecks is too large or if the separation distance is too large. A flow chart of the algorithm follows:

A.) Input parameters

   1.) initial number of points, the initial points

   2.) forward or reverse tracking flag

   3.) seperation distance tolerance, checking time step

   4.) input time, maximum time to track

B.) Move the particles 1 time step

   1.) use the particle tracker with the input variables it needs

C.) Check the separartion distance between particles

   1.) if the seperation distance is larger than the tolerance, input a particle at the midpoint of the separation distance

   2.) update the array that contains the current position of the particles for each addition

D.) Output to an output file if it is a time of interest

E.) If current time is less than total time allowed go to B

This method produces precise front postions if the separation limit and the time step check are sufficiently small (Scafer-Perini et al, 1990). Development of this front tracking algorithm makes efficient analysis of fronts in time feasible.

## STEADY STATE CAPTURE ZONES

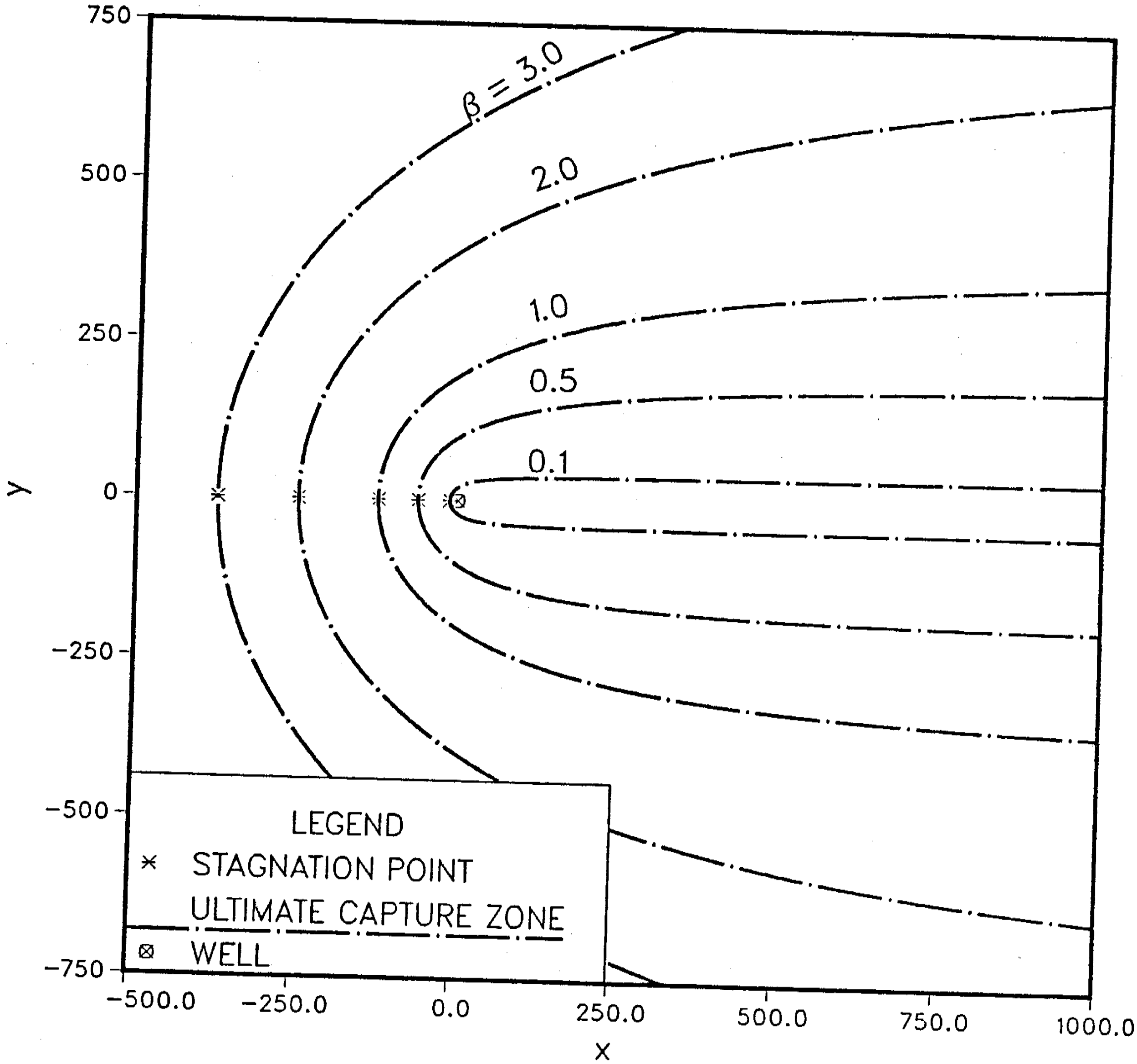
This section is provided as a very brief overview of steady state capture zones as they relate to the work presented here. There has been considerable work done on 2-dimensional steady state capture zones, much of it has been done by private consulting firms for specific problems. As mention previously Lee (1986) presented capture zones for a number of boundary conditions. Newsom and Wilson (1988) and recently Huyakorn and Blanford have made additional contributions. Wilson, Linderfelt, and the author have presented two talks recently on various aspects of steady state capture zones (Wilson, and Linderfelt, 1988; Linderfelt et al, 1989). Javendel and Tsang (1986), Bear, 1972, and many others have published work that is directly related to infinite flow field capture zones.

Steady state ultimate capture zones are defined primarily by stagnation points. An ultimate capture zone (UCZ) is actually a simulation of the groundwater divide (stagnation streamline) created by a pumping influence. They are generated (at least by our research group) by back tracking a particle from the stagnation point through the flow field. The position of the stagnation point and ultimately the groundwater divide depends primarily on the magnitude of the pumping rate and the magnitude and direction of ambient flow. Dimensionless parameters are often used to describe the ratio of these controlling influences. Javendel and Tsang applied the dimensionaless pumping coefficient $\beta$ to be the ratio of the pumping rate over the ambient flow rate times the depth of the aquifer.

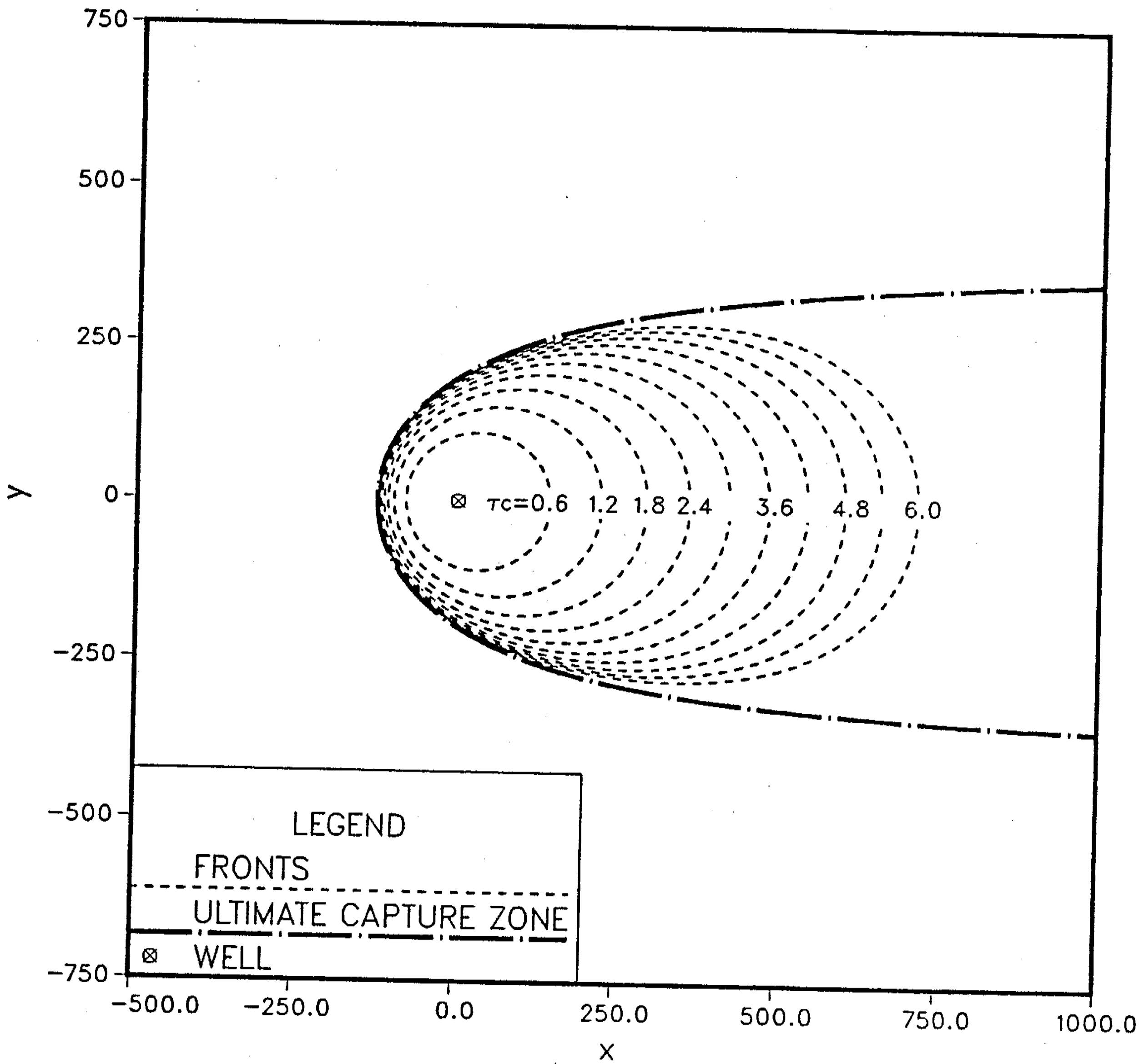
$$\beta = \frac{Q_w}{q_a\, b}$$

Figure (7) illustrates how the capture zone increases with increasing $\beta$.

Time dependent capture zones are simply the position of the front at a given time. These are generated using a dynamic front tracker like the one presented previously. Figure (8) illustrates how the time dependent capture zone evolves through time relative to the ultimate capture zone (chain-dotted line) or the time dependent capture zone at infinity. The dimensionaless time coefficient ($\tau_c$) is used here to demonstrate relative time fronts. The is given by the equation:




define β in figure caption


750
500
250
0
−250
−500
−750
y
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x
τc=0.6
1.2
1.8
2.4
3.6
4.8
6.0
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL

β=1.

$$\tau_c = \frac{\pi\, q_a\, t\, \beta}{b^2\, n}$$ (16) (modified from Lee, 1986)

The $\beta$ is the dimensionaless parameter from above, n is porosity, and b is aquifer thickness. Lee's parameter was:

$$\tau = \frac{\pi\, q_a\, t\, \beta}{d\, b\, n}$$ (17)

where d is the distance between the pumping well and the stream. Obviously there is no stream in the domain utilized in this paper so this parameter is substituted an additional b for d.

## CAPTURE ZONES IN TRANSIENT FLOWS

The approach to transients taken in this paper is one that assumes an instantaneous response in the variations in pumping or recharge. Storage is neglected. In effect the flow field is assumed to be quasi-steady state. Transient parameters are handled by coupling another subroutine to the particle tracker (see numerical methods section). The subroutine determines the value of the transient parameter(s) at each time step, using the total time elapsed, and those values are then used to calculate the velocities. This magnifies the importance of the time step. If the transient parameters are a continuos function the time step must be small enough to simulate the time variation in the parameter. The simulations that follow show that the flow fields under the given conditions can be model with excellent precision.

When simulating a steady state system a flownet and the front positions give most of the information needed to understand the flow field. This is not the case when simulating a transient system. The dynamics of the flow field are more complex, the flownet loses its meaning, and the fronts can be intricate. Therefore, other methods of analysis must be utilized. Front tracking and pathline generation continue to be the primary tools. These tools are applied to generate diagnostic graphs and parameters that help in determining the origin of pumped waters, and the path these waters take to reach the well. Certain terms are used to aid in the explanation of the dynamics of the transient capture zones during transient flows. They are defined as follows:

passed zone - the zone between the 100% and the 0% boundaries. The volume of fluid within this zone has a chance less than 100% and greater than 0% of being captured by the extraction well (time independent).

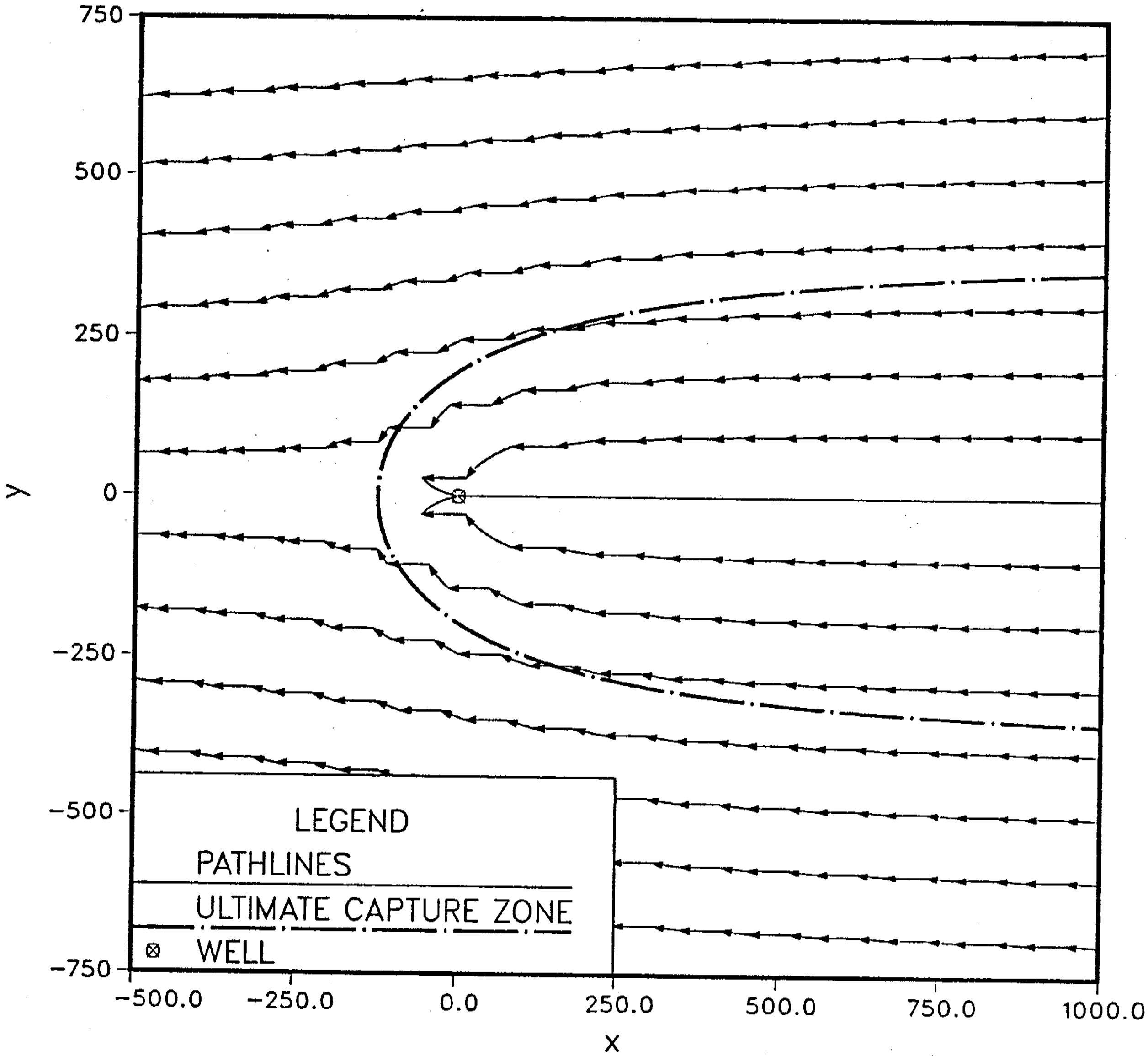
100% boundary - the maximum extent of the all the area that will always contribute to pumping well given enough time (time independent).

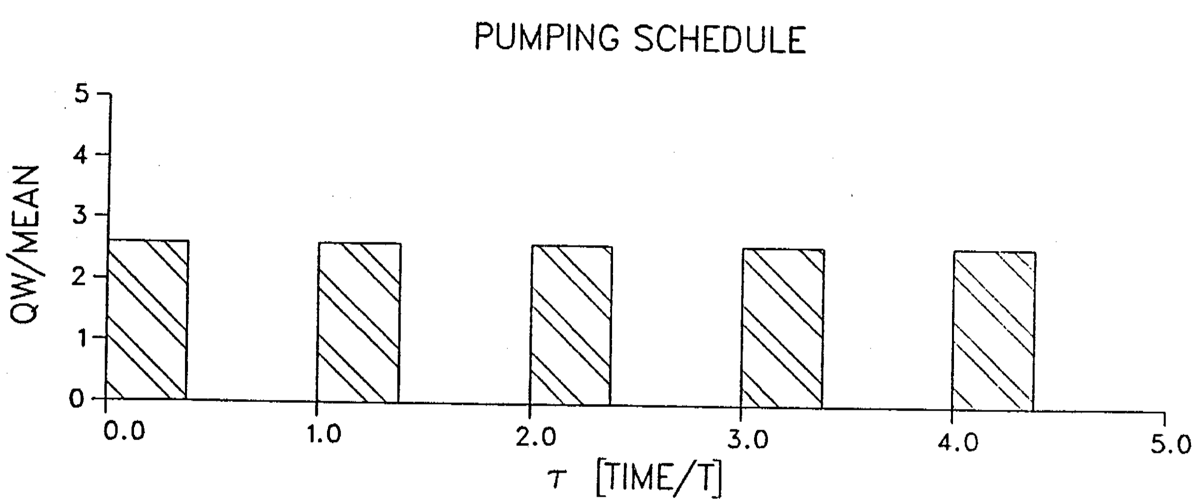
0% boundary - bounds the area that at some time the particles at those coordinates will enter the well (time independent). Beyond this boundary no water will ever contribute to the well.

period capture zone (PCZ) - time dependent extent of all the particles that will enter the well during a single periodic pumping or recharge period.

effective stagnation point - the maximum down-gradient position that can contribute to the well a periodic transient continues to infinity (time independent).

An arbitrary example is presented here to aid in demonstrating the approach taken to evaluate transient capture zones. The simulations presented here are purely hypothetical, no simulations have been modeled using real data. The input data is arbitrarily chosen to magnify the flow dynamics. They are all realistic values. The first example presented is used to compare with all subsequent simulations. The variables in the analysis are pumping rate $(Q_w)[\frac{l^3}{t}]$, ambient flow rate $(q_a)[\frac{l^2}{t}]$, true time (t)[t], period (P)[t], time pumped per period $(t_P)$[t], and percent time pumped per period (% on) [-] also of interest is the arithmetic mean pumping pumping per period $(\overline{Q_w})[\frac{l^3}{t}]$. Other parameters that will be considered constant throughout all simulations are; porosity n=0.1 [-] and depth of aquifer b=785 [l]. To make it easier to work with vary large times, from here on dimensionless time units $(t_c)$ will be used, defined by equation 16 (see steady state section). A problem arises when using the steady state dimensionless time parameter in transient systems. This problem is due to the fact that if a parameter used to define the coefficient changes during a flow period, so does the coefficient. Therefore, this coefficient is no longer only a function of time but also a function of the transient parameters. Consequently, the dimensionless time does not reflect a relative time. This problem is circumvented by defining the parameters utilized in equation 16 as values of those parameters at t=0. For the initial case the pumping rate is 0.157, the ambient flow rate is 0.0002, the period is $1.3t_c$ , pumping time is $.5t_c$ and the % on is 38. The wave form is a square wave with the minimum pumping rate equal to 0.0. This wave form exaggerates the flow effects of variable pumping, making it easier to analyze the dynamics of transient pumping schedules. In figure (9) the pumping schedule for the first simulation is shown, the x-axis $(\tau)$ is real time normalized by the period for this case which from now on will be noted as T. T remains constant for all simulations $(T=1.3t_c)$, allowing for easier comparison between different simulations. For this case the y-axis is the maximum pumping rate normalized by the mean

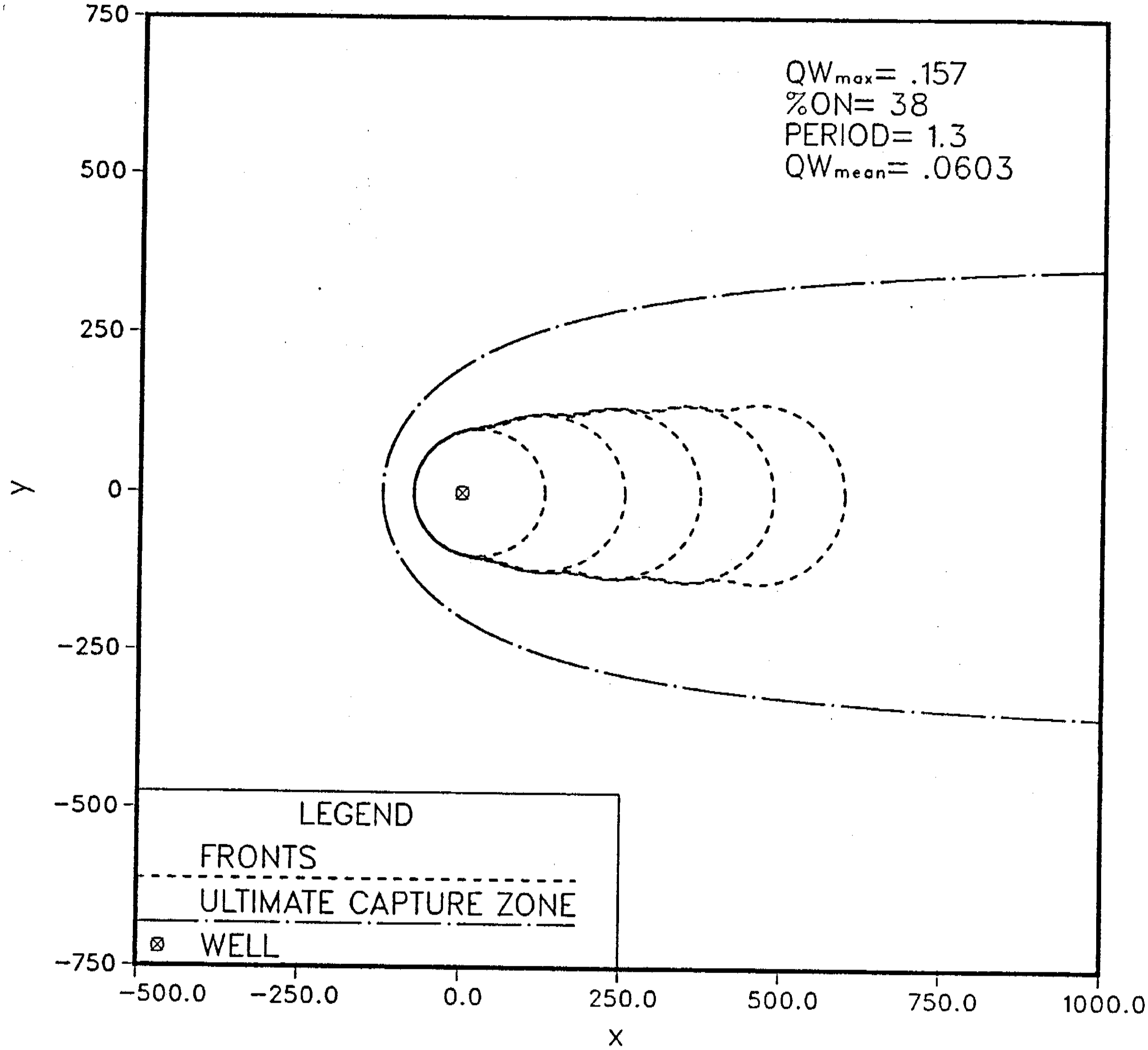

750
500
250
0
-250
-500
-750
y
-500.0
-250.0
0.0
250.0
500.0
750.0
1000.0
x
LEGEND
PATHLINES
ULTIMATE CAPTURE ZONE
WELL


PUMPING SCHEDULE
QW/MEAN
0
1
2
3
4
5
0.0
1.0
2.0
3.0
4.0
5.0
τ [TIME/T]

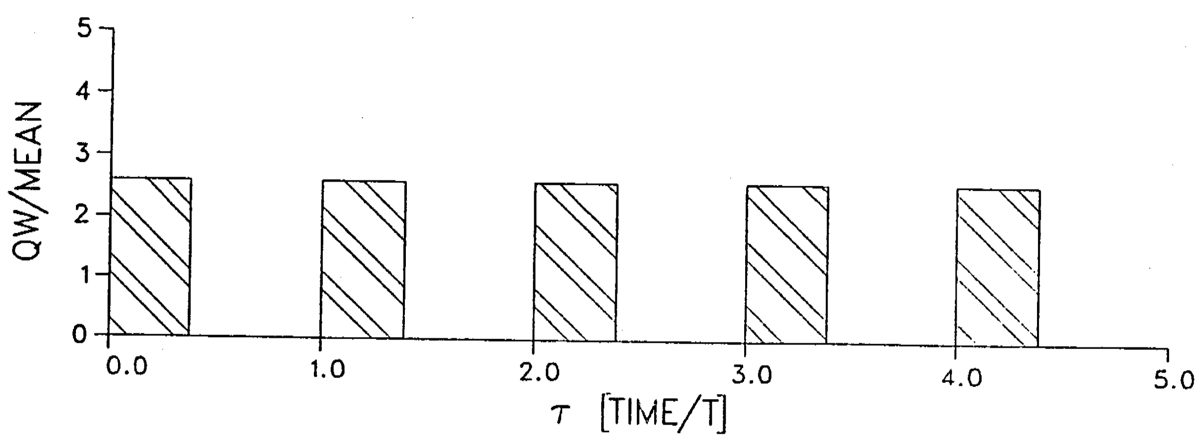
(0.0603), again the normalization factor remains constant throughout all simulations. For the purpose of this paper the values of the input variables are not very important and they are listed for reproducibility only. The primary focus of these simulations is to evaluate the origin of waters entering the well, their flow paths, and the dynamics associated with the ambient flow and pumping influences.

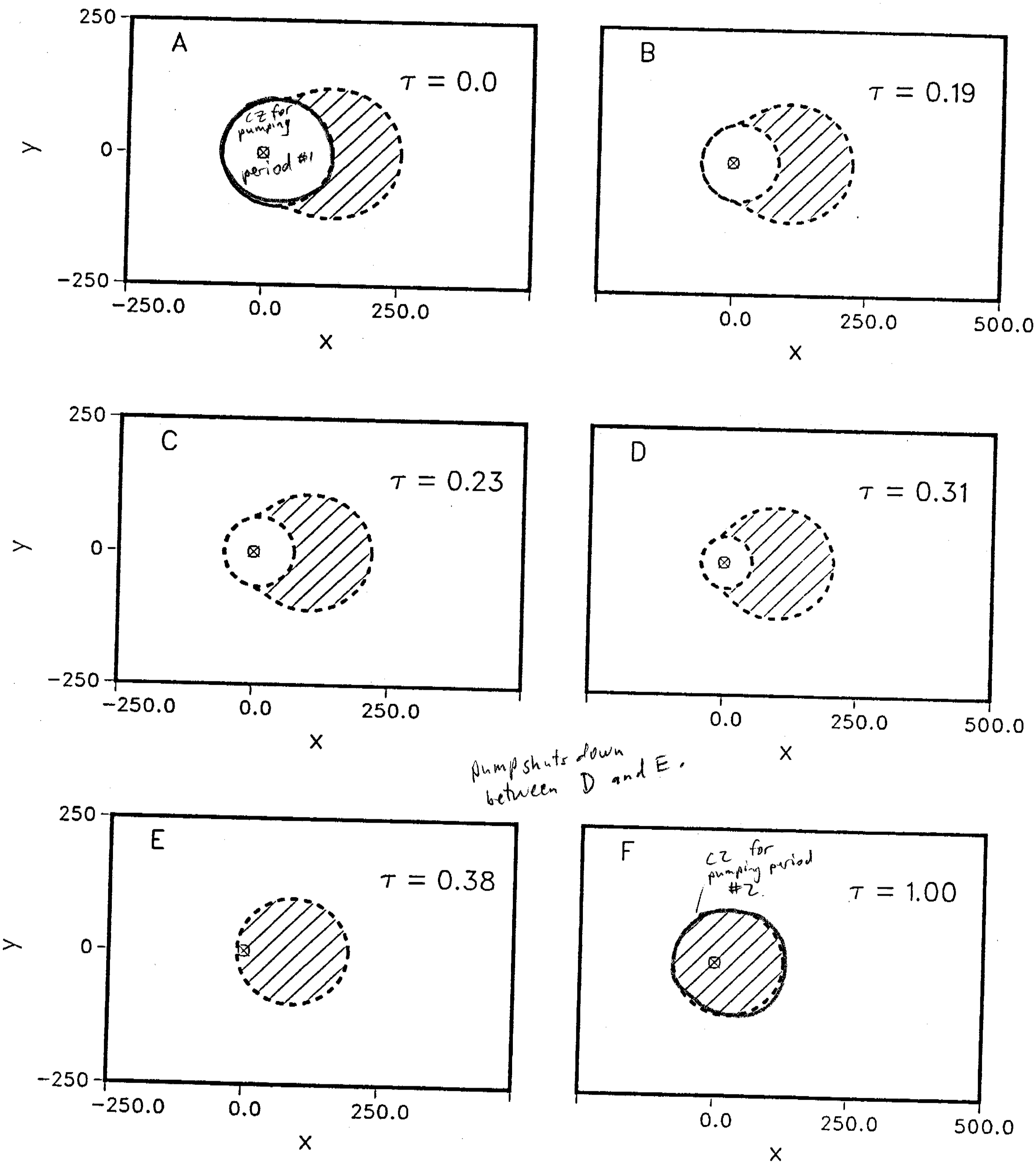
Figure (9) demonstrates how cyclic pumping effects the flow paths of 15 particles entering the flow domain. The chain-dot curve is the ultimate capture zone for the maximum pumping rate. It is obvious that this is not the capture zone under these transient conditions. A large portion of the particles that would normally be captured under steady-state pumping conditions actually pass the influence of the well. The arrow heads designate the position of the particles when there is a change in pumping rate. During pumping the paths are curved towards the well, during ambient flow periods the paths are parallel to the ambient flow direction. For the paths during ambient flow periods the length is constant for all particles and is exactly equal to the ambient flow rate times the time the pump is off, this value is called the ambient flow length ($l_a$). The distance the particle travels during pumping time is dependent on its position in the flow field. This figure demonstrates that the maximum pumping rate ultimate capture zone does a very poor job simulating the true capture zone.

Well-head protection is concerned with determining what is the distribution of particles that will eventually enter the well. For a cyclic pumping schedule this can be refined further so that the particles that enter the well at each pumping cycle is defined at some time. Figure (10) is a 'snapshot' at t = 0 that shows the distribution of the particles that will enter the the well during each pumping cycle (PCZ). The first front defines the limit of capture by the first cycle, the difference between the second front and the first encompasses all the particles that is captured by the second period and so-on. These fronts are defined by injecting particles at the well, back tracking, and plotting the front positions at the instant when the pump is turned off. To demonstrate the validity of this approach, the first two period capture zones are forward tracked illustrating their evolution through time (figure 11). In figure (11A) the two PCZs are shown at t=0 with the shaded area defining the second period capture zone. The volume of water defined by the first PCZ can be shown to be captured by the well during the first cycle




PUMPING SCHEDULE





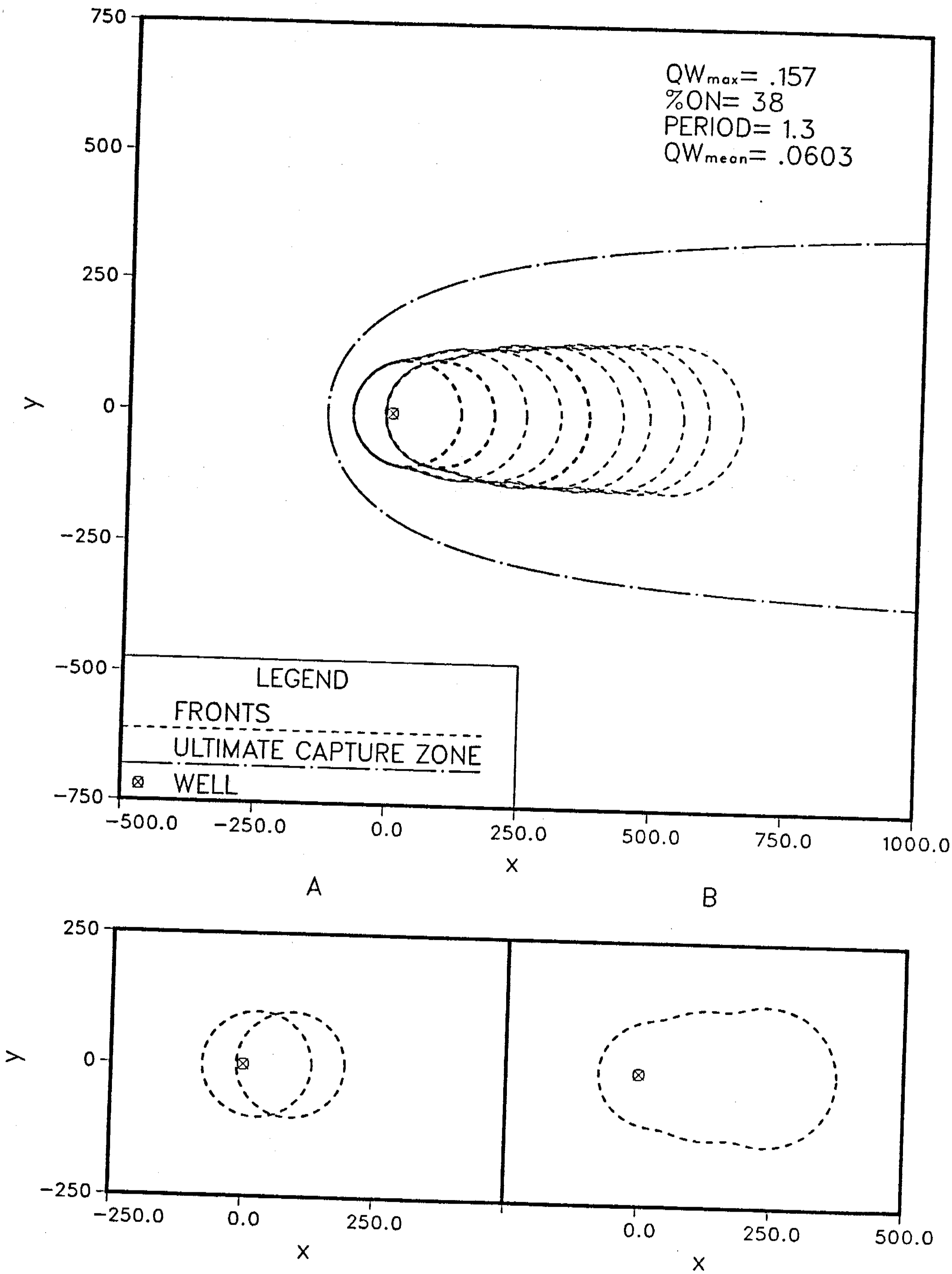
A
τ = 0.0
CZ for pumping period #1
B
τ = 0.19
C
τ = 0.23
D
τ = 0.31
Pump shuts down between D and E.
E
τ = 0.38
F
CZ for pumping period #2
τ = 1.00
y
x
250
0
−250
−250.0
0.0
250.0
500.0

using a constant pumping rate for a time equal to the $t_P$, this zone is definitely the volume of water captured during the first pumping period. The question is whether the zone defined by the second PCZ is truely all the water pumped by the second period. At the time $\tau = .19$ (figure 11B) the area of the first zone has been significantly reduced simulating extraction of water while zone two has changed its shape, converging on the well. This trend continues through figures (11C) and (11D). In figure (11E) $\tau = .38$, the pumping time for the first period ceases. The first zone has completely disappeared, simulating the completion of the extraction of the first pumping cycle. The second is completely closed. For the rest of the period there is only ambient flow, no extraction. In figure (11F) the period is over and the pump is just about to be turned on again. The shaded zone now occupies the exact zone that the first zoned occupied at $\tau = 0.0$. This is very good evidence that endeed the zones defined in figure (10) are the extent of the waters captured during specific pumping periods.

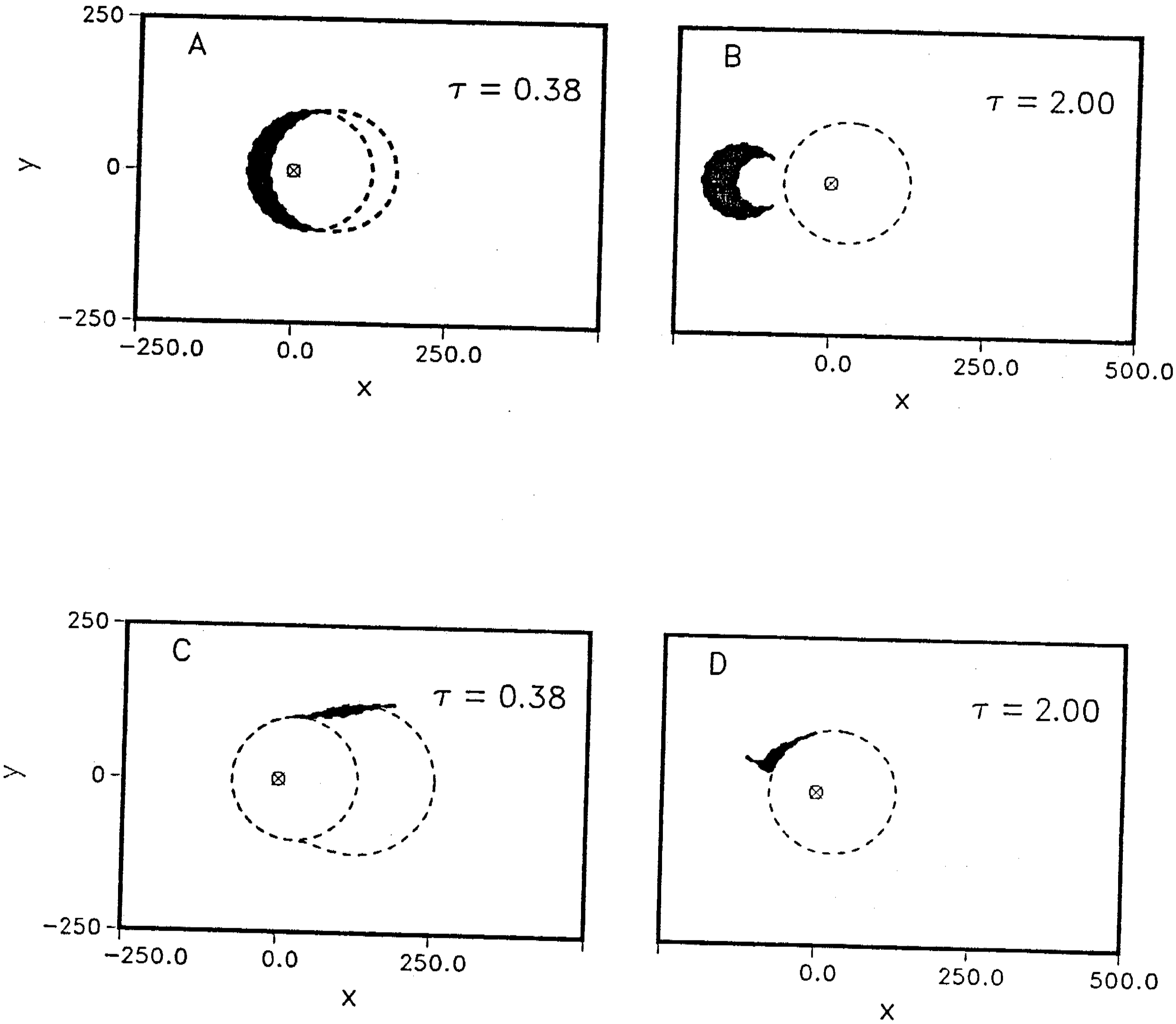
The front positions in F figure(10) ignore the dynamics of the flow field within the period. The paths of particles during the period could reveal more specific details of the system. Since this simulation is a square wave, examining the positions of the front just before the pump is turned on (at the end of the cycle) yields the supplementary information needed to understand the dynamics of this system. This is due to the fact that during ambient flow periods the size and shape of the fronts do not change they simply translate ~~relative to the~~ in the direction of ambient flow ~~direction~~. Figure (12) shows the positions of the front at the time the pump is turned off and just before it is turned on again. Inset A demonstrates how the front translates during the first ambient flow period. The shape of the front at some later time is shown in inset B. This pattern will be significant in subsequent sections. The maximum pumping rate ultimate capture zone is again shown to ~~give relative~~ for reference. Two concepts? points are evident when examining the way the fronts move through time. First, the position of a front mimics the front that occurred exactly one period before, except it is extended toward the ambient flow source by ~~the~~ an area that is a function of

? the volume of water extracted per cycle. The longitudinal ? lateral growth is usually significantly larger than the

?? transverse growth. It is also apparent that three zones have been created, the boundaries of these zones have exactly the same shape, ? one has simply been translated a distance equal to the ambient flow length. The first zone is the passed zone, that is the fluid in this zone will only enter the well at certain


QWmax= .157
%ON= 38
PERIOD= 1.3
QWmean= .0603
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL
750
500
250
0
-250
-500
-750
y
-500.0
-250.0
0.0
250.0
500.0
750.0
1000.0
x
A
B
250
0
-250
y
-250.0
0.0
250.0
x
0.0
250.0
500.0
x

G87



A
$\tau = 0.38$
B
$\tau = 2.00$
C
$\tau = 0.38$
D
$\tau = 2.00$
250
0
−250
y
−250.0
0.0
250.0
500.0
x

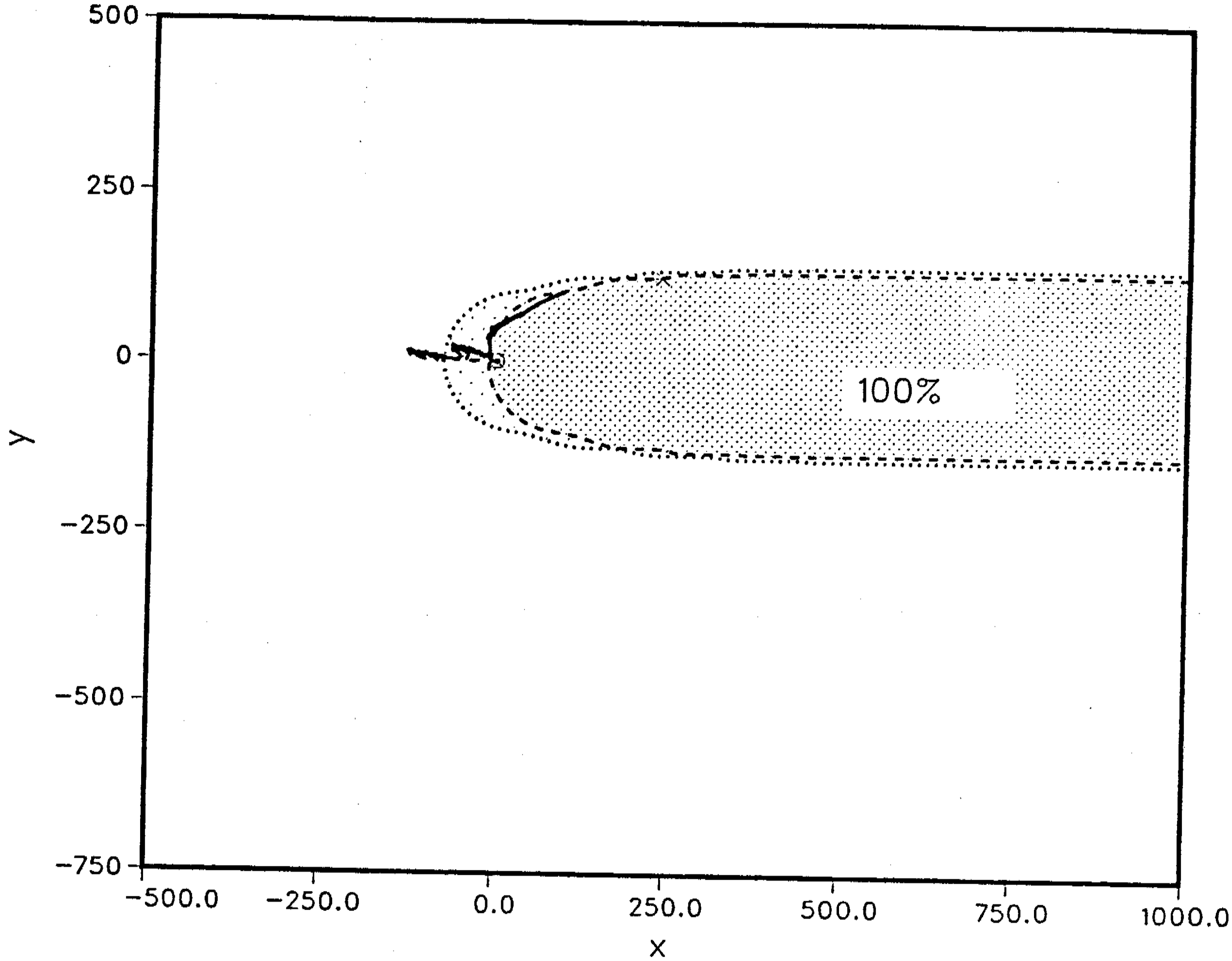
times depending whether the pump is inducing flow. The second, bounded by the inner group of fronts, is the 100% zone or the zone that the always contributes to the extracted water. And finally, the zone outside the ~~100%~~ 0% boundary that never contributes to produced water.

Figure (13) depicts how the passed zone functions. In part A, $\tau$ = .38 (the well has just been turned off). The shaded area is located within the passed zone. If the shaded area is tracked forward, figure B shows the position just before the third pumping cycle. Clearly, this volume of water has passed the well since it lies outside of the period capture zone for the next cycle. In part C a shaded area is created within the wing of the passed zone, the outline of the second period capture zone is shown to illustrate that this shaded area is encompassed by the PCZ. The particles in the shaded area are tracked forward. Again, the particles are clearly outside the zone that ~~with~~ will be captured by the next pumping cycle (figure 13D). For this case, it is obvious that the transient pumping generates an area that has a time dependent probability of being captured between 0 and 100%. These zones cannot be considered capture zones in the same sense as the capture zones derived for steady state pumping scenarios. This creates a dilemma on just how to treat these areas in a well head protection scheme.

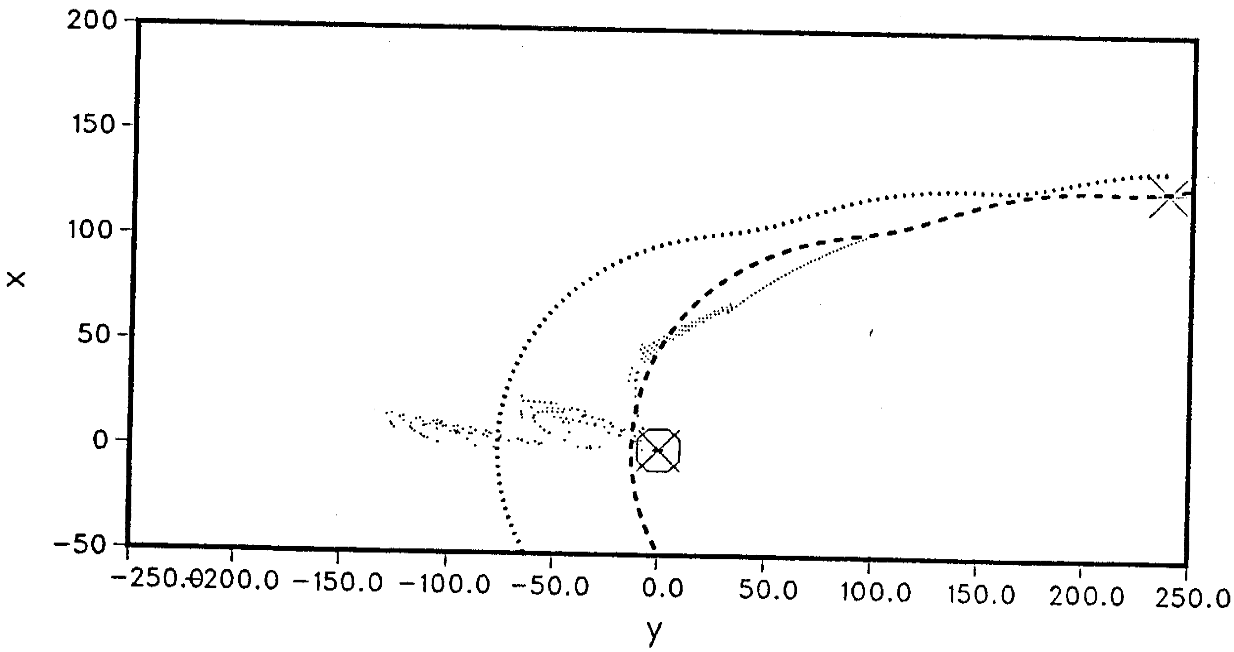
The passed zone for this example is defined by the 100% and 0% boundaries (figure 14). Both of these boundaries are considered the limits at t=∞. For this case the 100% capture boundary is essentially the set of fronts that occur at the end of the ambient flow portion of the period (when back tracking).

wtf ? The front has to be adjusted slightly to compensate for the front positions that are not shown. The boundary that separates the water that will never be captured from the ~~the~~ water that has some probability to be captured, the 0% probability boundary, is the set of fronts that occur just after the pumping portion of the period. ~~Similar adjustments have to made.~~

Figure (14) demonstrates the possible extent of a contaminant plume due to seepage of a continuos contamination source located at the X mark as a result of transient pumping. The example 100% and 0% boundaries are shown with the denser shading being the 100% zone. The inset is magnified to illustrate the irregular nature of the plume. There were 238 particles released at even intervals over 2.38 periods. This is an instant in time realization just prior to the ambient flow portion of the period.


500
250
0
-250
-500
-750
y
100%
-500.0
-250.0
0.0
250.0
500.0
750.0
1000.0
x

INSET


200
150
100
50
0
-50
x
-250.0
-200.0
-150.0
-100.0
-50.0
0.0
50.0
100.0
150.0
200.0
250.0
y

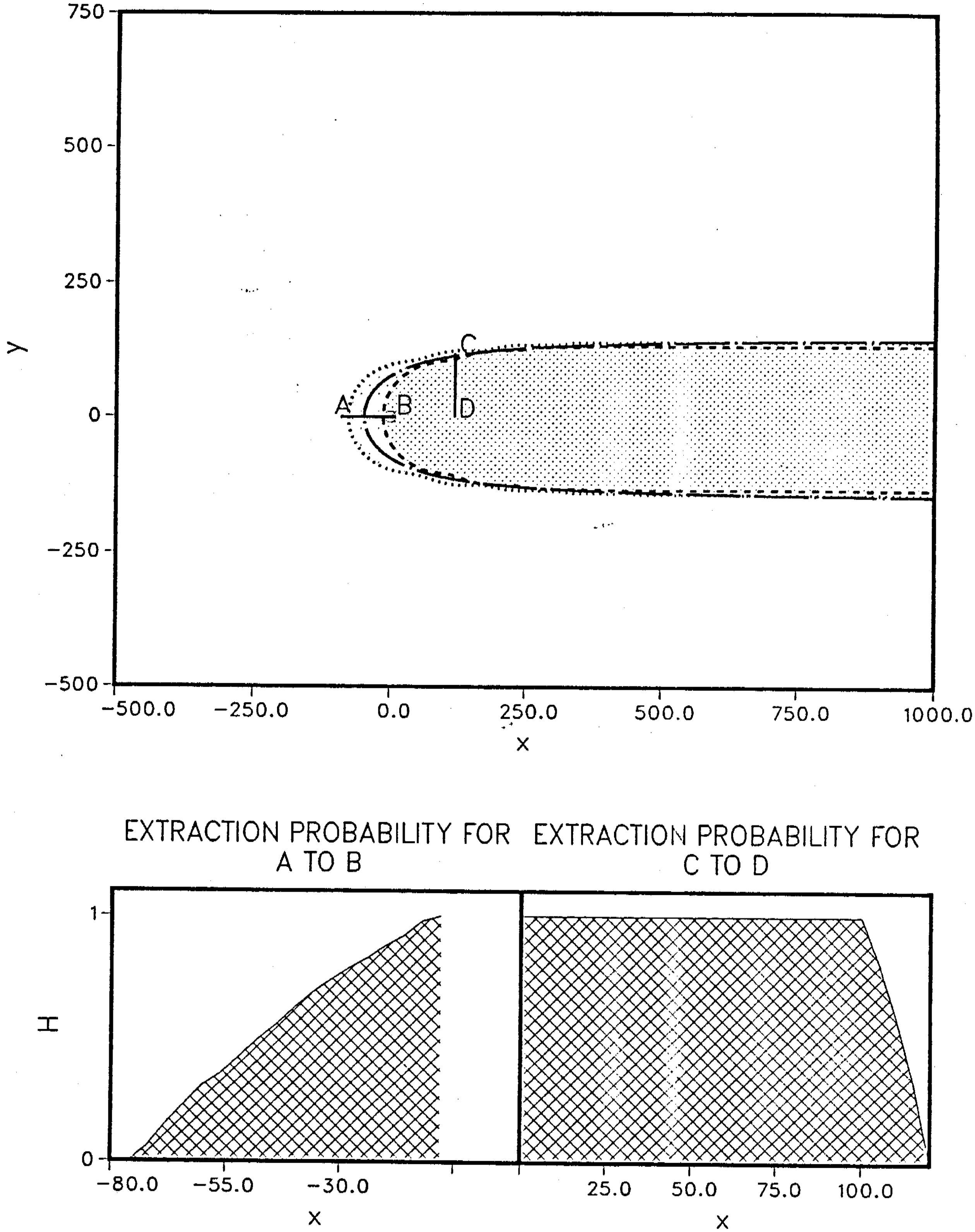
This figure demonstrates how transient parameters may spread a plume over a larger area than a steady state system would. This example neglects dispersion, if a random walk model was employed it would have even greater dispersive capability.

In many cases it may be important to make decisions from a probabilistic point of view. The 0 and 100% contours are defined by the position of fronts in time (above), but in some problems the distribution of probabilities within the passed zone may provide detail that could aid the decision making process. Particularly in contaminant studies the probability that a particle from a given position will reach the well could be a valuable piece of information.

Probabilities for cyclic transients can be determined in a simple, and inexpensive manner. This is based on the nature of the system. In a regular cyclic function the probability? value is only a function of its relative position within the period. Therefore, describing yuk the variation of a parameter within one period, describes that parameter for every period. This concept provides a way to determine the cumulative distribution function of a transient parameter with respect to time as it approaches $t=\infty$. Therefore, the probability that a particle, given enough time, will reach the well when the duration of the exposure is one period is the same as the probability using an infinite number of particles released over an infinite amount of time.

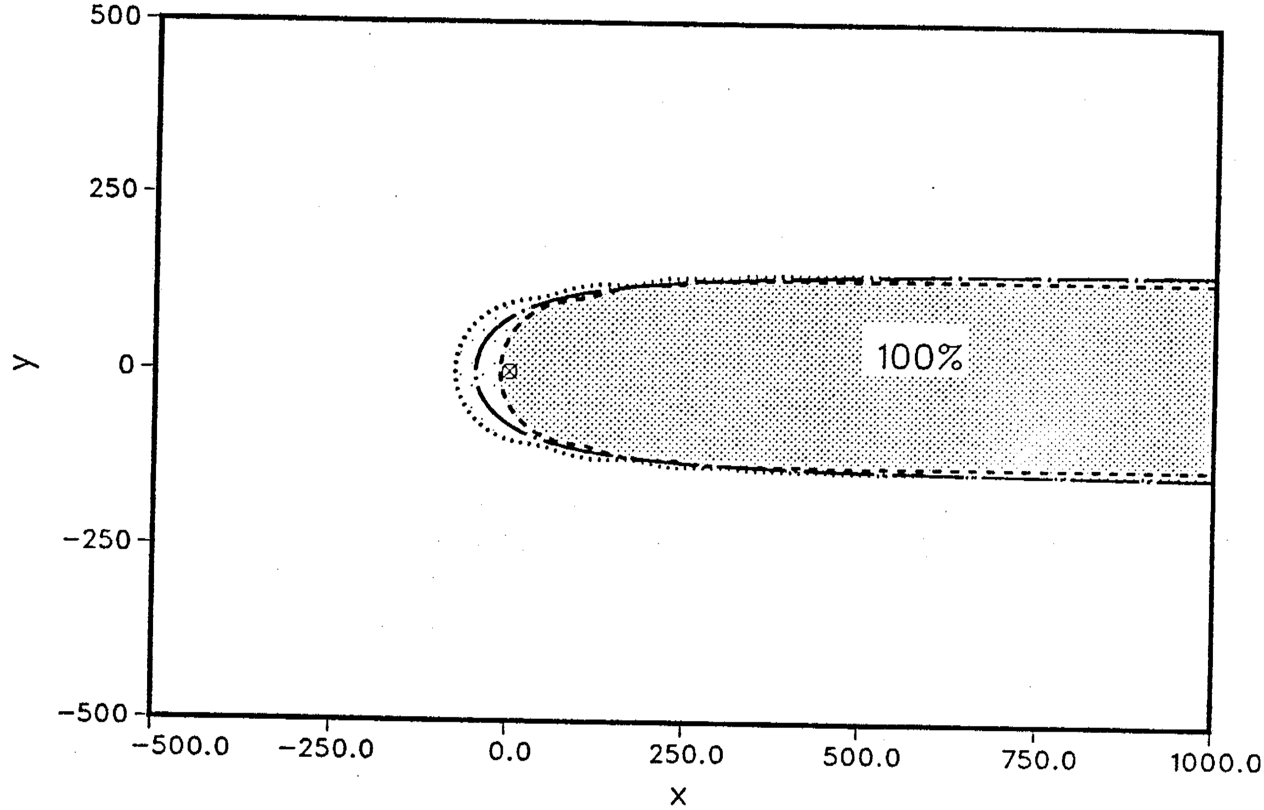
This concept is utilized to determine the probability that a particle released at a given point will enter the well given an infinite amount of time (extraction probability). The probabilities are determined by first dividing the period into $n_p$ equal intervals, where $n_p$ is the number of particles to be released. At $t=0$, the first particle is released into the transient flow field and particle tracked for a sufficient time to either reach the well and be extracted or to travel beyond the 0% boundary. If the particle reaches the well a counter is increased by one, if not the final location is saved in an array. The next particle is released at $t=\frac{P}{n_p}$ and it is tracked forward, and so on until every particle has been released. The probability (H) is calculated from the total number of particles that are extracted, $n_w$, divided by the total number of points,

What is P - Particle #? Period length?

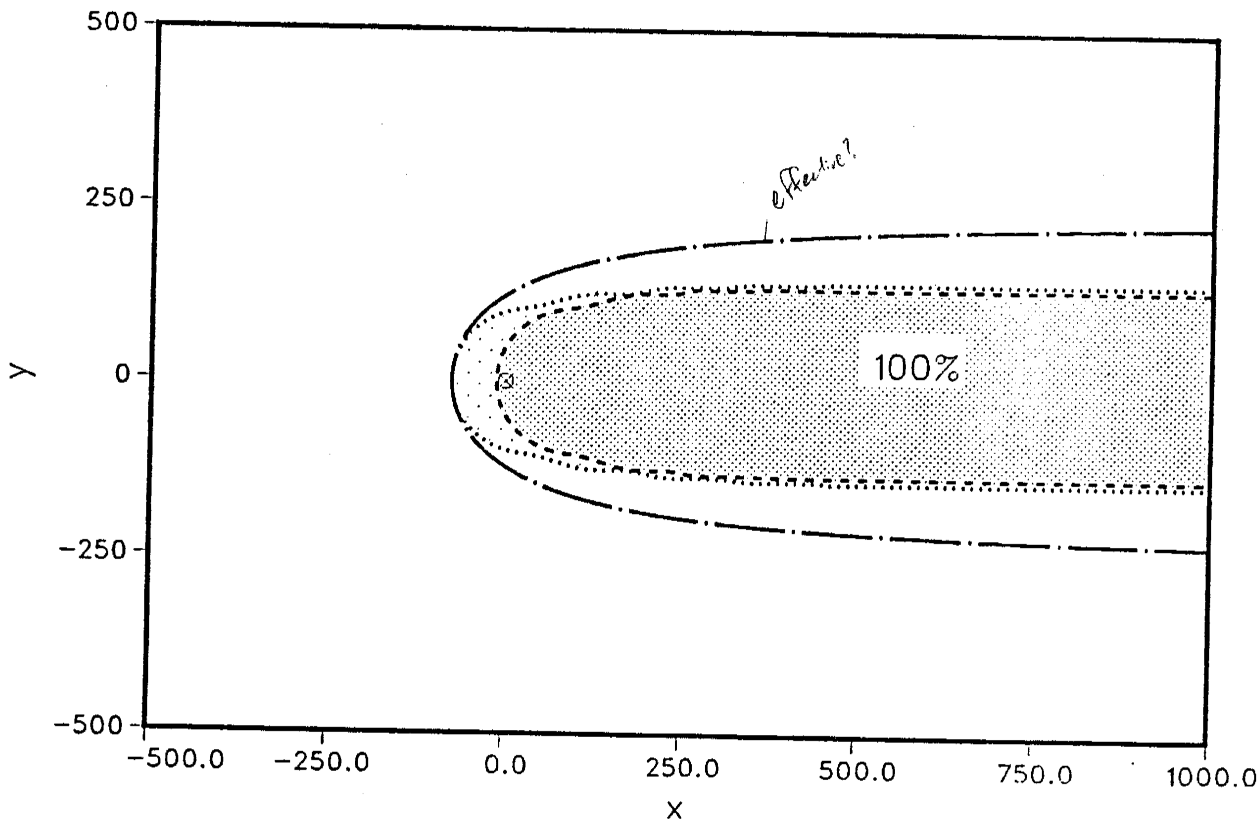


750
500
250
0
−250
−500
y
A
B
C
D
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x
EXTRACTION PROBABILITY FOR
A TO B
EXTRACTION PROBABILITY FOR
C TO D
1
0
H
−80.0
−55.0
−30.0
x
25.0
50.0
75.0
100.0
x

PROBABILITY ZONES WITH RESPECT
TO MEAN PUMPING RATE UCZ




PROBABILITY ZONES WITH RESPECT
TO EFFECTIVE STAGNATION POINT UCZ



$$H = \frac{n_w}{n_p}$$

This technique is very sensitive and has to be done very careully. As mentioned previously, the particle tracking method utilized in this paper works best when back tracking as complications may arise when forward tracking. These complications are compounded when considering that a particle may take a tortuous path to reach the well which may result in a much longer residence in the system than would normally be expected. Total particle tracking times must be extended to accommodate the increased distance the particle has to travel to reach the well. Particles trajected passed the well can give anomalous probabilities. When a particle is positioned at a distance to the well less than the length constant the particle may traject over the well, if the trajected position is greater than the extraction criterion, the particle may bounce back and forth indefinitely. Eliminating over shooting can be done by assigning a maximum value of the length constant to be less than the extraction criterion. If the extraction criterion (the distance the particle has to be from the well before it is considered to be extracted, halting the tracking routine) is too large the precision of the technique suffers, especially in transient systems where particles may track very near the well but never enter. The length constant has to be reduced to accommodate for these particles. Subsequently, the time step is decreased and the number of iterations to track the particle increases, increasing the time and cost. Therefore speed is sacrificed for precision or visa versus. For the purposes of this paper speed is sacrificed and precision is retained.

yuk!

yuk!

Extraction probabilities for points along two cross-sections located in the passed zone along with the 0% and 100% boundaries are presented in ~~f~~ Figure (~~17~~ 15??). The mean UCZ is included for comparison. not previously defined! Both cross-sections are oriented to delineate how the extraction probabilities vary within the passed zone. Probability contour graphs will not be presented due to cost and time limitations. Transect extraction probabilities will be presented for all major passed zones.

One goal of this study is to help determine whether or not the true capture zone for a transient scenario can be adequately approximated by any steady state ultimate capture zone. Previous figures demonstrate how poorly the maximum pumping rate ultimate capture zone simulate the true capture zone. The arithmetic mean pumping rate UCZ would be the obvious choice for an approximation, but as figure (16) illustrates the mean UCZ underestimates the extent of the true capture zone. For this case the effective stagnation point UCZ does the best job of simulating the true capture zone using a steady state approach. An effective stagnation point is not a stagnation point at all, the velocities are never zero there, it is actually the position directly down gradient from the well? (natural gradient) that the exerting forces in a cycle completely balance out. What? That is, if a particle begins a period at the effective stagnation point, during the pumping period it moves exactly the distance equal to the ambient flow length. At the beginning of the next period it will again be at the effective stagnation point. For this case the effective stagnation point lies near the front defining the period one PCZ ($stag_{eff}$ = negative? 75.7). The effective stagnation point UCZ is calculated by substituting the effective stagnation point into the equation to calculate the stagnation point (equation 14) and backing out the associated pumping rate and applying the particle tracking routine.

**Constant Pumped Volume Simulations**

When designing an extraction plan, usually the volume of water to be extracted and the time frame to accomplish the extraction is known, therefore in this engineering approach to the sensitivity analysis these are held constant (period (P)and the mean pumping rate $\overline{Q_w}$). Six simulations are presented in this section:

(1) 2 cyclic on and off simulation

(2) an irregular pumping rate simulation

(3) 2 cyclic reduced pumping rate simulation

(4) a sine wave simulation

Each simulation has 4 graphs associated with it, a period capture zone (PCZ) graph, a time front position graph, a probability graph, and a prediction graph all similar to the graphs presented above. The variables in this section are the amplitude ($Q_w$) and the time on ($t_P$). The value for these parameters are determined arbitrarily on the basis of illustrating the sensitivity of the capture zone to the transient flow field. The conclusions drawn from these simulations is presented in the preceding section. This section only presents the results of the simulations.

? 17a-d, 18a-d, 19a-d (?)

The first set of graphs (figures 17, 18, 19) have cyclic square wave time functions that have different maximum pumping rates but all have minimum pumping rate of zero. This illustrates one extreme of the possible realm of variations, the other extreme would be if the ambient flow rate would be zero for a portion of the period. The values that are being used for the pumping rate are generated by deciding what ratio of on versus off time that is appropriate and using this equation:

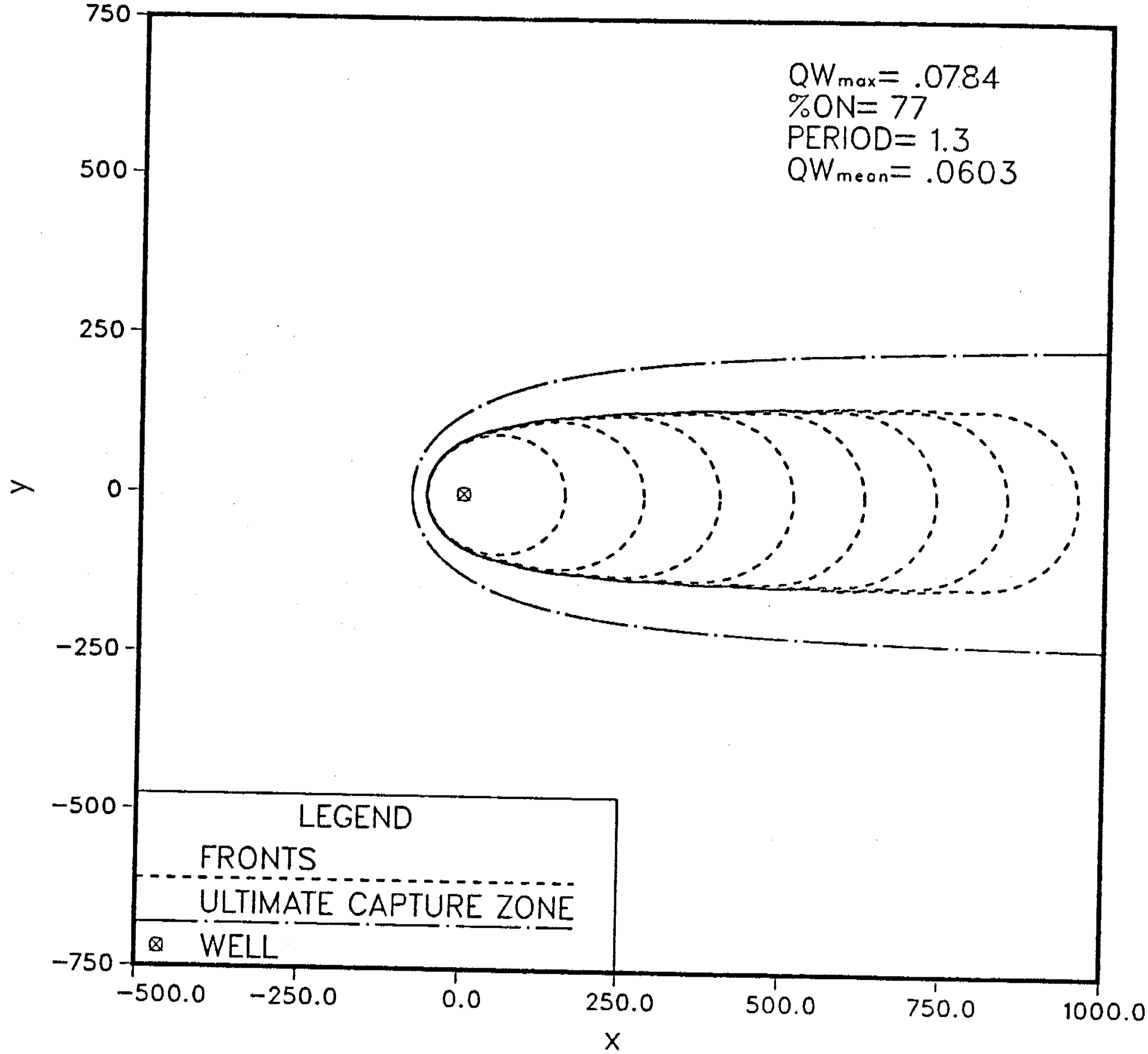
$$Q_w = \frac{\overline{Q_w} * P}{t_P} \quad (17)$$

tp - time on
P - period
$\overline{Q_w}$ - mean pumping rate

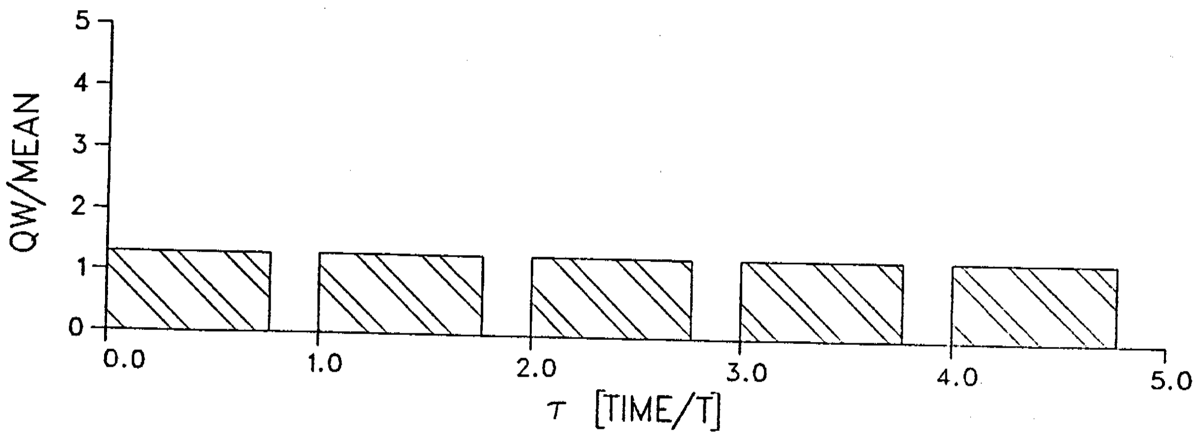
what is initial example! Figure 17?

Figures 17A through D is a simulation that is meant to represent a situation where the pump is turned off for only a short period of time (compared to the first example). The percent on is 77% and the maximum pumping rate is 0.0784. The period capture zones show a more profound lateral elongation than does the initial example (figure 17A). ?? The front pattern exhibits a smooth transition from one pumping period to the next in comparison to the hummocky front pattern for the initial example (figure 17B). The 100% and 0% boundaries converge much quicker almost eliminating the wing portion of the passed zone, and the head of the passed zone is significantly smaller than the previous example (figure 17C). Only one transect probability graph is presented due to the small size of the passed zone. The positive x-direction variation in extraction probability has a significantly greater slope than does the first example. The comparison to ultimate capture zones in figure 17D demonstrates the non-conservative nature of the mean pumping rate ultimate capture zone, on the other hand the effective stagnation point UCZ, as expected, is conservative and a fair approximation of the simulated capture zone (considered to be the 0% boundary).

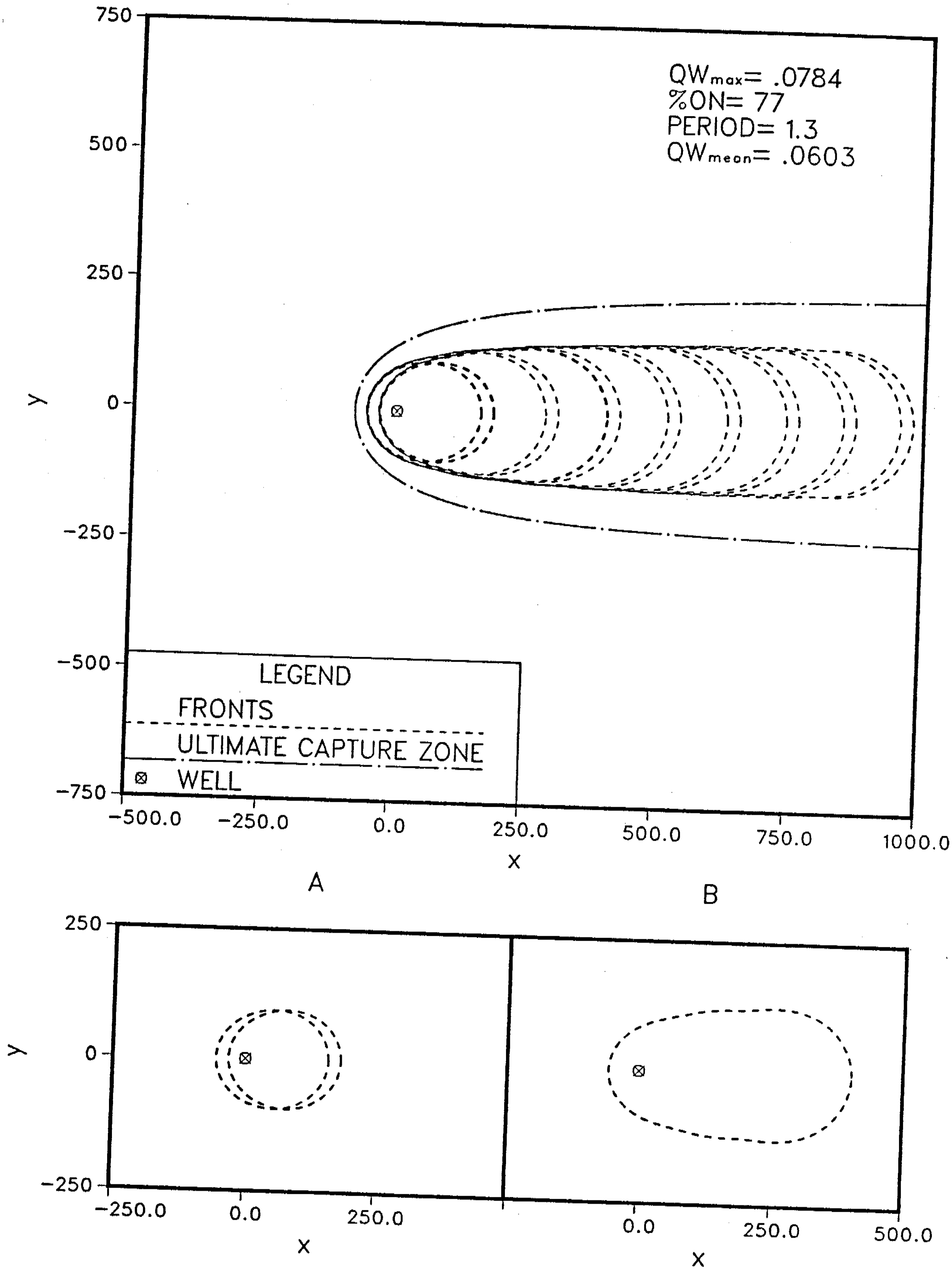
In the next simulation (figures 18A through D) the time off is exaggerated for comparison purposes. The ~~%~~ percent on is 3% and the maximum pumping rate is 1.96. This is the only pumping schedule that has an exaggerated y-axis. The normalized pumping factor is 32.5 making it impossible to fit on a similar scale as all the others (figure 18A). The PCZs exhibit a greater transverse expansion than either of the previous examples. The crescent moon pattern shown in figure (18B) demonstrates a disconnected front,



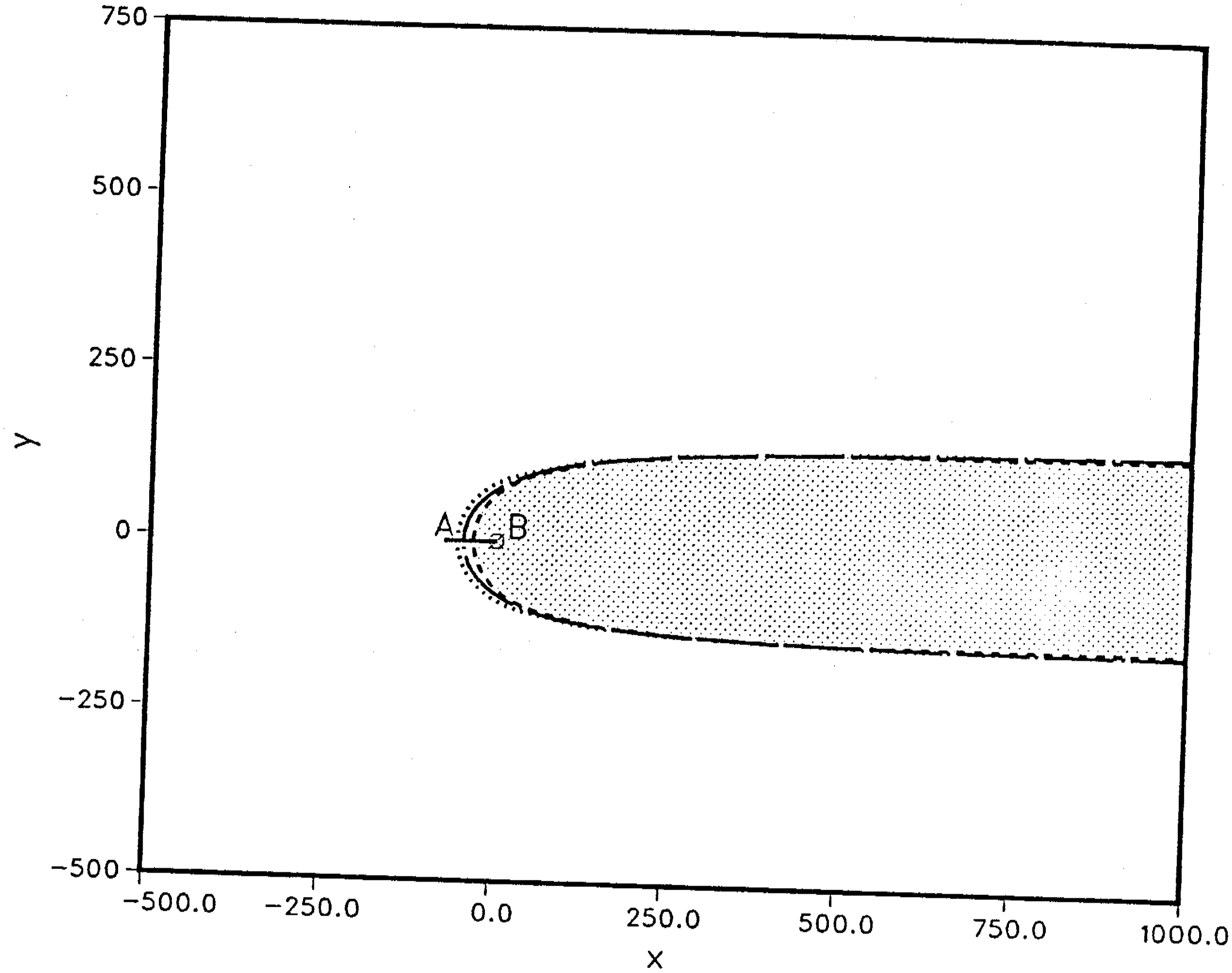

PUMPING SCHEDULE



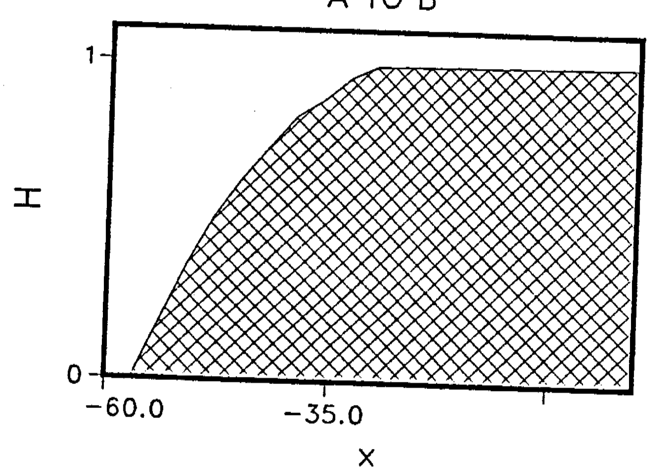
680


QWmax= .0784
%ON= 77
PERIOD= 1.3
QWmean= .0603
750
500
250
0
-250
-500
-750
y
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL
-500.0
-250.0
0.0
250.0
500.0
750.0
1000.0
x
A
B
250
0
-250
y
-250.0
0.0
250.0
x
0.0
250.0
500.0
x

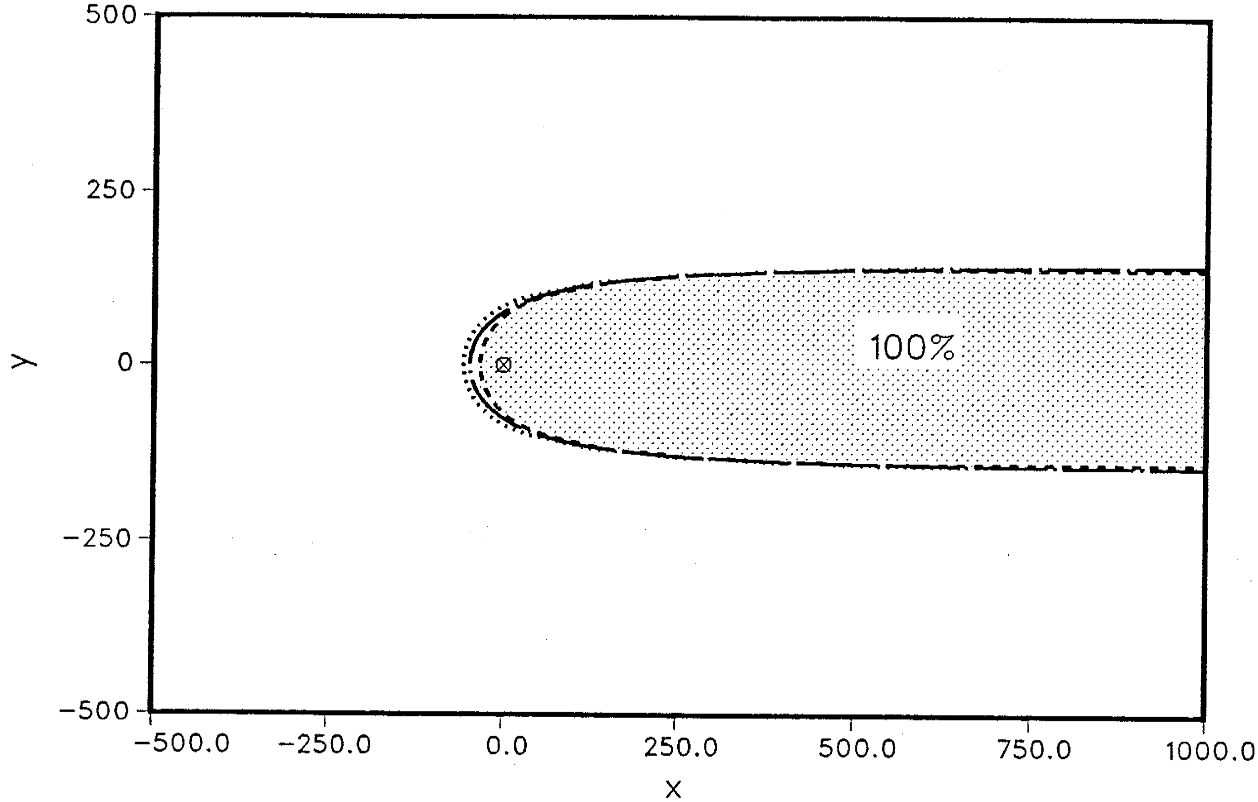
17



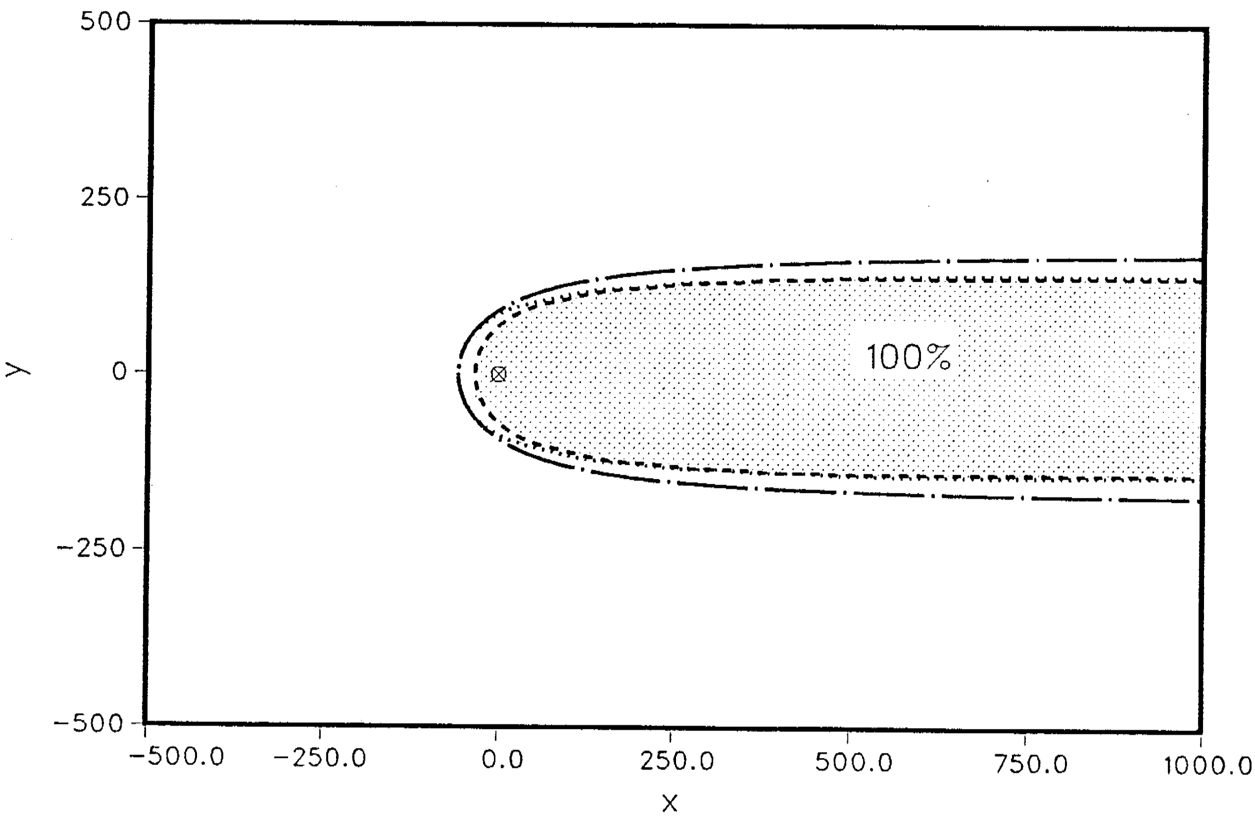

EXTRACTION PROBABILITY FOR
A TO B

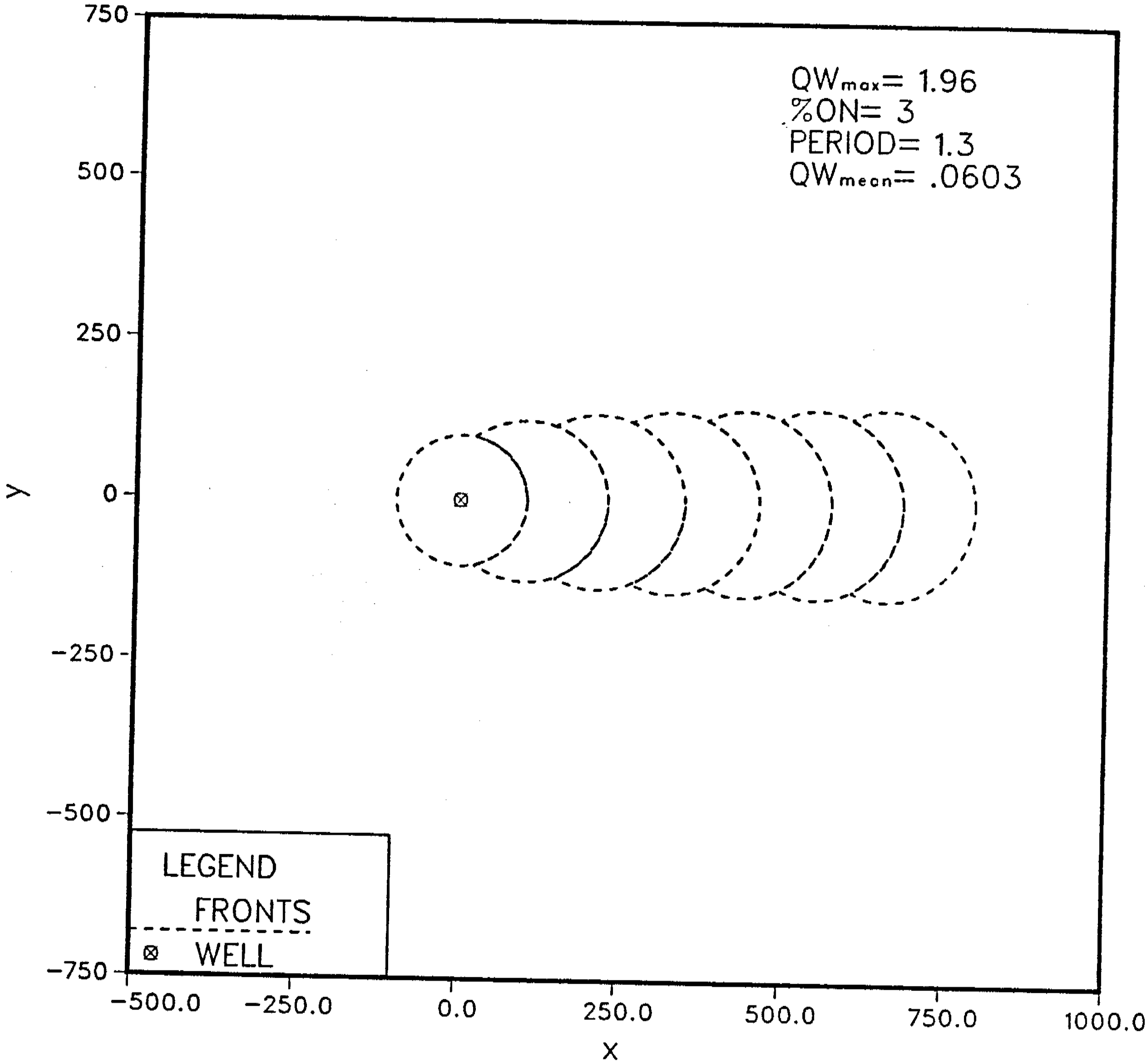


PROBABILITY ZONES WITH RESPECT
TO MEAN PUMPING RATE UCZ

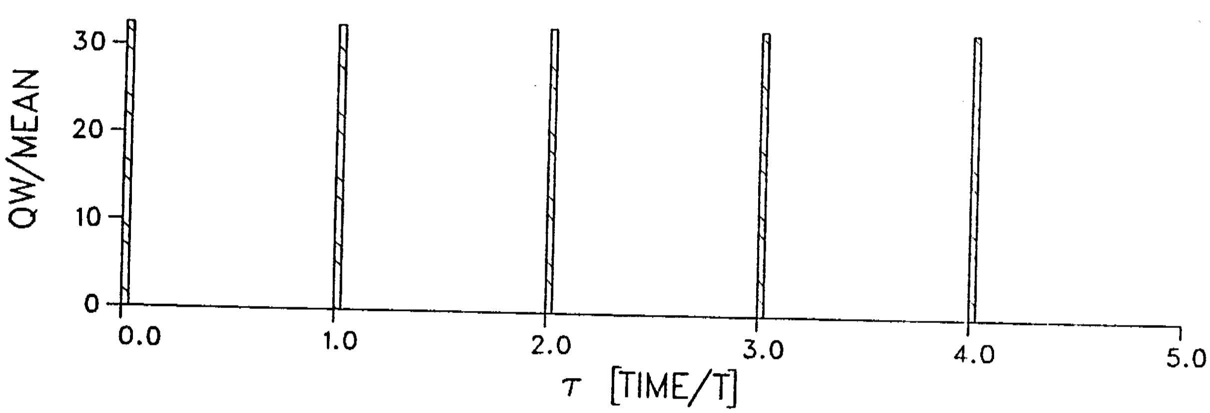


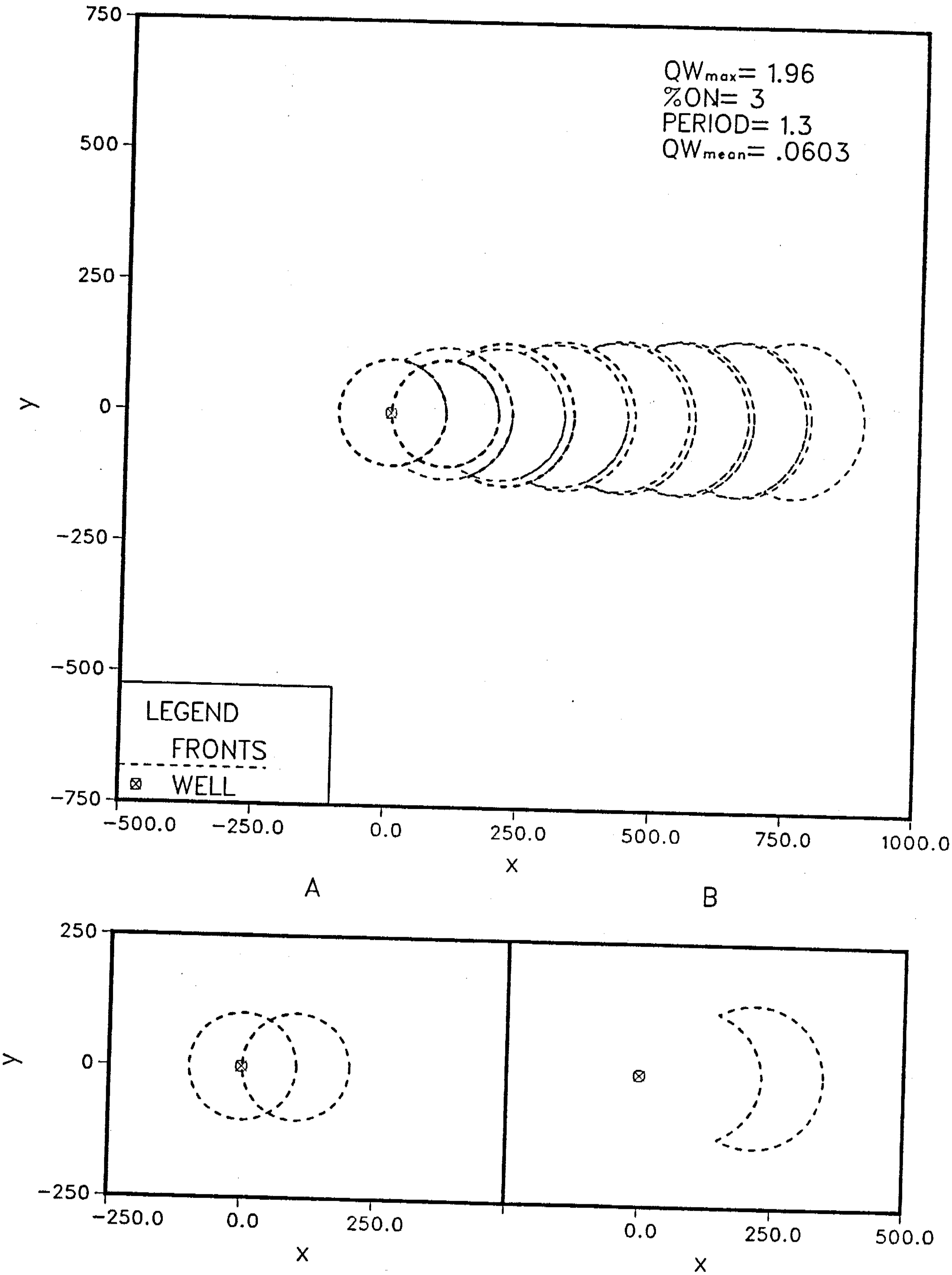

PROBABILITY ZONES WITH RESPECT
TO EFFECTIVE STAGNATION POINT UCZ






PUMPING SCHEDULE

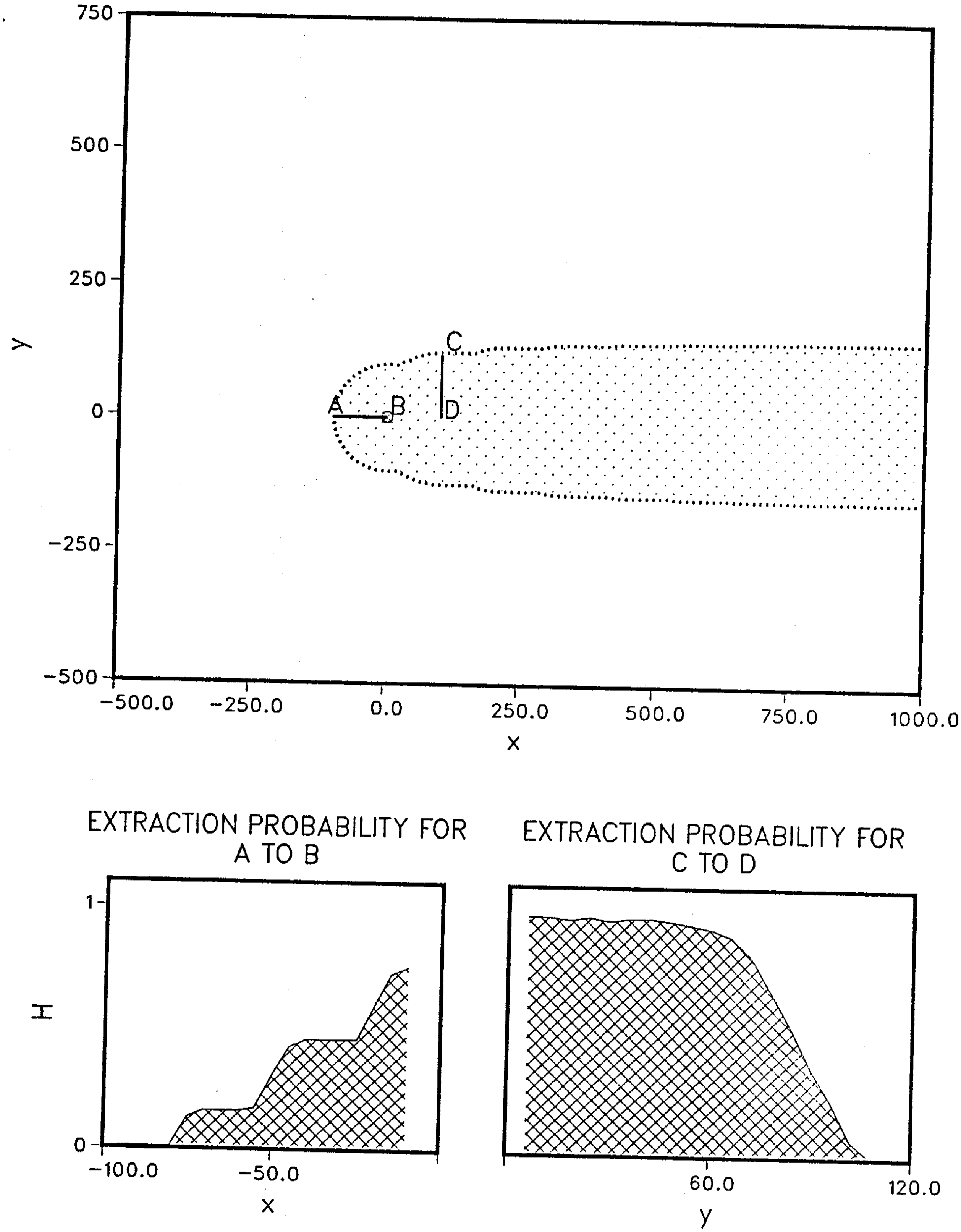



QWmax= 1.96
%ON= 3
PERIOD= 1.3
QWmean= .0603
LEGEND
FRONTS
WELL
y
x
750
500
250
0
−250
−500
−750
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
A
B
−250.0
0.0
250.0
0.0
250.0
500.0

one that does not encircle the well. Since the front is disconnected their is no 100% boundary, the passed zone is essentially the maximum extent of the fronts presented (figure 18C). There are two transect probability graphs, one through the head of the passed zone, parallel to the ambient flow direction and intersecting the well and one perpendicular to the other defining the transverse probabilities. The head transect probabilities has an obvious stair step shape and never reaches 100% (as expected). The transverse probabilities never reach 100% either which supports the theory of disconnection. The transverse probabilities do approach the 100% suggesting only slight disconnection. The transverse probabilities have a trend that looks like an exponential growth what? to a maximum. In figure 18D the 0% boundary is compared to the mean pumping rate UCZ and the effective stagnation point UCZ. Again the mean is non-conservative and it does a very poor job of estimating the simulated 0% boundary. The 0% boundary increases in a step fashion in the positive x-direction. Similar to previous simulations, the effective stagnation point UCZ is a much better conservative estimate of the capture zone than the mean, but over estimates it considerably.

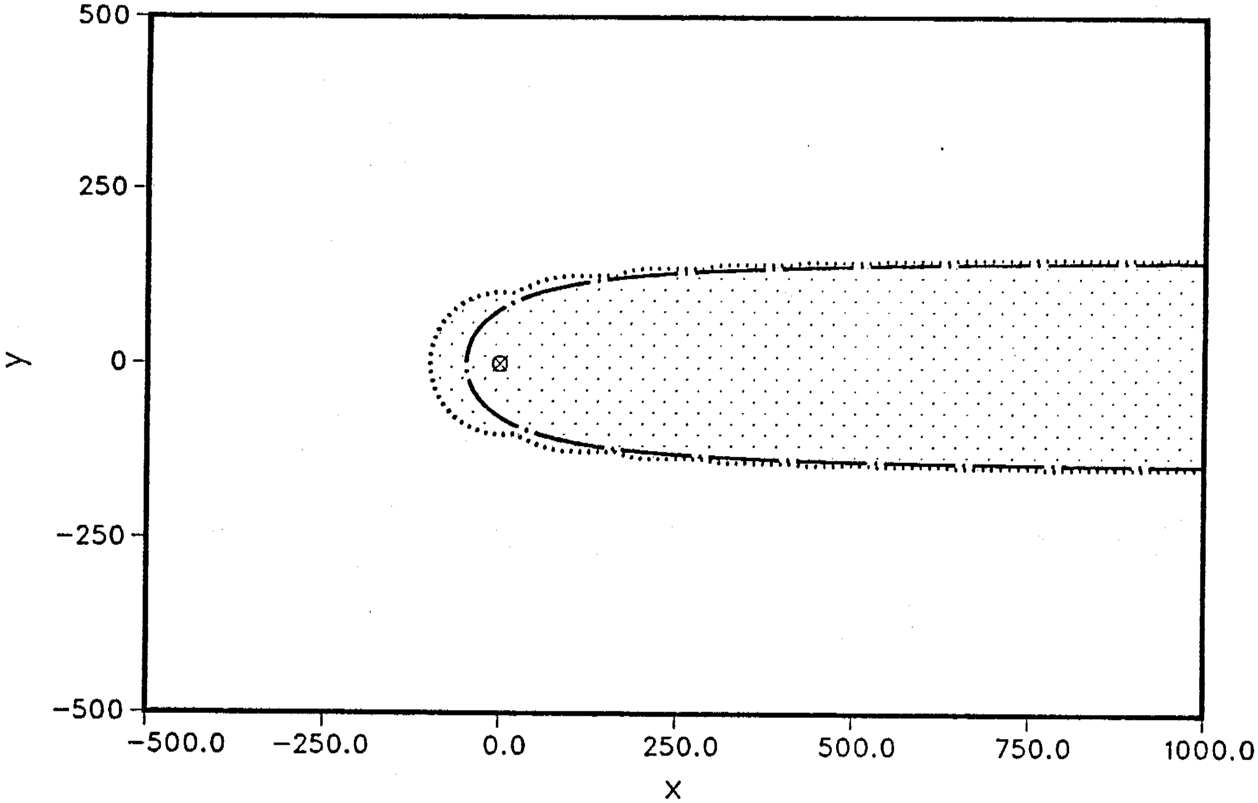
Often, the pump is not turned off, only reduced, the following examples model this type of scenario. The first example shows the resulting capture fronts when the pumping rate is only slightly reduced and the following simulation illustrates the effects when the reduction is more significant. The wave forms are still square waves. In the first example the %on is reversed from the initial example, the maximum pumping rate is 0.0708 and the pump is at this rate for .8 dimensionless time units. During the remaining portion of the cycle, the pumping rate is .0436. The PCZ's are illustrated in Figure 19A, They look very similar to the simulation with a 77% on. Examining the front pattern (Figure 19B), it has a similar smooth evolution through time. Inset A now shows the position of the front at the time the pumping is reduced and at the end of the first period. These two fronts do not intersect. In this example it is not easy to delineate the passed zone. In Figure 17C a transect probability shows a much more abrupt shape than does the one in the 77% on example, this seems to be the only significant difference between the two. The comparison with the ultimate capture zones looks very similar also (Figure 19D).

In the next simulation the difference between the two pumping rates is much greater, the maximum is

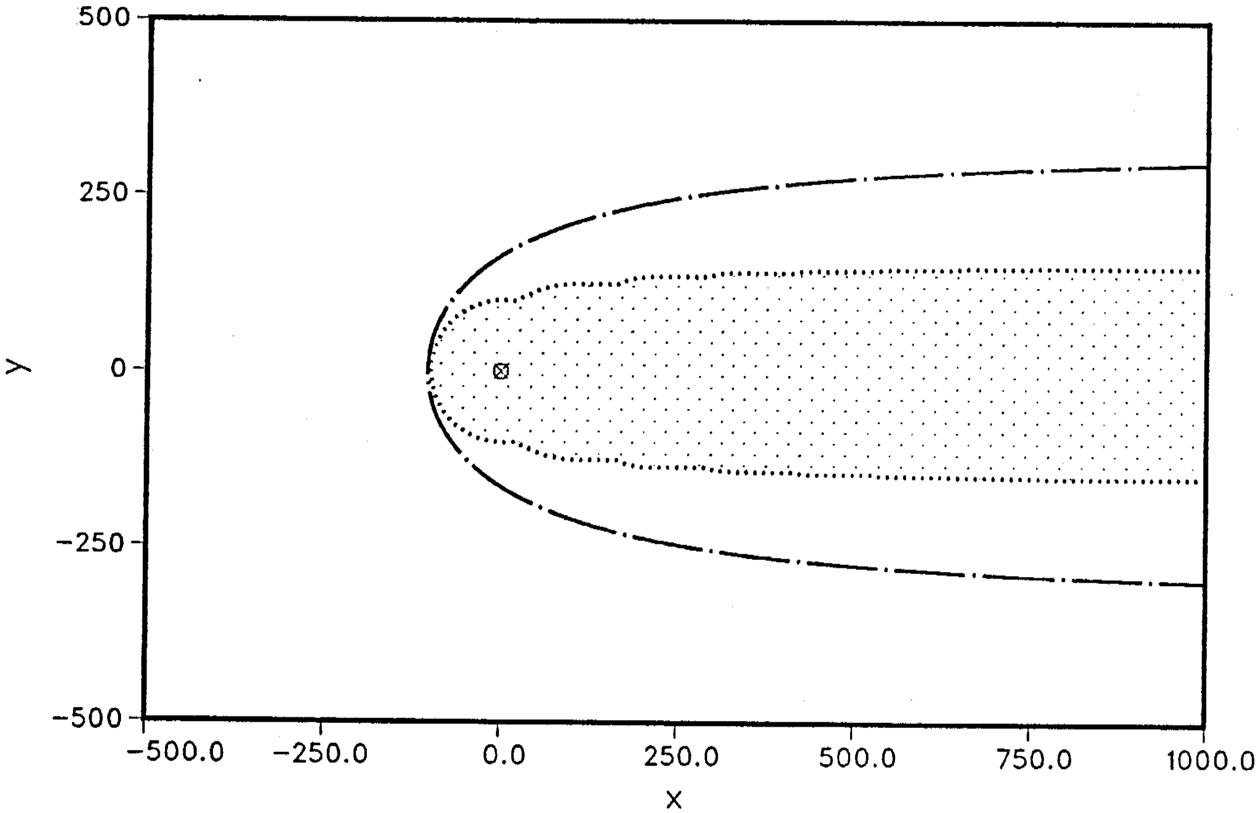

750
500
250
0
−250
−500
y
C
A
B
D
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x
EXTRACTION PROBABILITY FOR
A TO B
1
0
H
−100.0
−50.0
x
EXTRACTION PROBABILITY FOR
C TO D
60.0
120.0
y

G90

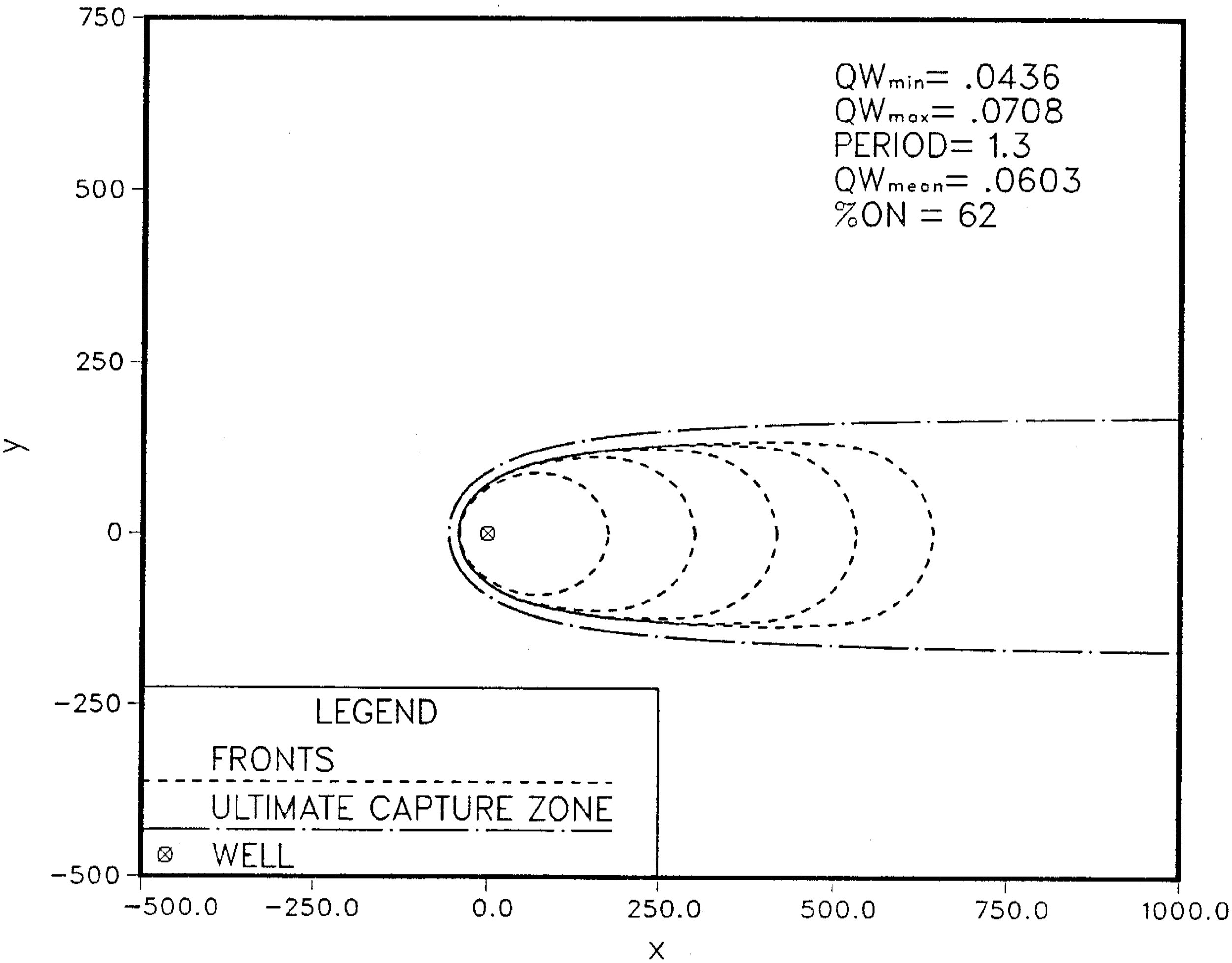
PROBABILITY ZONES WITH RESPECT
TO MEAN PUMPING RATE UCZ



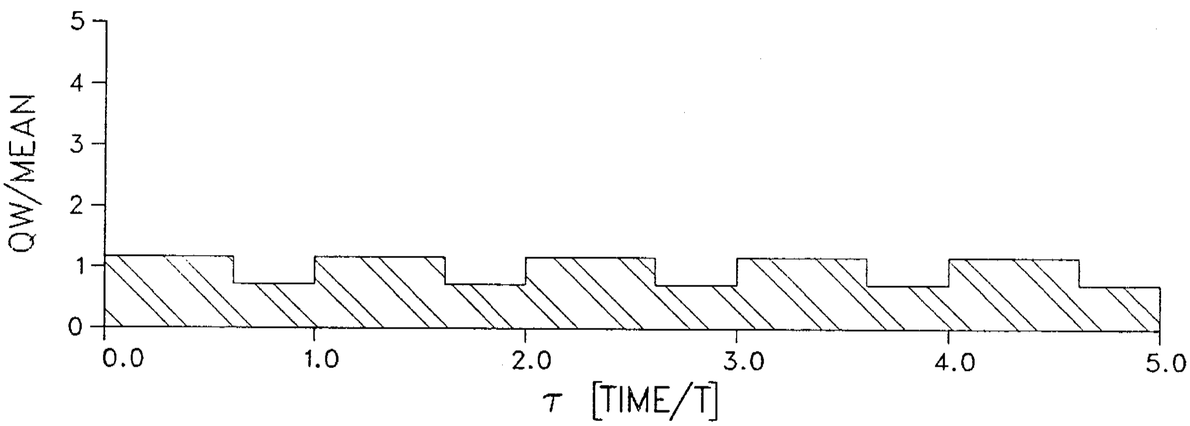

PROBABILITY ZONES WITH RESPECT
TO EFFECTIVE STAGNATION POINT UCZ

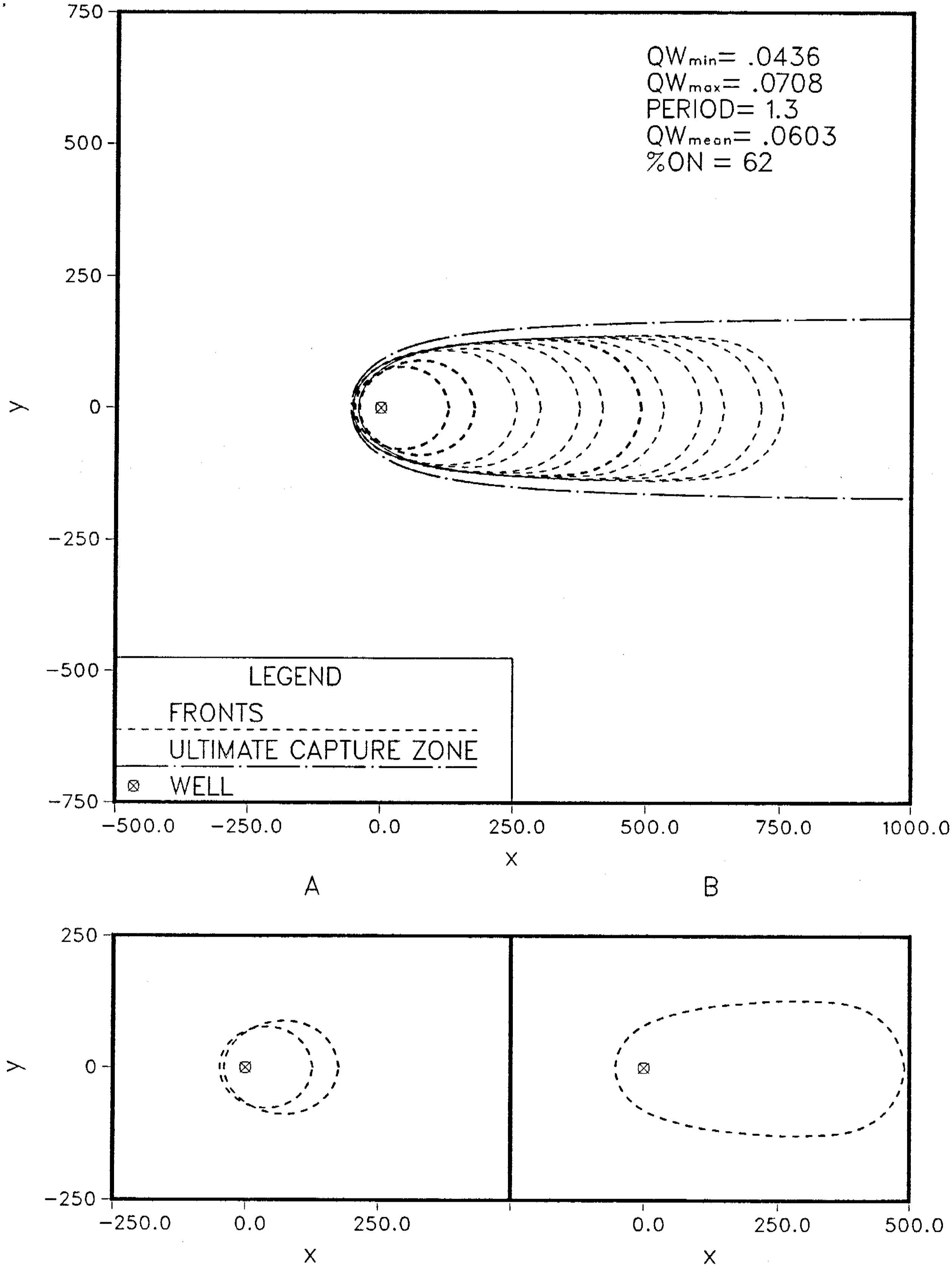


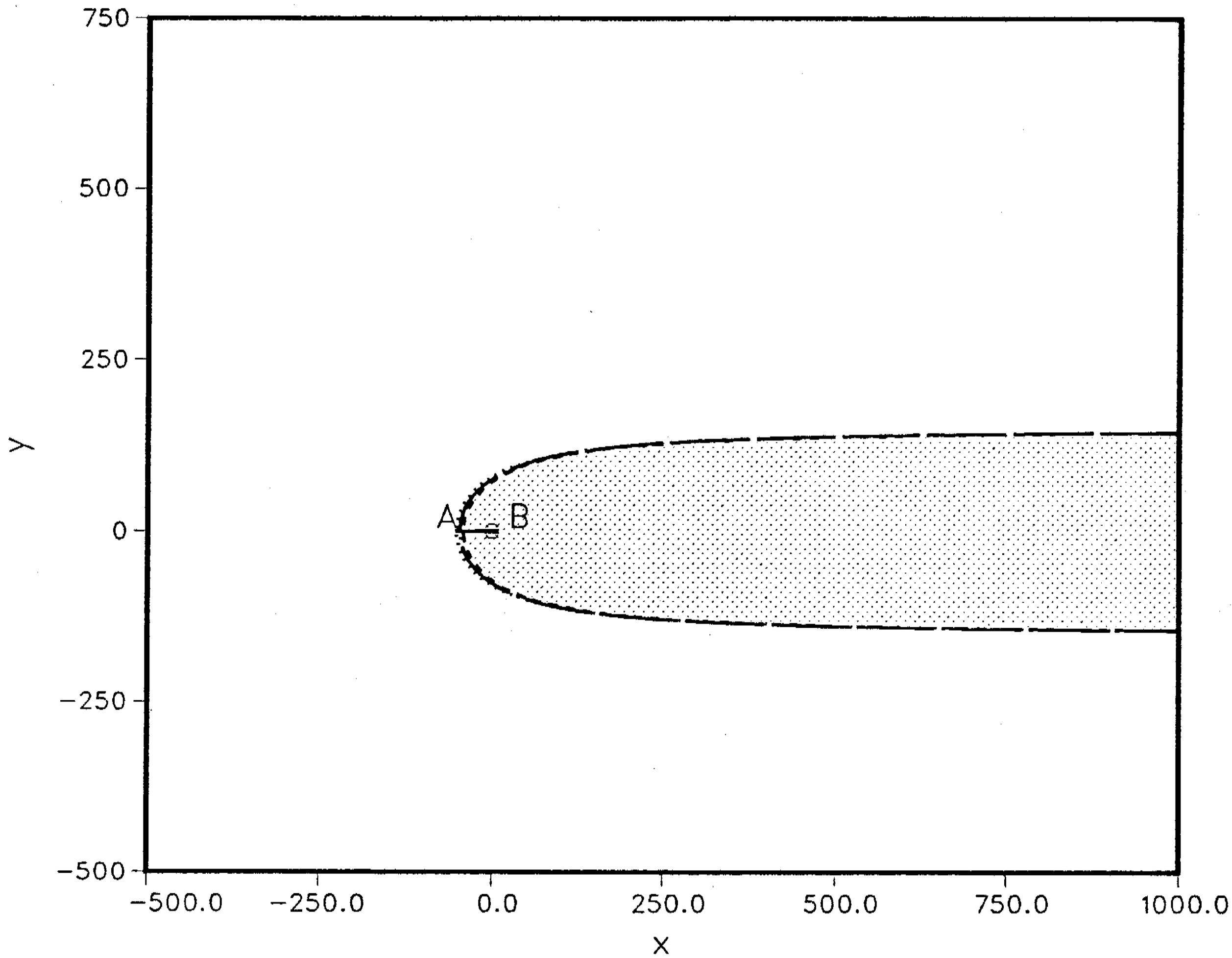

18D



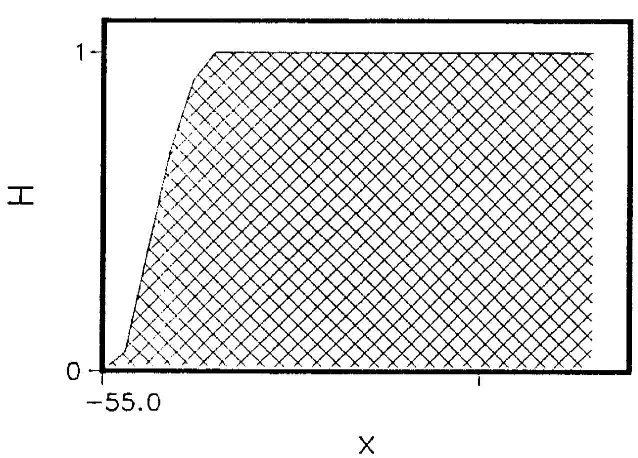

PUMPING SCHEDULE

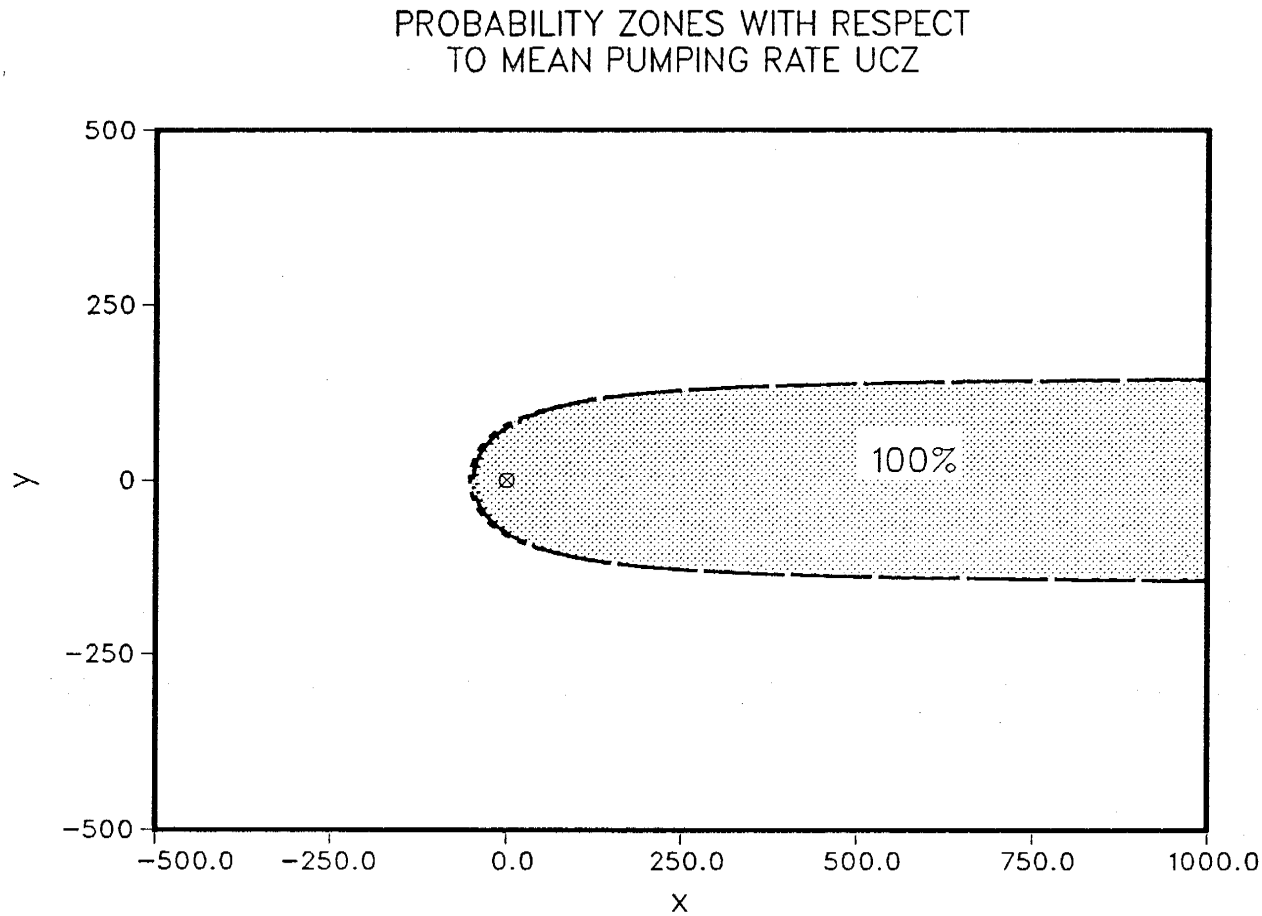


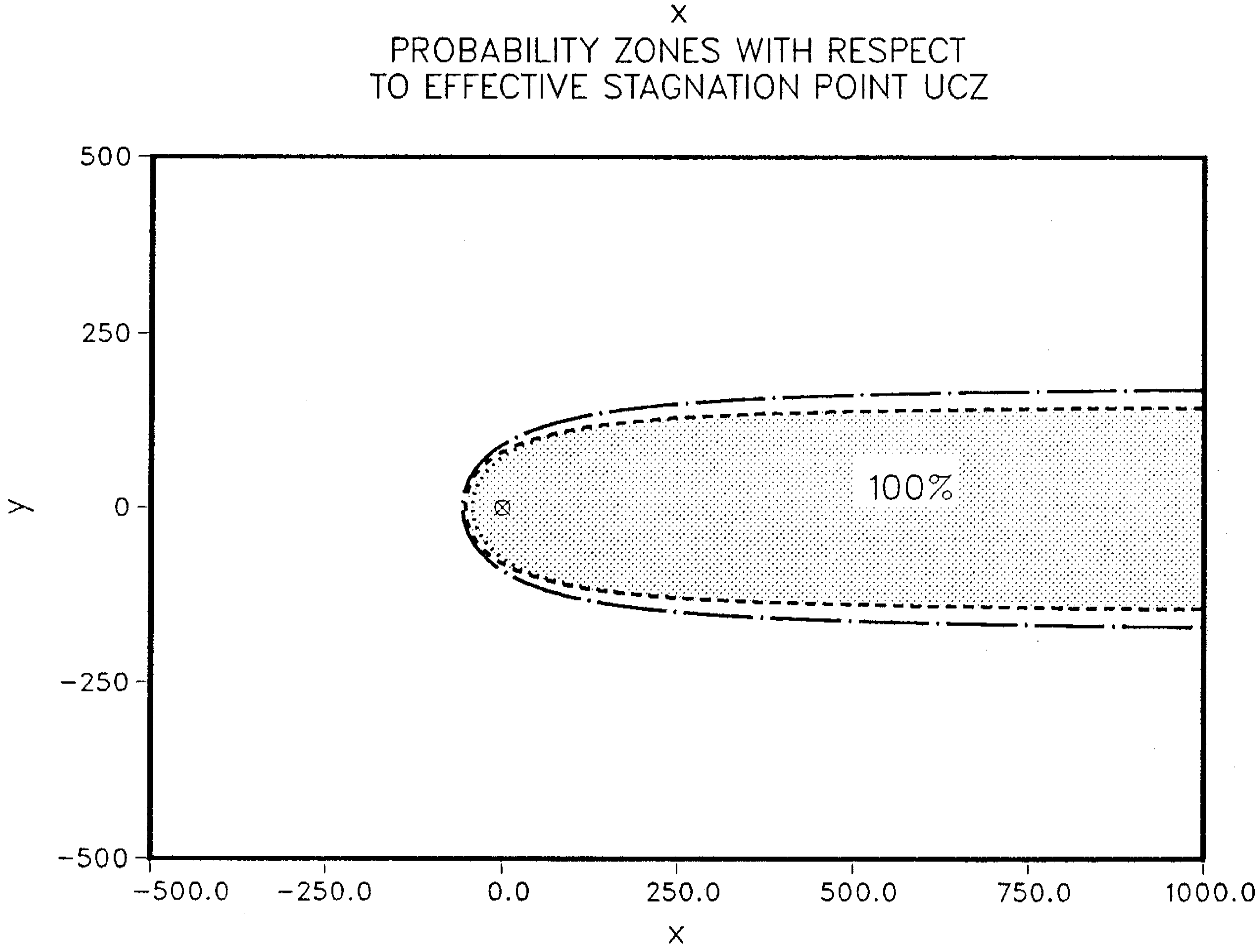

QWmin= .0436
QWmax= .0708
PERIOD= 1.3
QWmean= .0603
%ON = 62
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL
750
500
250
0
-250
-500
-750
y
-500.0
-250.0
0.0
250.0
500.0
750.0
1000.0
x
A
B
250
0
-250
y
-250.0
0.0
250.0
x
0.0
250.0
500.0
x




EXTRACTION PROBABILITY FOR
A TO B




PROBABILITY ZONES WITH RESPECT
TO MEAN PUMPING RATE UCZ
500
250
0
-250
-500
y
100%
-500.0
-250.0
0.0
250.0
500.0
750.0
1000.0
x


PROBABILITY ZONES WITH RESPECT
TO EFFECTIVE STAGNATION POINT UCZ
500
250
0
-250
-500
y
100%
-500.0
-250.0
0.0
250.0
500.0
750.0
1000.0
x

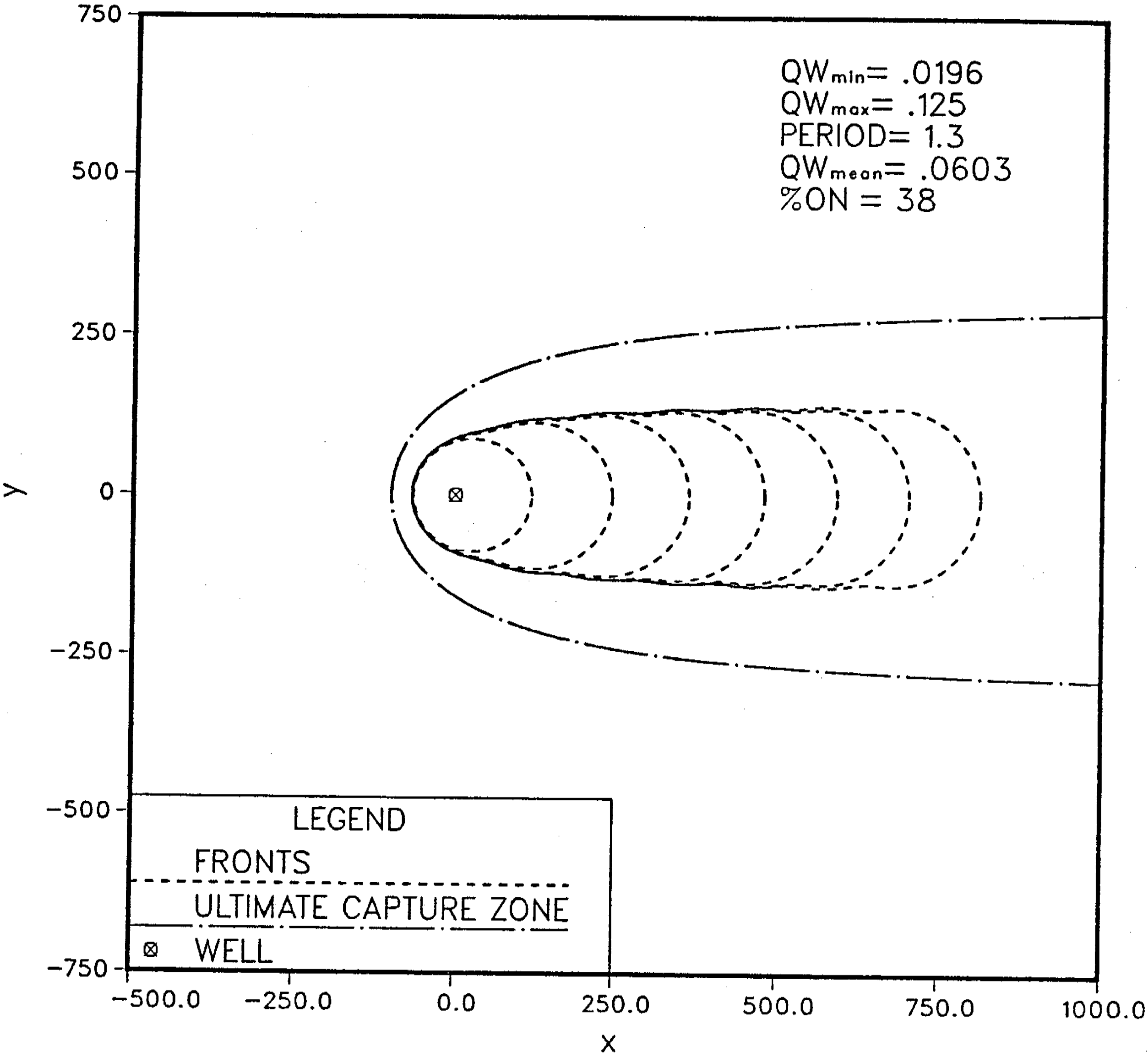
0.125, the time on is switched ($t_P$ = .5) and the minimum is .0196 (figure 20A). The front pattern for this case has a faint hummocky shape (figure 20B). The passed zone is much more prevalent in this simulation compared to the previous example (figure 20C). The cross-section probabilities for the head of the passed zone is almost perfectly linear, and the transverse shows very little passed zone. The mean pumping rate UCZ still is non-conservative unlike the effective stagnation point UCZ.

A simulation is presented here that has an irregular pumping rate. The value of pumping per portion of the period is generated from a random number generator. The change in pumping rate occurs at the same intervals as in the first example. The PCZs are difficult to discern (figure 21A). They have an irregular shape, but they do not cross. The front pattern in figure 21B illustrates the unusual shape of the fronts. Inset A shows the position of the front at the change in pumping and at the end of the first period. There are only five pumping cycles represented here so the 100% boundary and the 0% boundary are not for $t = \infty$ and don't extend passed the fifth cycle. This illustrates a time dependent capture zone. The extraction probabilities do not reach 100% due to the fact that the final particles do not reach the well before the five cycles have been completed (21C). The transect probabilities reflect the irregular nature if the flow field. The comparison graph (figure 21D) does illustrate the possibility that a 100% boundary can cross a mean pumping rate UCZ.

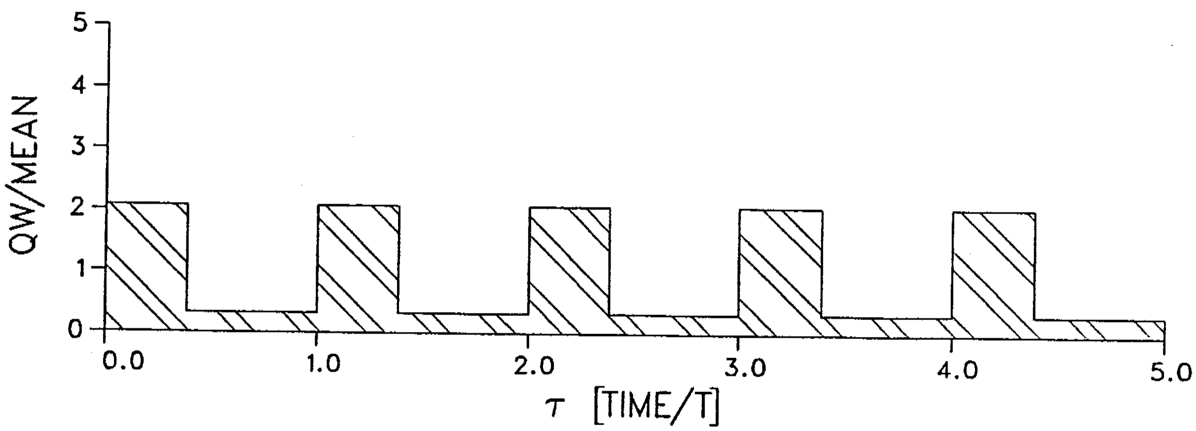
The final simulation presented in this section is one with a continuously varying pumping rate. The pumping rate is a sine wave with the minimum value being zero. The mean is still 0.0603 and the period is 1.3 (figure 22A). The fronts are very uniform and no trace of a hummocky pattern is apparent (figure 22B). There is no passed zone, therefore, the transect probabilities is 100% for the entire zone. This example illustrates the only time the mean pumping rate UCZ is actually conservative (figure 22C). This is exactly equal to having a mean constant pumping.

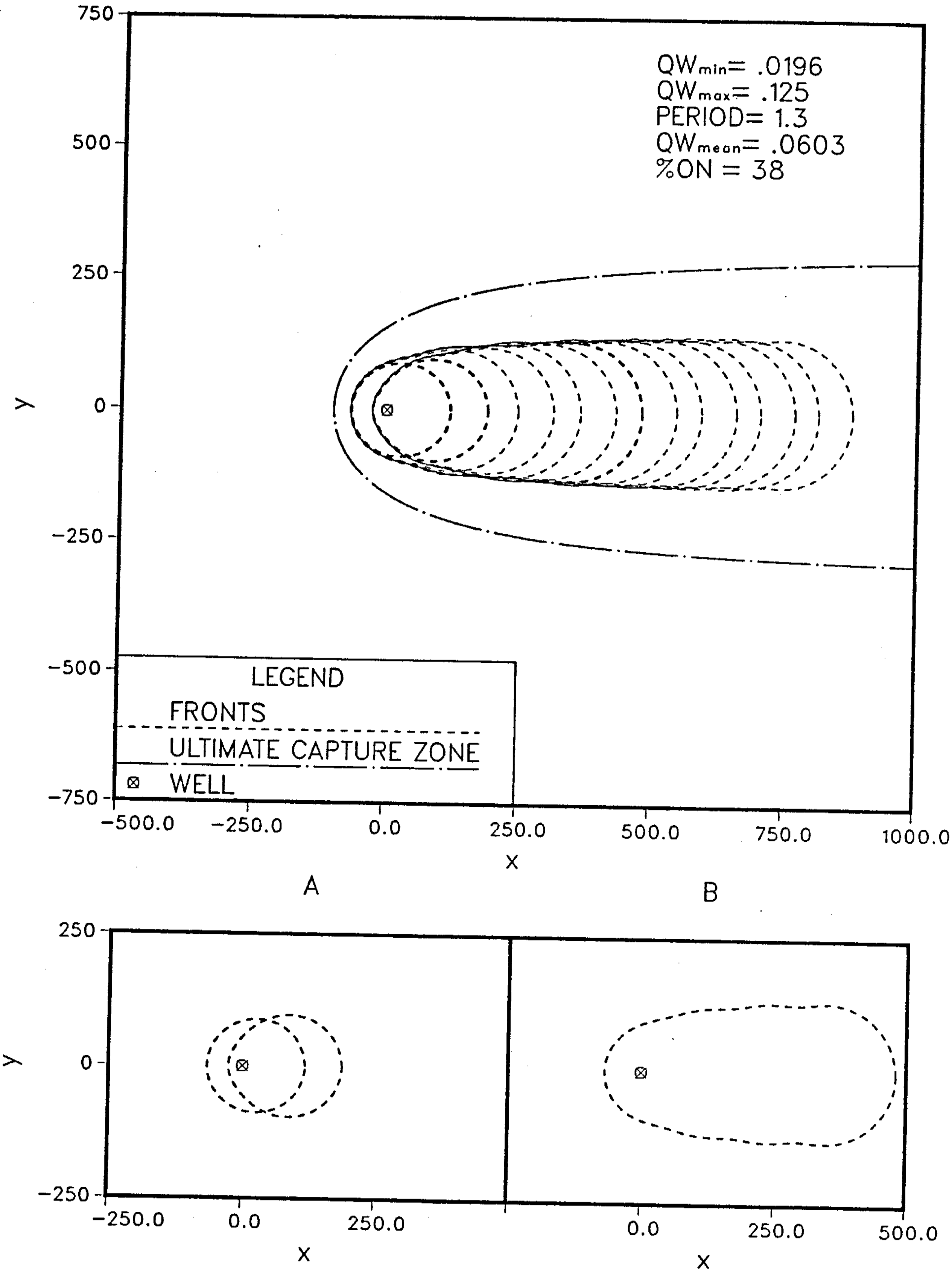
**Constant Amplitude Simulations and Dynamic Analysis**

In this second portion of the sensitivity analysis the amplitude (pumping rate) and the pumped time (on time) remain constant in an attempt to simplify comparison between simulations. This approach suppresses one degree of freedom (influence due to pumping) for each of the following simulations.

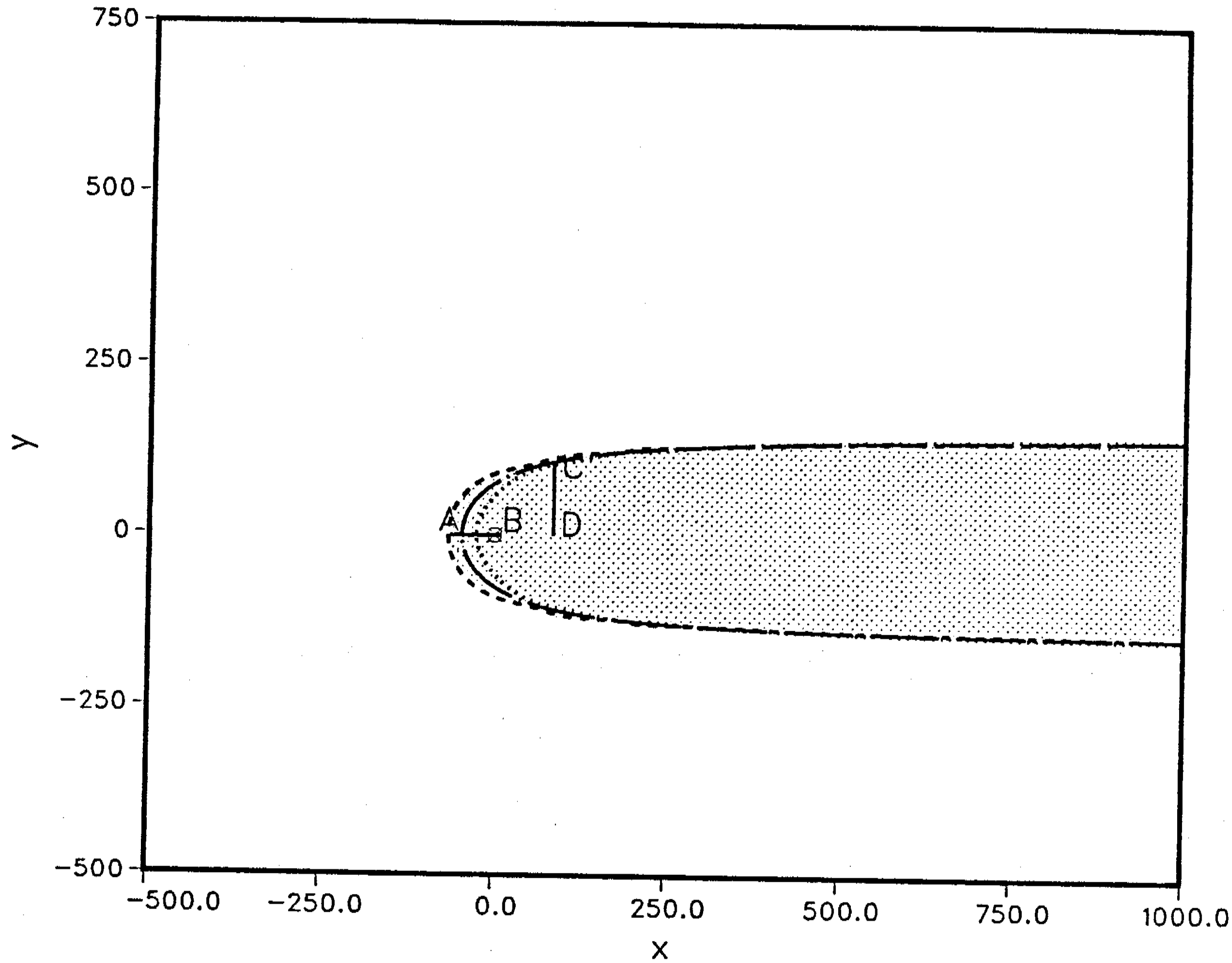


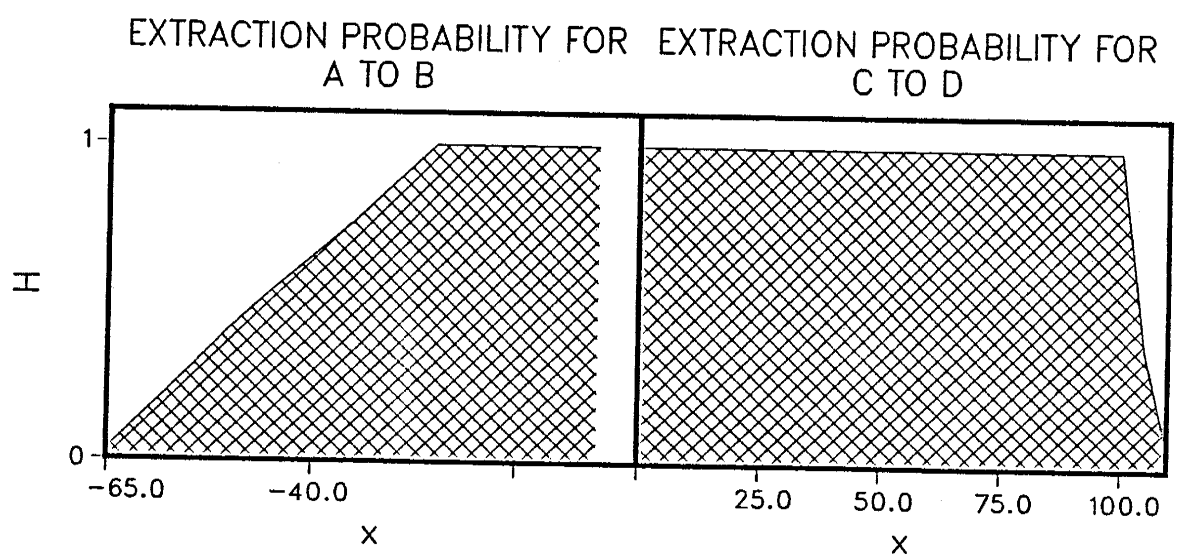

PUMPING SCHEDULE



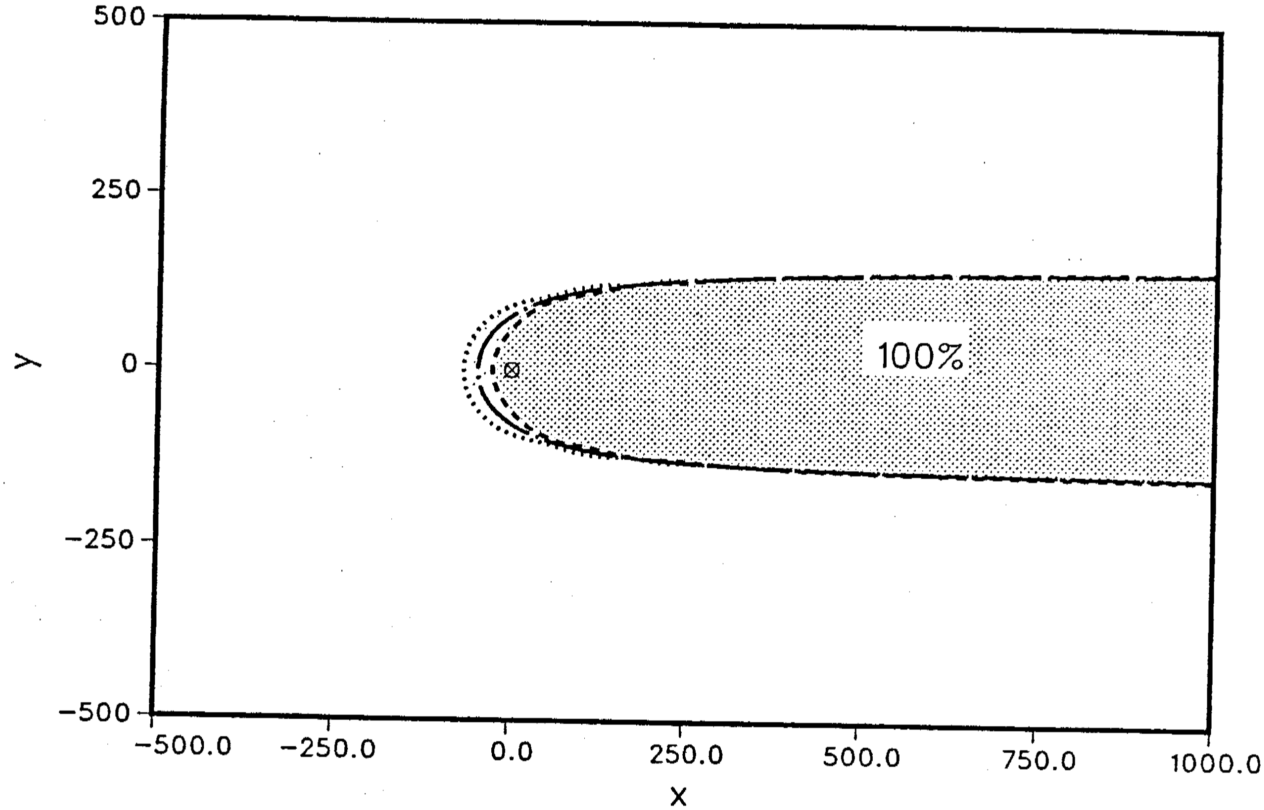


$QW_{min}$= .0196
$QW_{max}$= .125
PERIOD= 1.3
$QW_{mean}$= .0603
%ON = 38
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL
y
x
A
B

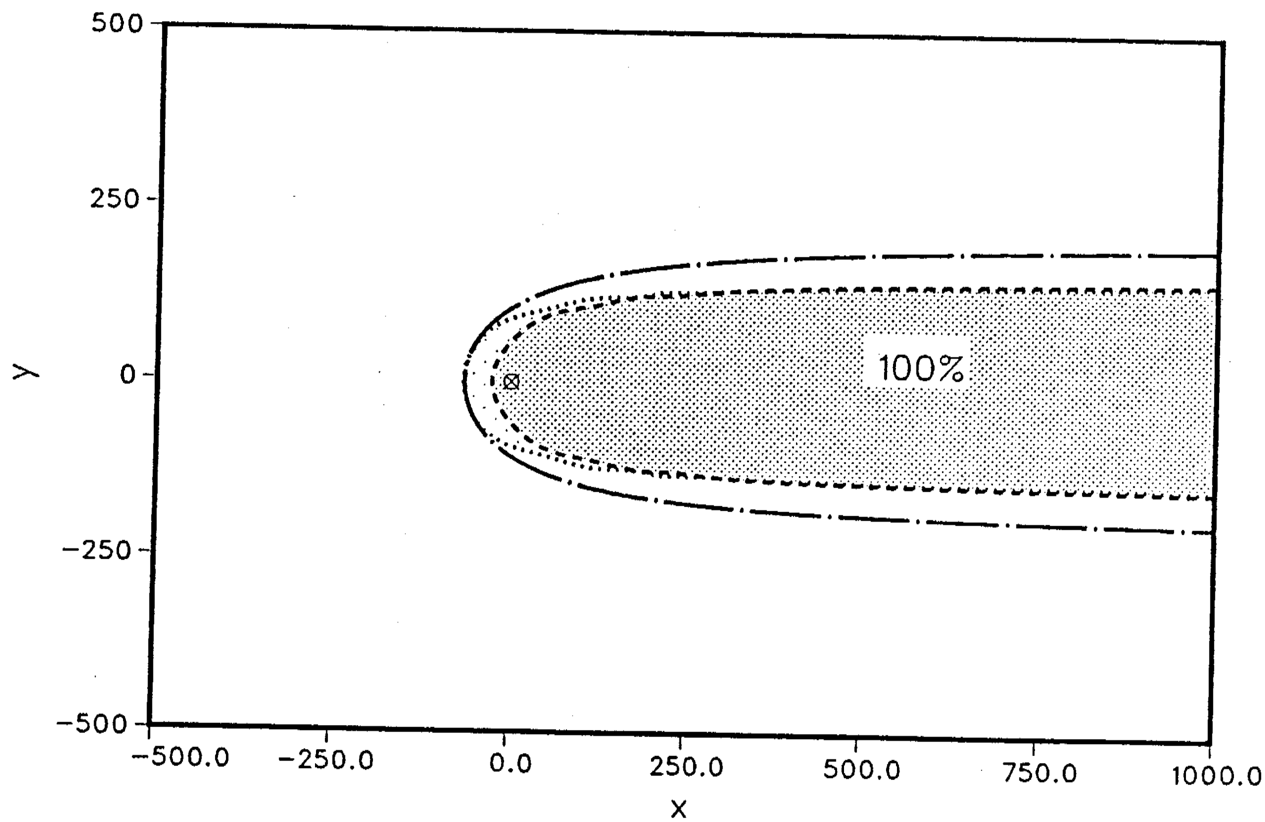

750
500
250
0
−250
−500
y
A
B
C
D
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x

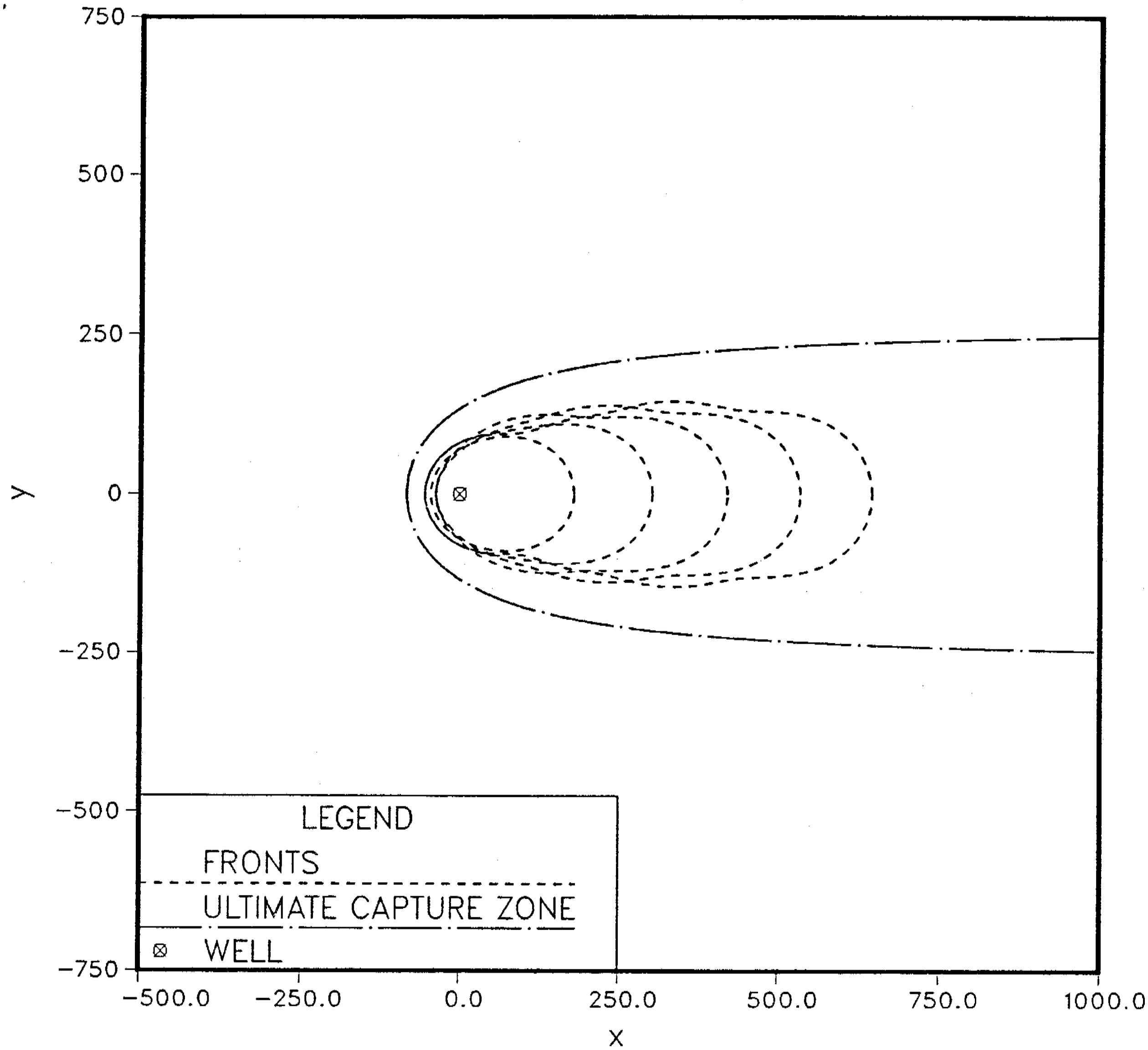

EXTRACTION PROBABILITY FOR
A TO B
EXTRACTION PROBABILITY FOR
C TO D
H
1
0
−65.0
−40.0
x
25.0
50.0
75.0
100.0
x

PROBABILITY ZONES WITH RESPECT
TO MEAN PUMPING RATE UCZ

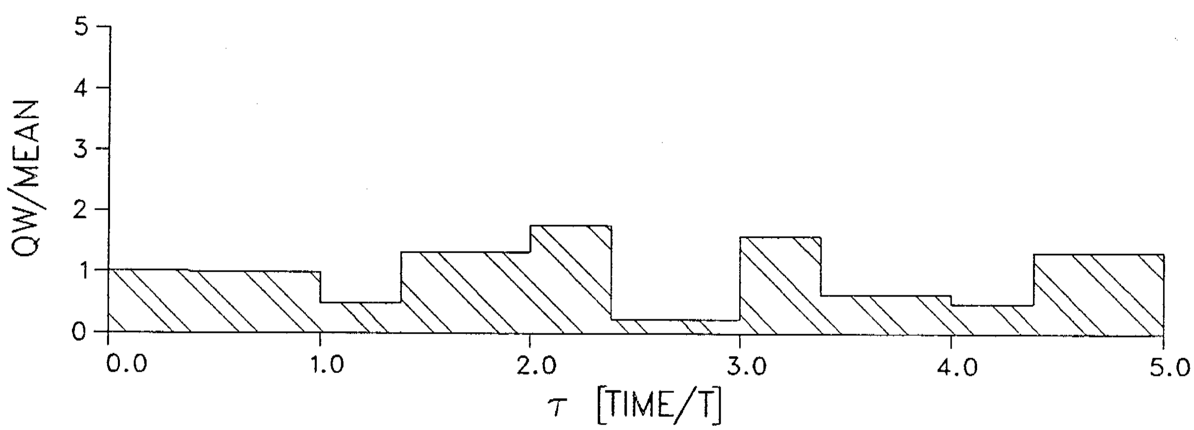


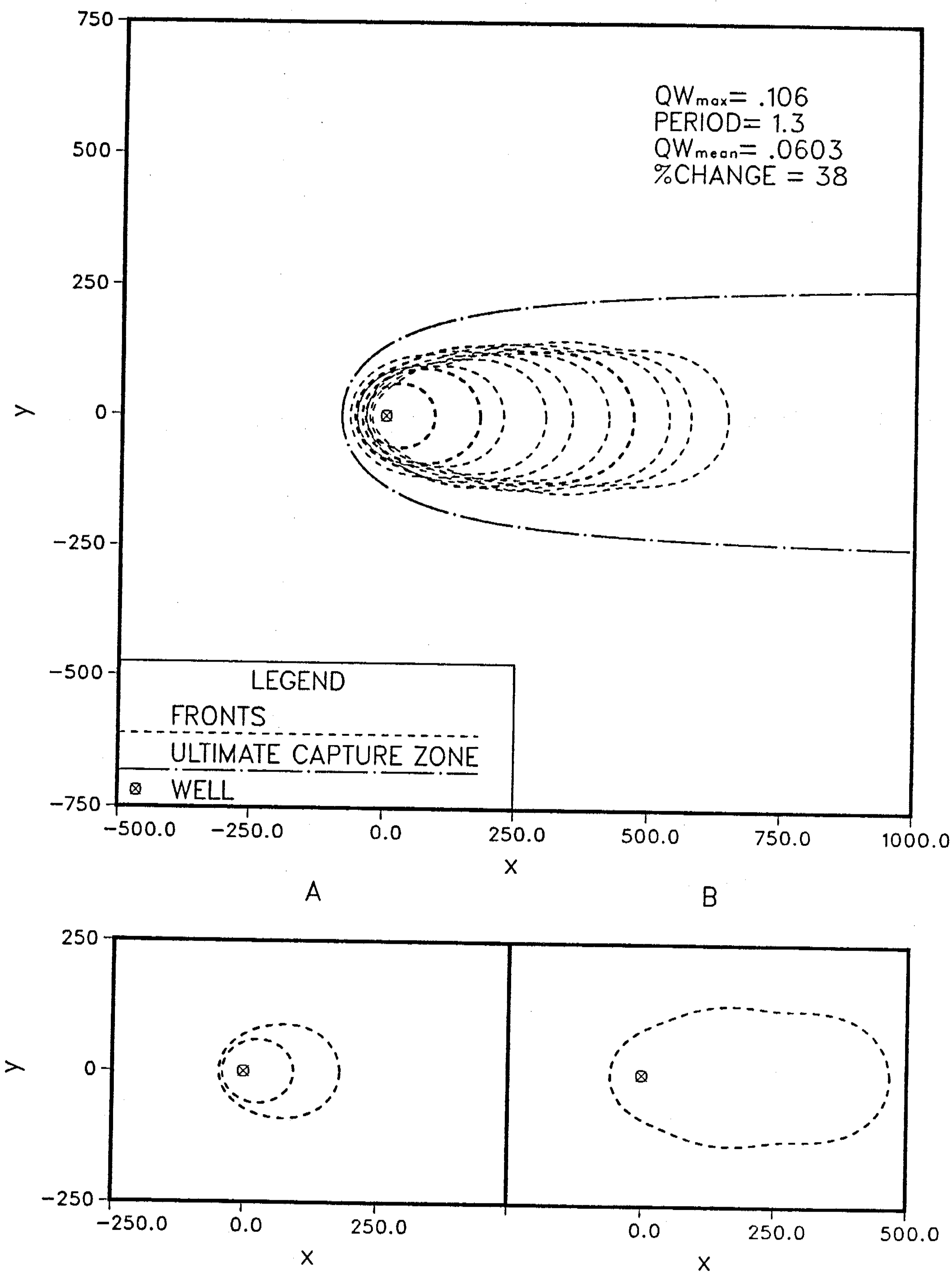

PROBABILITY ZONES WITH RESPECT
TO EFFECTIVE STAGNATION POINT UCZ






PUMPING SCHEDULE

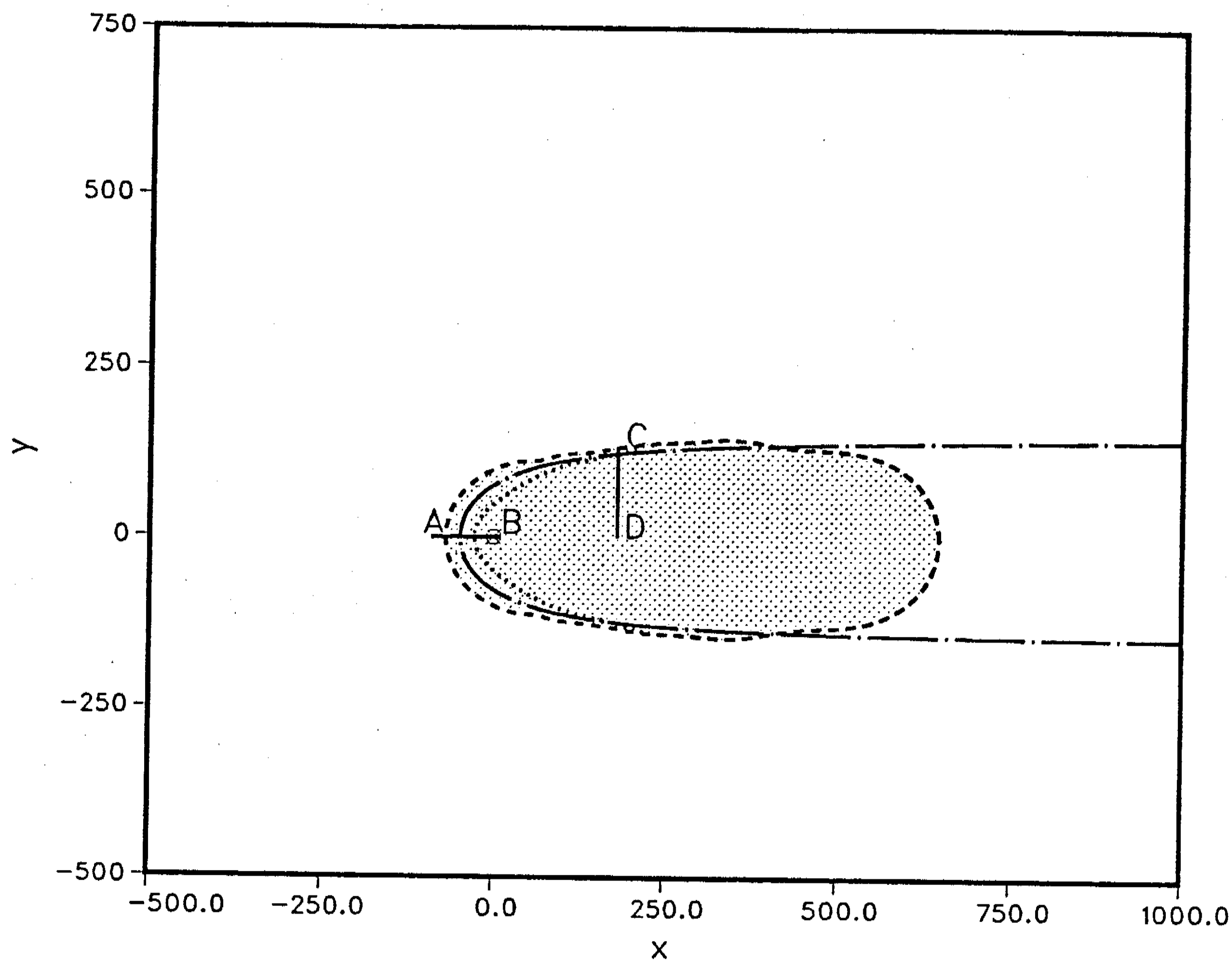


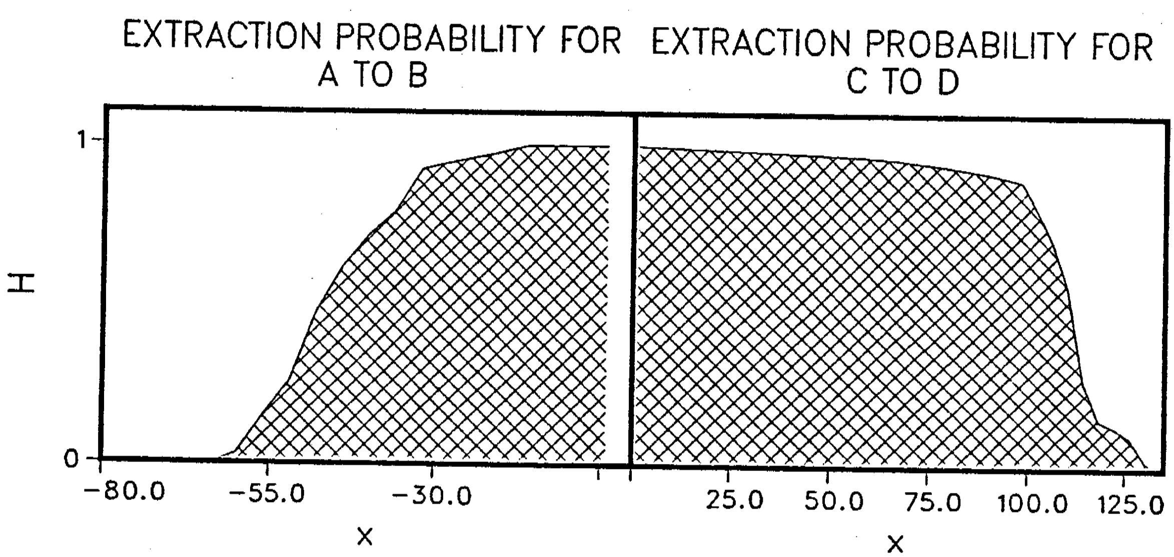


QWmax= .106
PERIOD= 1.3
QWmean= .0603
%CHANGE = 38
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL
750
500
250
0
−250
−500
−750
y
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x
A
B
250
0
−250
y
−250.0
0.0
250.0
x
0.0
250.0
500.0
x

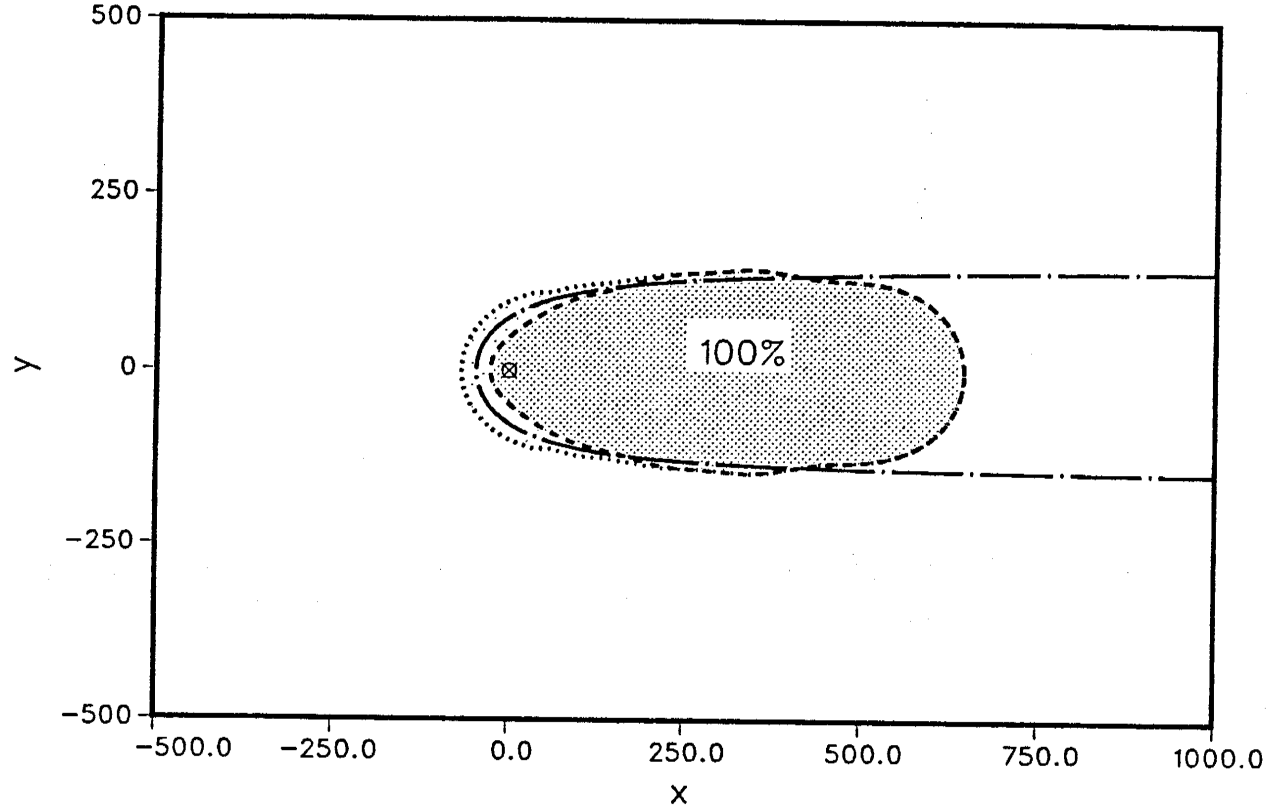

5

213

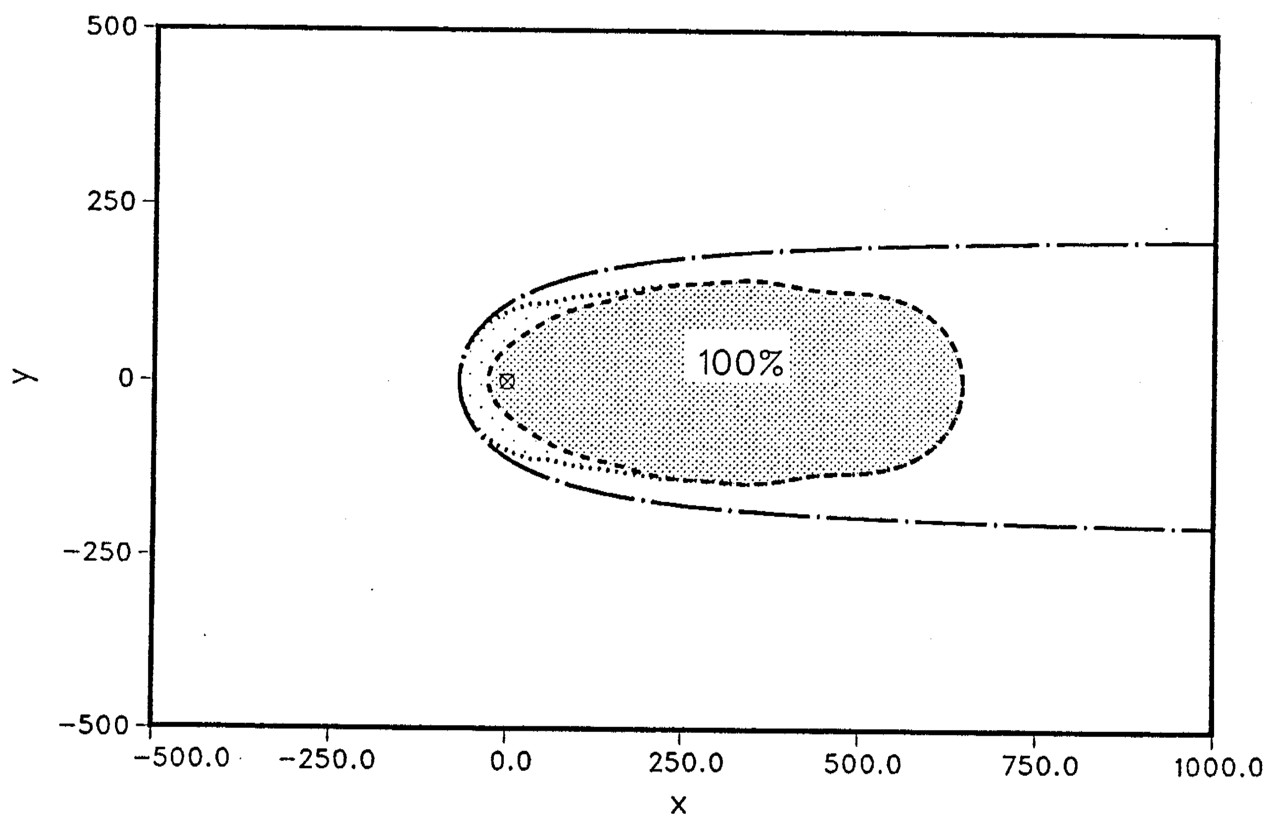

750
500
250
0
−250
−500
y
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x
A
B
C
D


EXTRACTION PROBABILITY FOR A TO B
EXTRACTION PROBABILITY FOR C TO D
1
0
H
−80.0
−55.0
−30.0
x
25.0
50.0
75.0
100.0
125.0
x

PROBABILITY ZONES WITH RESPECT
TO MEAN PUMPING RATE UCZ

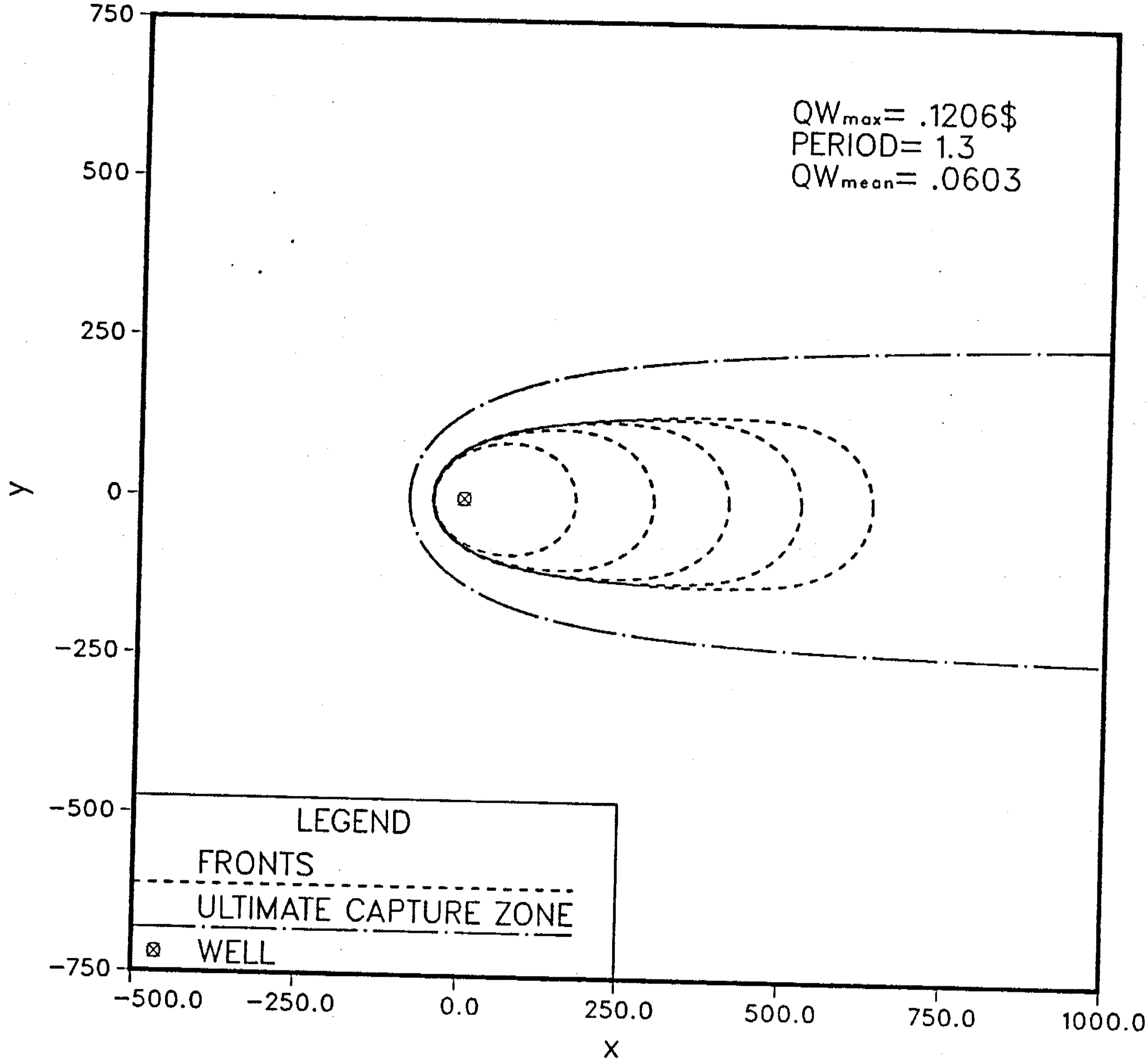



PROBABILITY ZONES WITH RESPECT
TO EFFECTIVE STAGNATION POINT UCZ

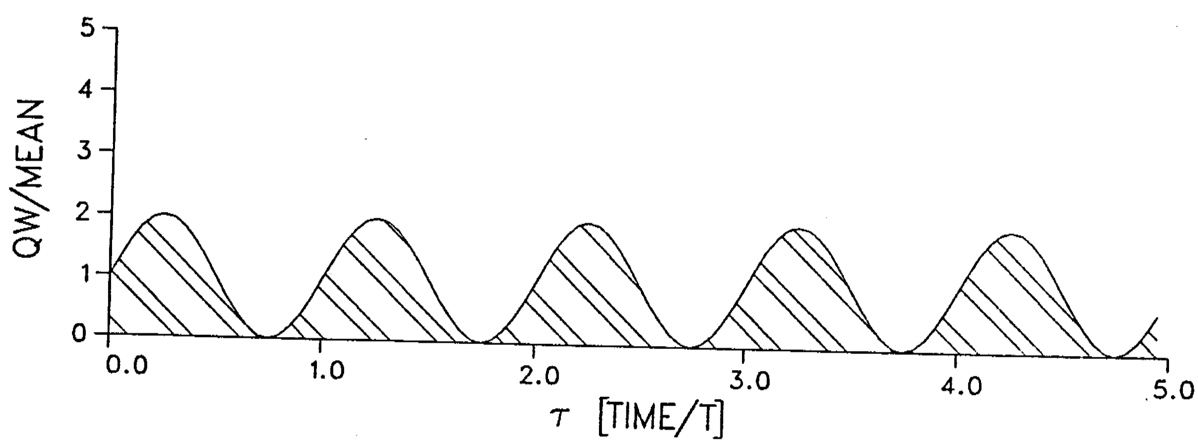


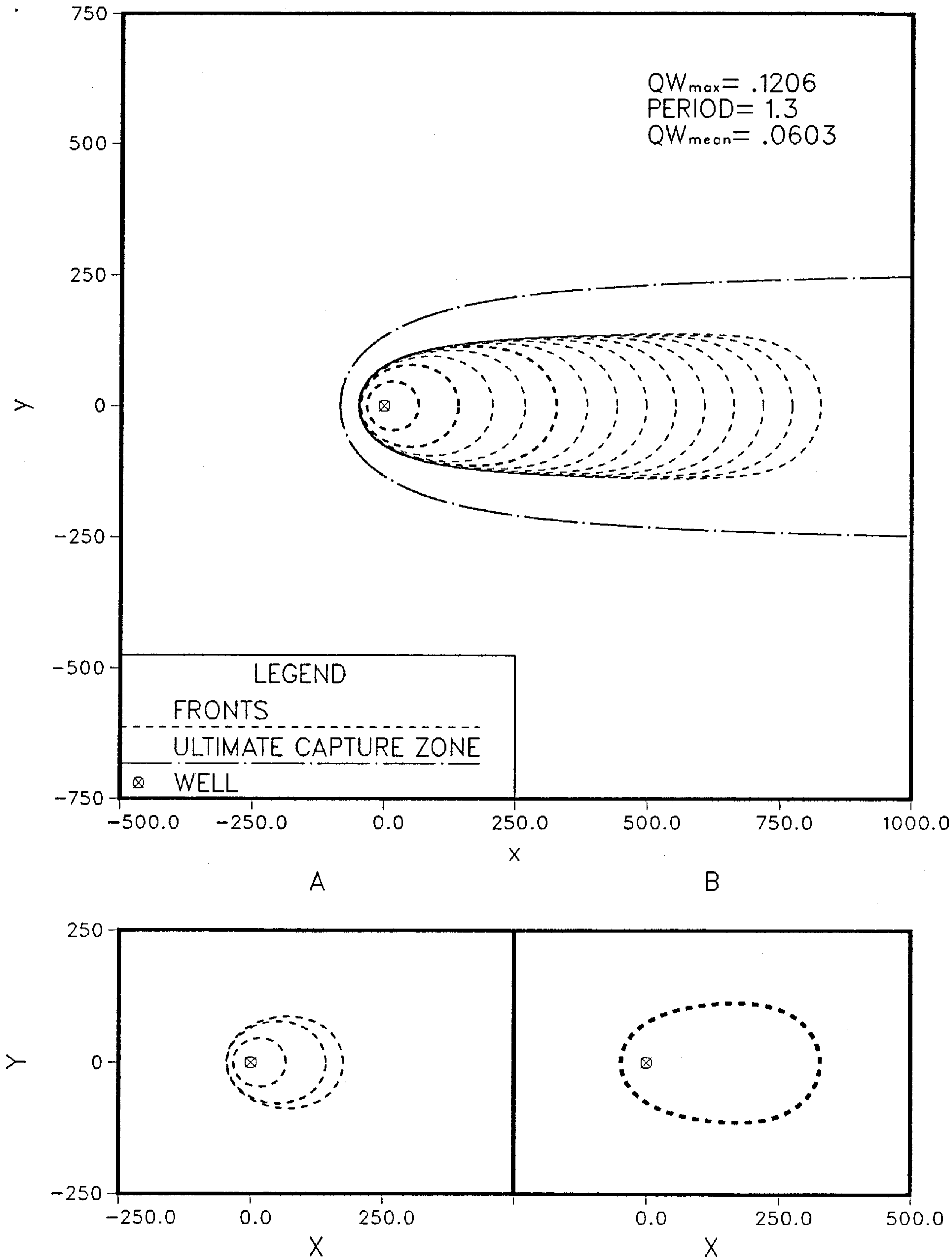

71

G64




PUMPING SCHEDULE

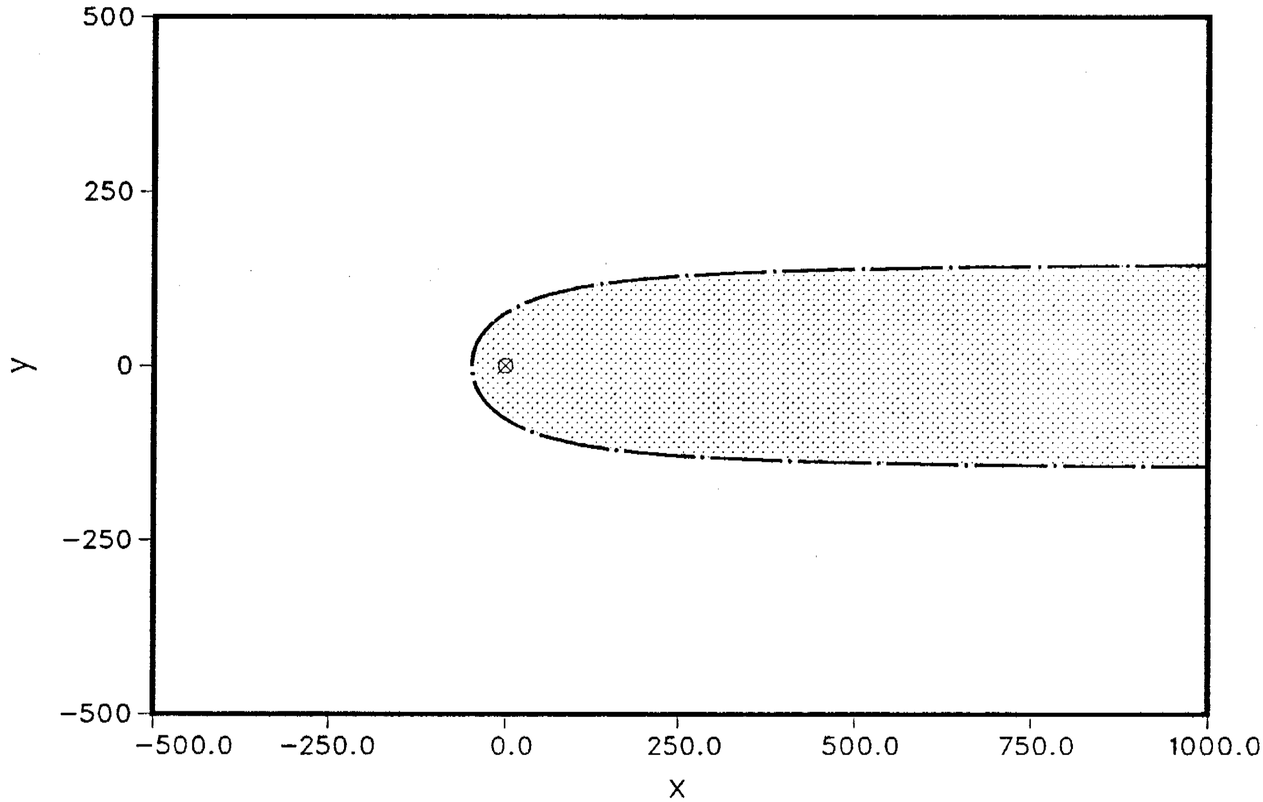




QWmax= .1206
PERIOD= 1.3
QWmean= .0603
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL
750
500
250
0
-250
-500
-750
Y
-500.0
-250.0
0.0
250.0
500.0
750.0
1000.0
X
A
B
250
0
-250
Y
-250.0
0.0
250.0
0.0
250.0
500.0
X
X

G C C

PROBABILITY ZONES WITH RESPECT
TO THE MEAN PUMPING RATE UCZ




75 C

Therefore, the variations between simulations will be only a function of the time the flow field is ~~allowed~~ subject to ambient flow alone. Previous simulations have given a great deal of information that can be used in this section to define the dynamics of transients in general. Some important conclusions that can be derived from the previous section are:

(1) The effective stagnation point is the best predictor for the 0% boundary

(2) There are three basic patterns that the fronts may simulate

  (a) Fully connected - consecutive period capture zones never cross

  (b) Intermediately connected - consecutive PCZs share a portion of their boundary (hummocky patterns)

  (c) Disconnected - consecutive PCZs are completely disconnected

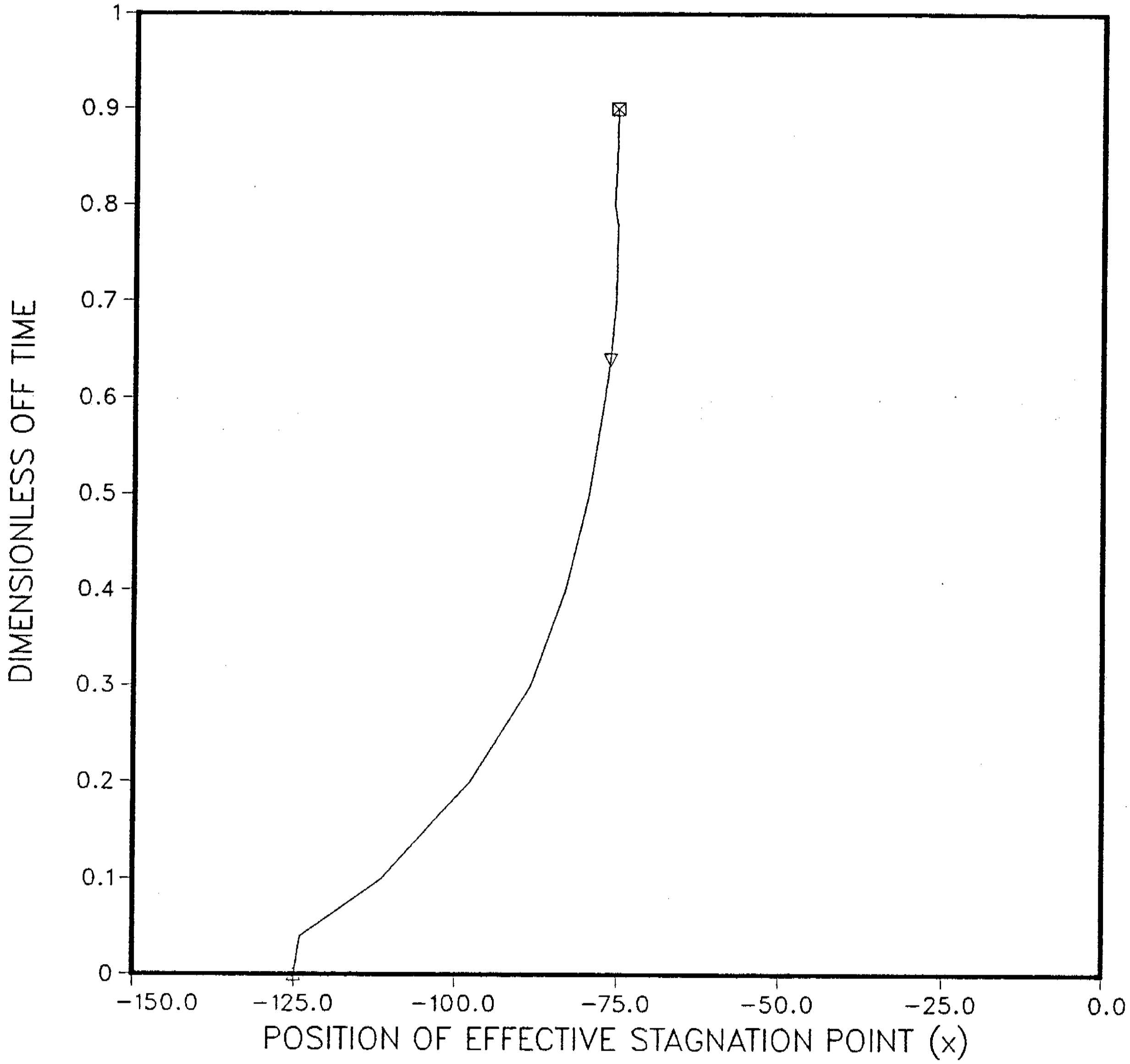
?

A close inspection of the the first two parts of the above outline suggests that the front pattern is a function of the effective stagnation point. This phenomena is examined using the constant amplitude simulations, keeping the pumping rate at 0.157 and the time on $t_P = 0.5\tau_{dim}$. The off time is the only parameter allowed to vary (actually the period). These simulations are used to test the conclusion that the front pattern is a function of the effective stagnation point. (?)

Figure 23 shows how the position (on the axis y= 0) of the effective stagnation point changes as the off time changes. This curve is generated by tracking a particle from the well on the y-axis backward in ? time using a very small length constant (this increases the accuracy) for a large number of period durations, determining the maximum distance the particle has moved from the well, and plotting them as a function of off time. Two points on the curve are very important. The first is the time when the particle does~~n't~~ not move beyond the maximum extent of the first pumping period (x= -76.3). This point is labeled by the triangle on the graph. The second is the time when there ceases to be an effective stagnation point. This is represented by the crossed-box on the figure.

wtf is this period !!!

When the duration of the period is such that it creates an effective stagnation point at the maximum extent of the first period capture zone, the flow of water becomes impeded along the exterior flow



1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
DIMENSIONLESS OFF TIME
−150.0
−125.0
−100.0
−75.0
−50.0
−25.0
0.0
POSITION OF EFFECTIVE STAGNATION POINT (x)

paths. The particles no longer have smooth transitions between pumping periods, they travel along

short on-time hummocky paths. The pumping portion of the period induces these particles that are outside and up-gradient of the current PCZ to flow at angles near 45° toward the well. When the well is off they move with the ambient flow. This type of front only exists if the most exterior particles that may reach the

? well can travel around the position of the the second period capture zone at $t=\tau_P$. For the square wave with a minimum amplitude of 0, this is the position of the first PCZ translated (see figure 11E). The fluid must also not be allowed to pass around the extent of the first PCZ at t= 0. This motion also creates lower extraction probabilities.

there is no 11E

For figure 23 the off time that corresponds to the point on the first PCZ at t= 0 is .64. Figures 24A-D illustrate a simulation that models a condition when the flow field is influenced by a pumping rate of 0.157 for a $t_P$ = 0.5 and an ambient flow duration of 0.64, when the effective stagnation point is on the first PCZ. In figure 24A the period capture zones intersect at the maximum/minimum y-coordinates. The front pattern demonstrates a smooth transition (figure 24B). The 100% boundary passes through every period capture zone intersection (figure 24C). The transect probability is not shown.

24A - period: 1.14

24B period: 1.1

24C.?

not even on Fig. 23

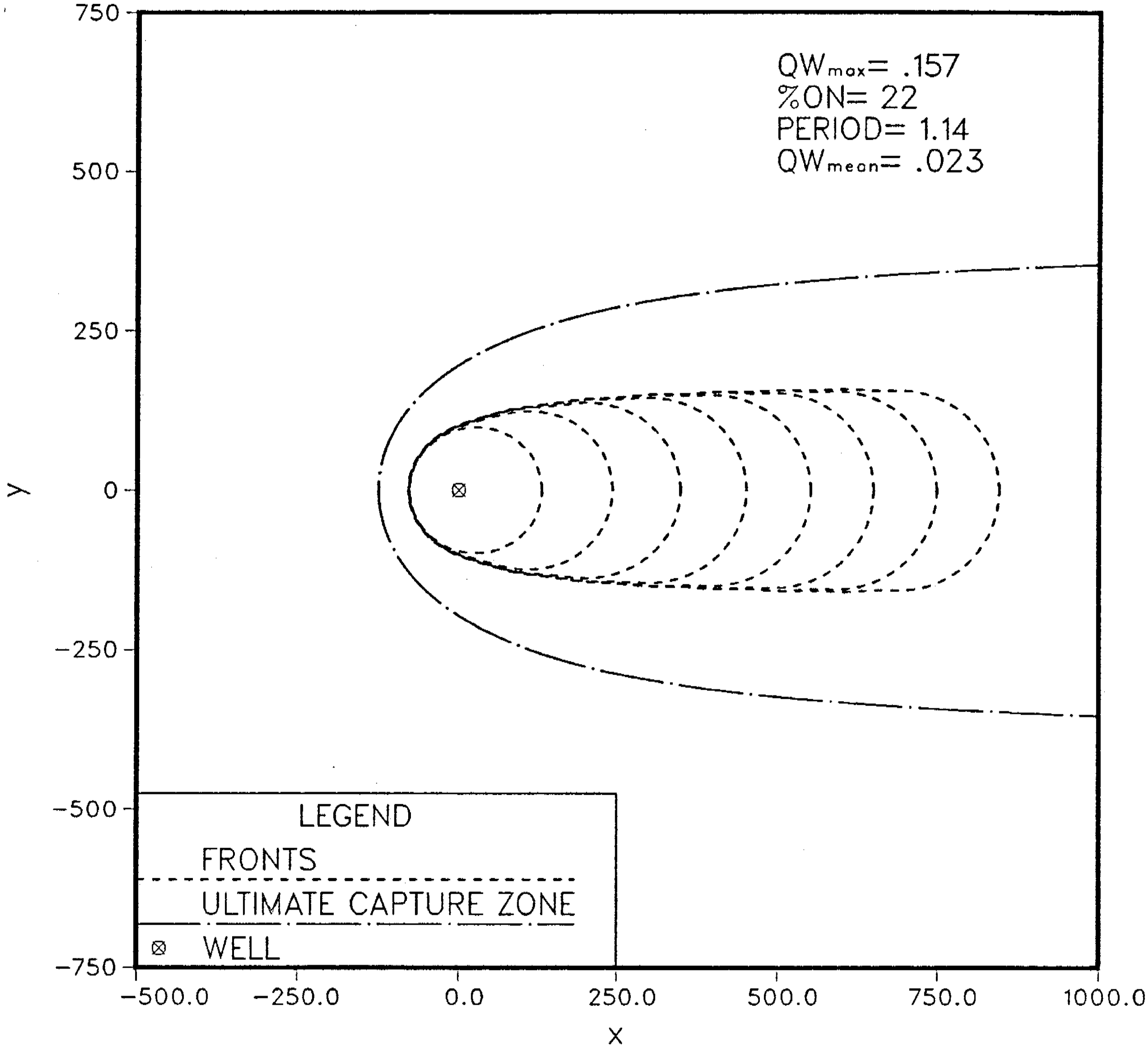
In the next simulation the off time is adjusted so it is slightly larger than the previous simulation (period= 1.24). The period capture zones are shown in figure 25A. Notice that they no longer intersect at their maximum/minimum y-coordinates. Figure 25B illustrates that the front pattern is beginning to exhibit a hummocky nature. The comparison between these two simulations support the theory that there is a major change in flow behavior where the curve in figure 23 predicted. whatever that was!

what is the significance of this

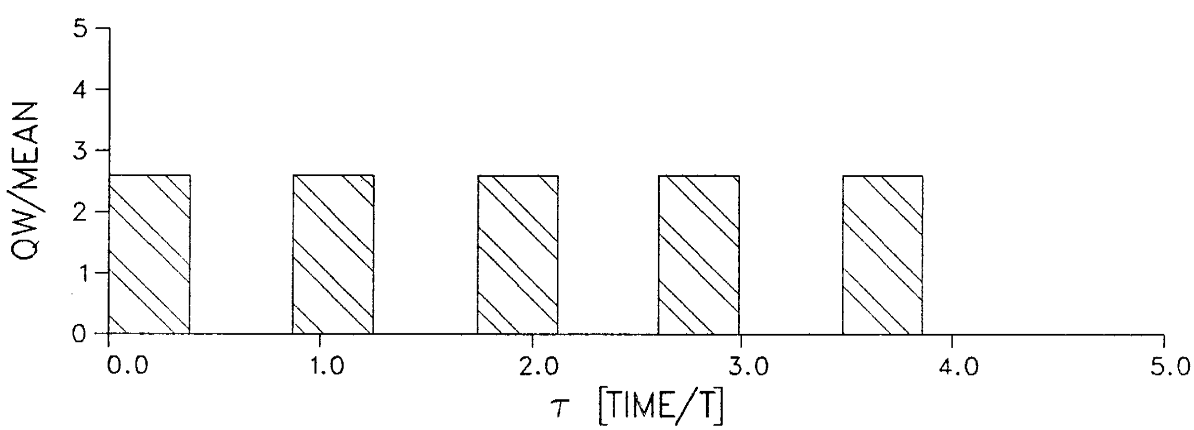
The effective stagnation point doesn't exist past the point discussed above, due to the fact that under these conditions it passes the well and travels up-gradient to infinity (when back tracking). This is significant, for just that reason. Up until this point there is 100% boundary, after, the fronts become disconnected, and the fronts only define regions that have less than 100% chance of entering the well. This occurs as a result of the exterior particles in the flow field not being able to pass the maximum/minimum reach of the the position of the second PCZ at $t=\tau_P$. This further decreases the extraction probabilities in the region. — not well defined

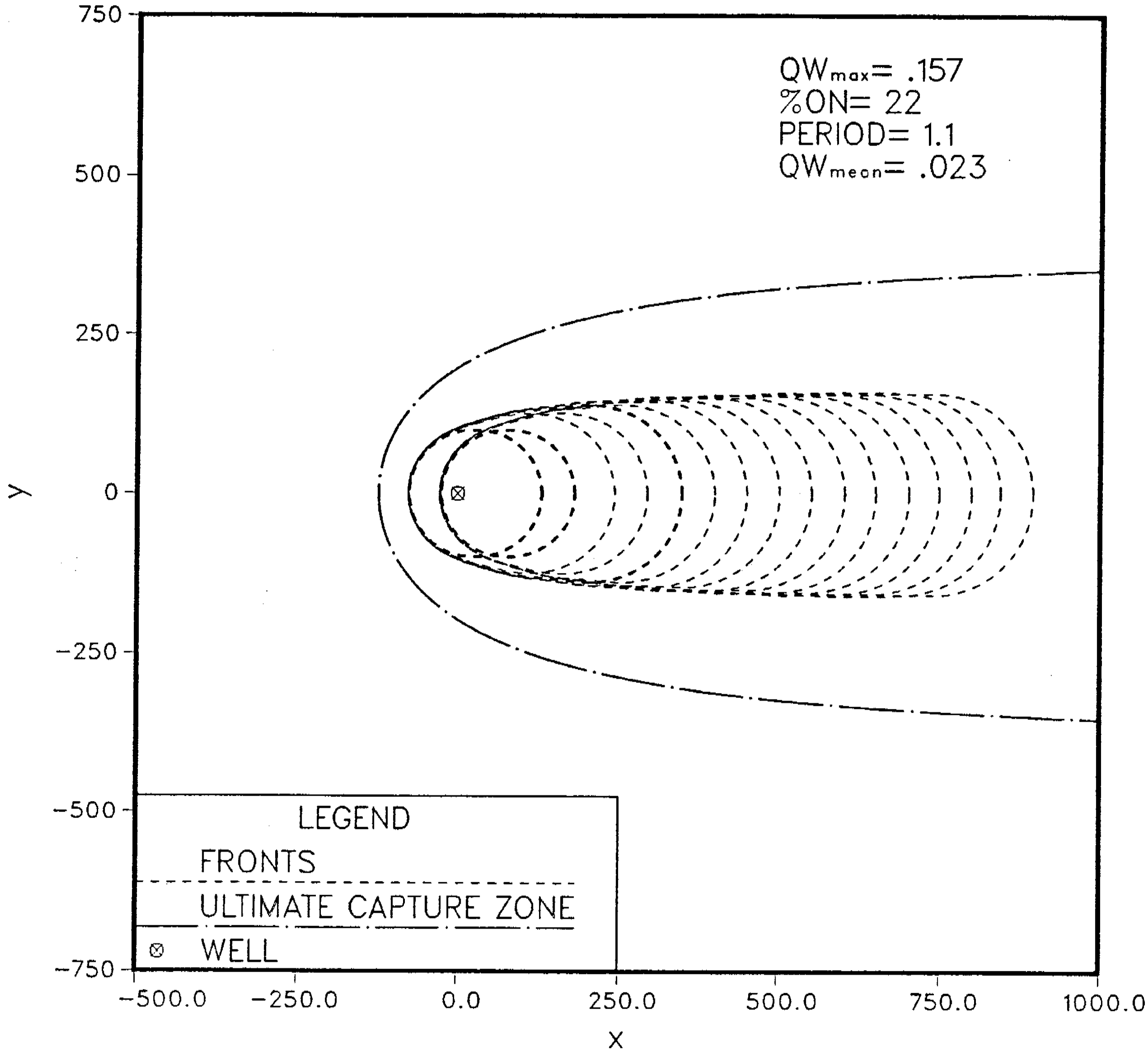
?

wtf

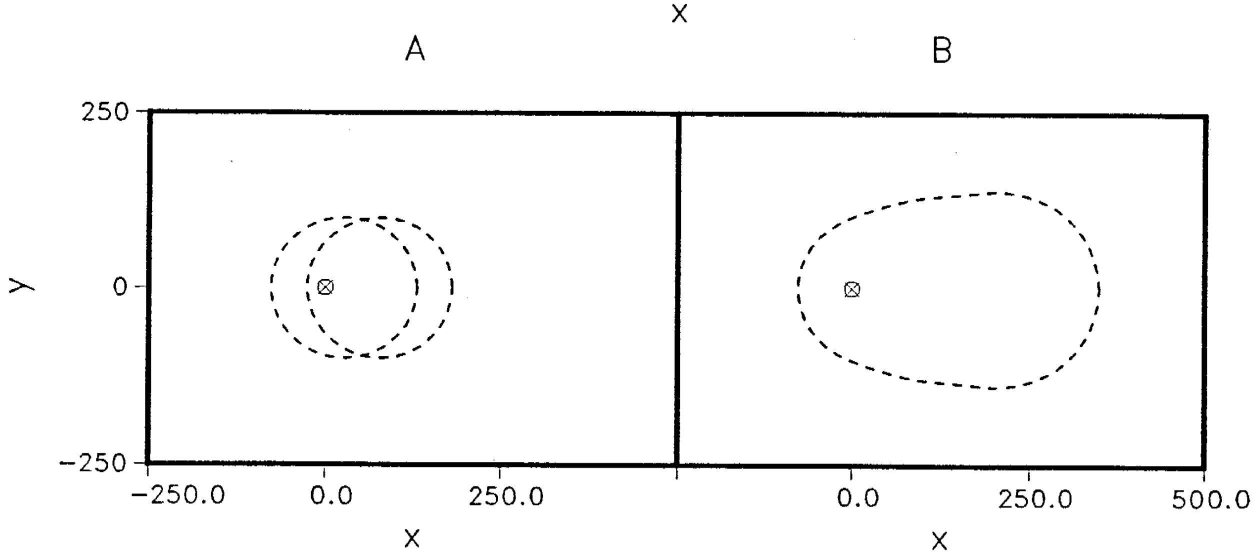



PUMPING SCHEDULE



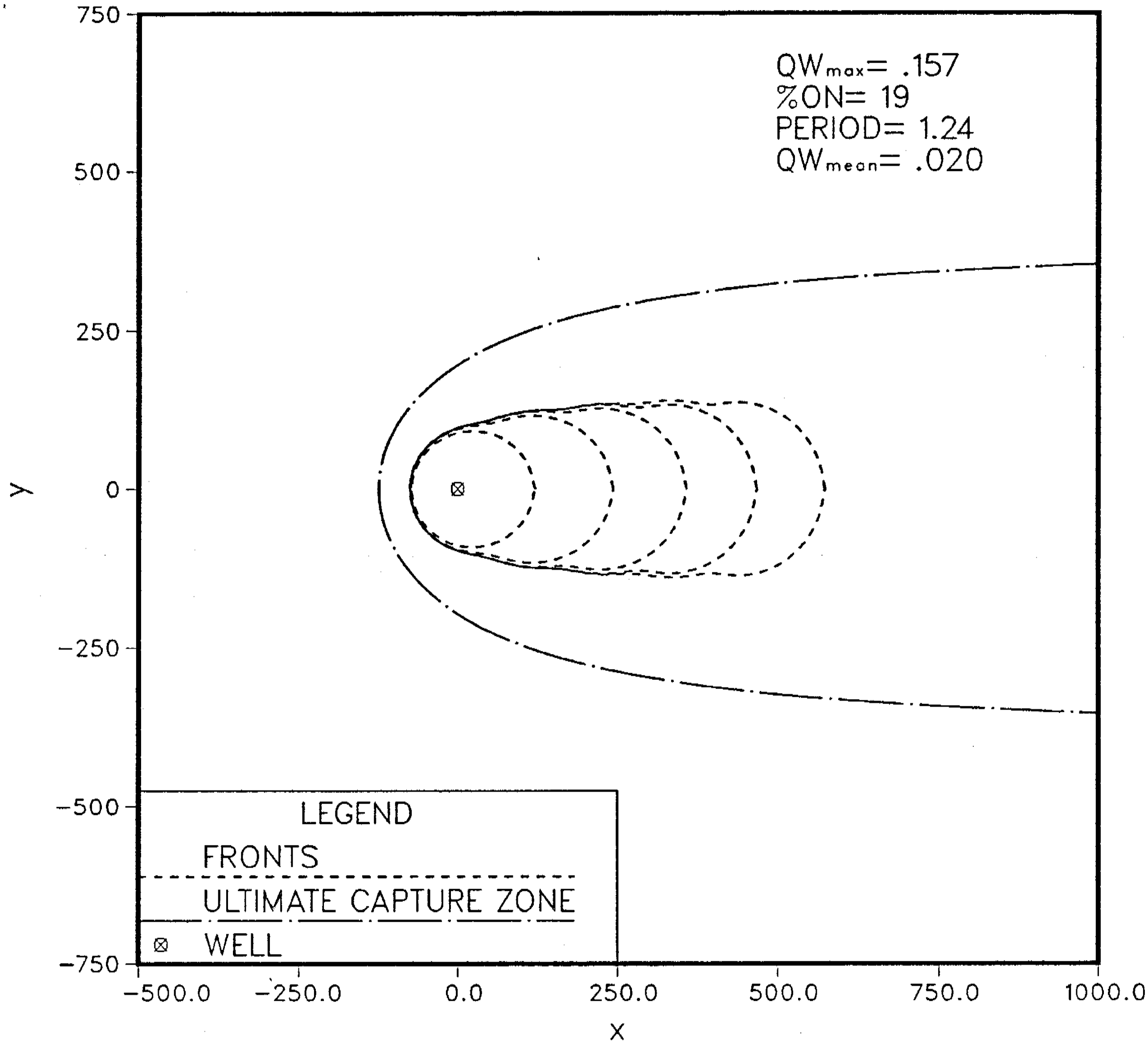


QWmax= .157
%ON= 22
PERIOD= 1.1
QWmean= .023
750
500
250
0
-250
-500
-750
y
-500.0
-250.0
0.0
250.0
500.0
750.0
1000.0
x
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL

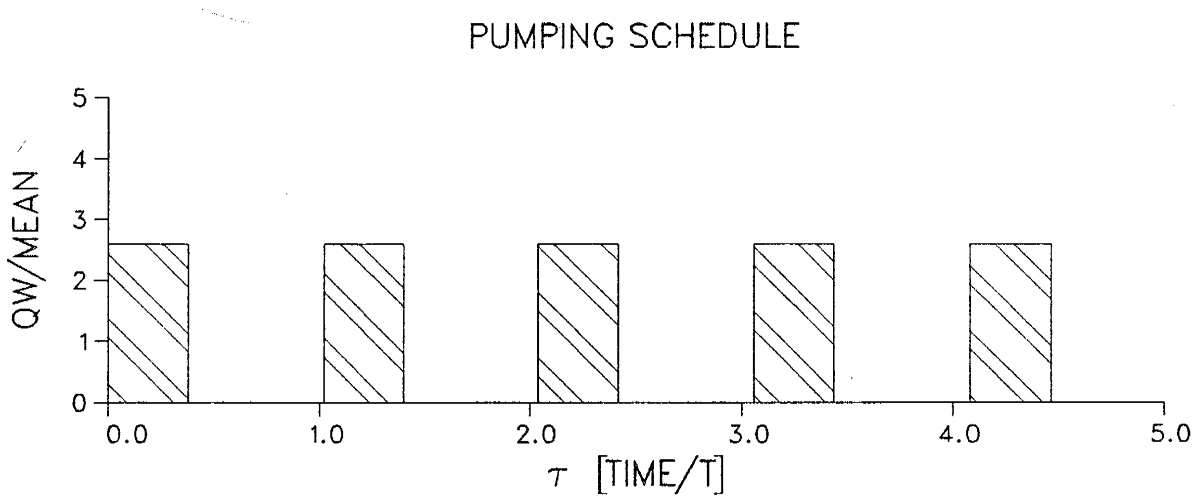



A
B
250
0
-250
y
-250.0
0.0
250.0
x
0.0
250.0
500.0
x

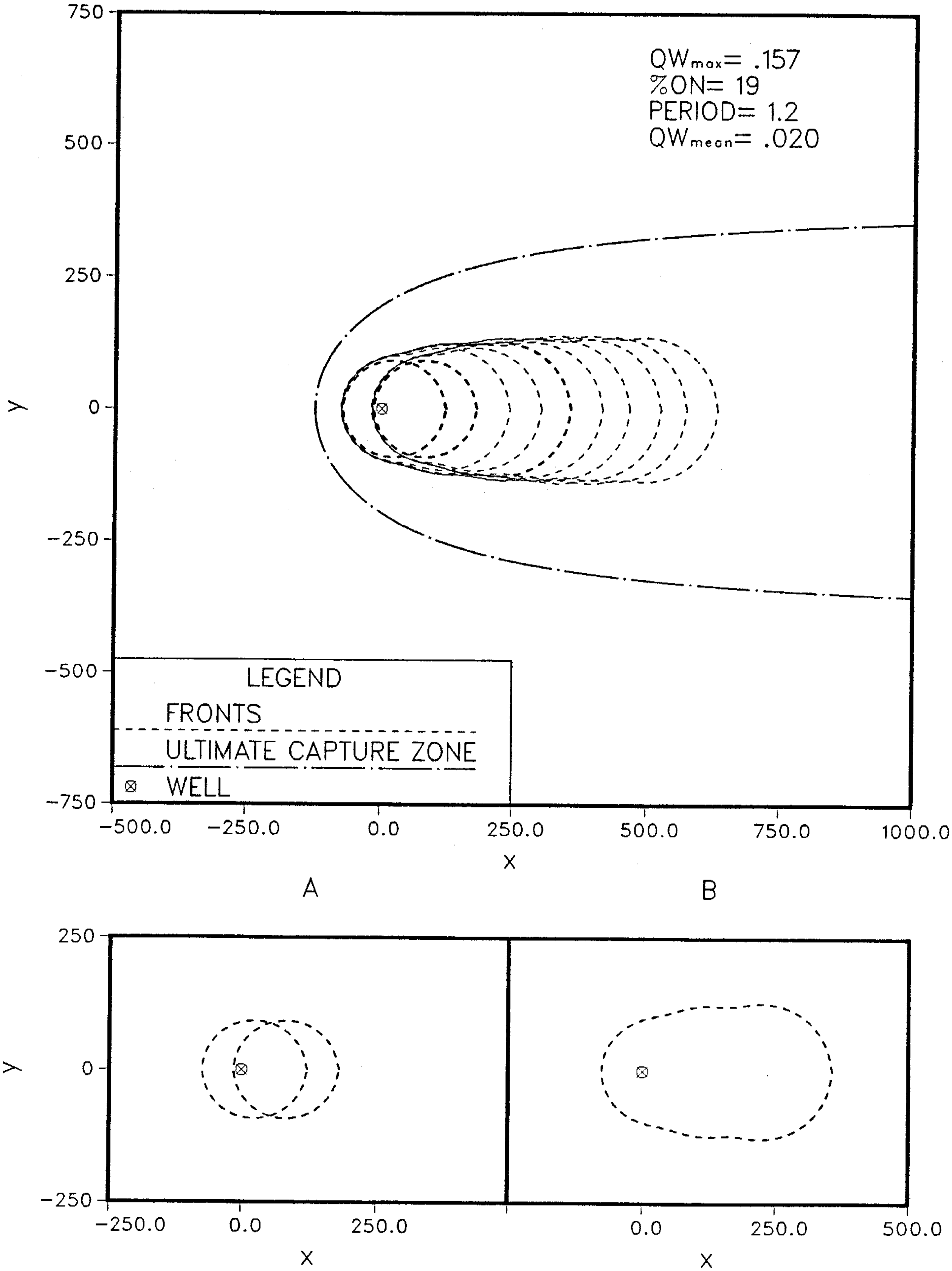

24



$QW_{max}$= .157
%ON= 19
PERIOD= 1.24
$QW_{mean}$= .020
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL
y
x
750
500
250
0
−250
−500
−750
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0

PUMPING SCHEDULE
QW/MEAN
5
4
3
2
1
0
0.0
1.0
2.0
3.0
4.0
5.0
$\tau$ [TIME/T]



QWmax= .157
%ON= 19
PERIOD= 1.2
QWmean= .020
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL
750
500
250
0
-250
-500
-750
y
-500.0
-250.0
0.0
250.0
500.0
750.0
1000.0
X
A
B
250
0
-250
y
-250.0
0.0
250.0
X
0.0
250.0
500.0
X

of what!!

The value from figure 23 for the maximum off time when an effective stagnation point still exists (time off= 9.6$\tau_{dim}$) is utilized to simulate this limiting condition. Figure 26A demonstrates the position of the period capture zones at t= 0. Notice the cusps between the PCZs. This is a relict of the hummocky patterns. Figure 26B demonstrates that there is a triple point intersection where the first and second PCZs meet and also the position of the second PCZ at t= $\tau_P$. The front pattern is disconnected, and is crescent shaped. In figure 26C the transect probabilities, along with the 0% boundary are illustrated. The probability of extraction is never 1., but comes very close.

.96$\tau_{dim}$?

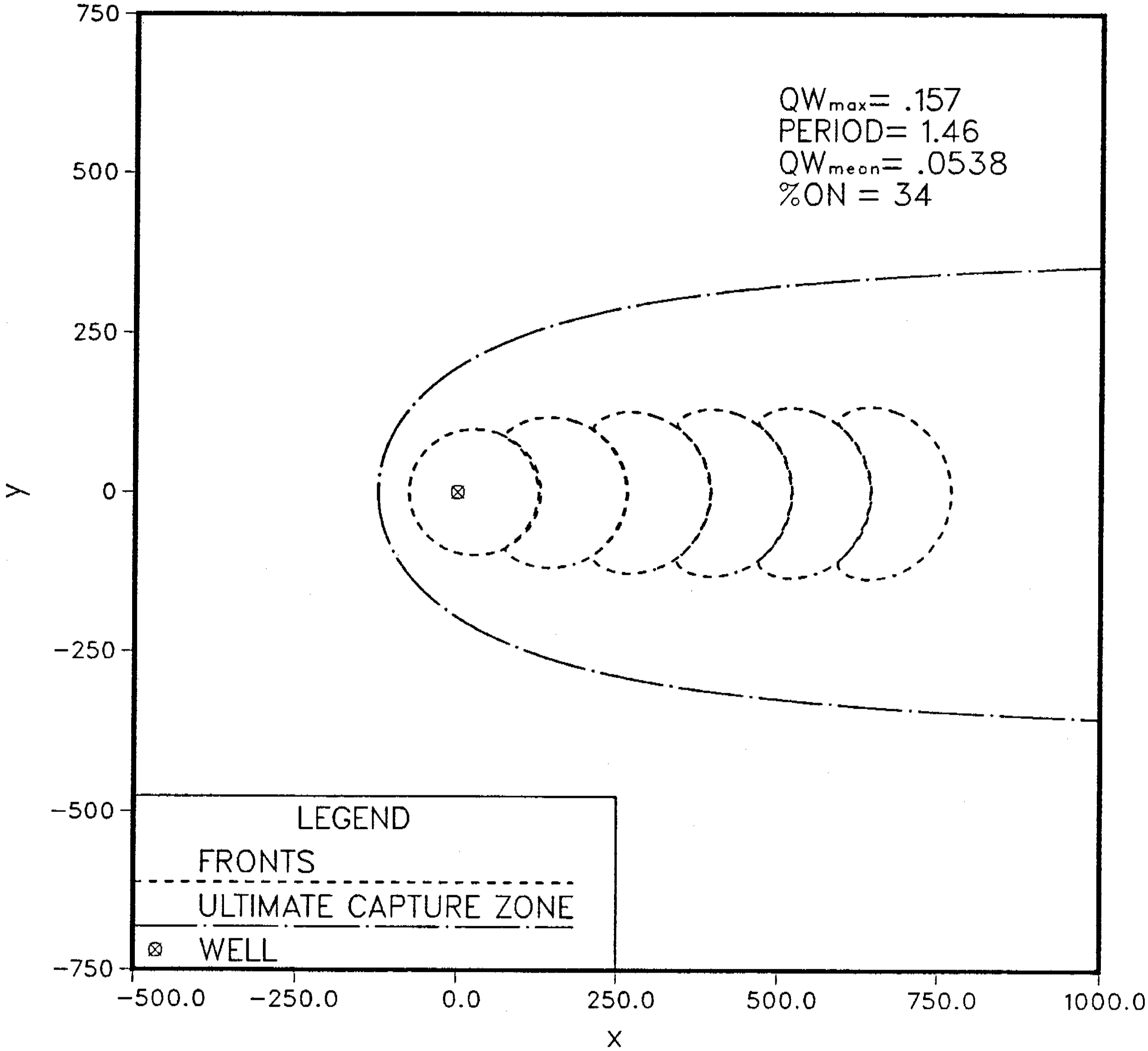
Figure 27A are the period capture zones for an off time of 1.4. They are rounded and the disconnection is much more prevalent than with any previous simulation. The transect probabilities (figure 27C) exhibit the step effect seen in figure 18C. The passed zone is almost horizontal from the maximum/minimum y-coordinate of the first period capture zone. This suggests almost no transverse movement of water away from the well (as expected).

The final simulation in this section is one where the ambient flow rate almost completely dominates. The period is 4.0. Figure 28A illustrates the positions of the period capture zones. They are separated by large distances and no longer have a crescent shape, rather they are spher~~ical~~oidal. The transect probabilities in figure 28B demonstrate the low probabilities throughout the flow field, never reaching much above 50%.

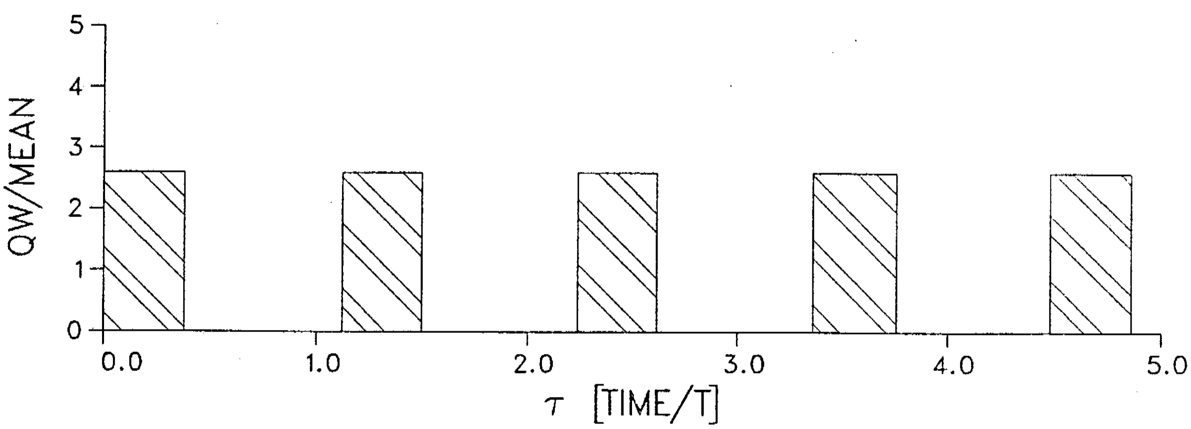
The probability for a particle to reach the well from a given position is a function of the transient conditions in the flow field. They are (what 'are') dependent on the duration of pumping, the amplitude, and the duration and magnitude of ambient flow. It can be shown that the probability density function for a particle reaching the well as t approaches ∞ is reflected by the transient cycle inducing flow. That is, the probability density function for a particle input into the passed zone when a square wave pumping schedule is implemented is also a square wave. The probability therefore, could be derived by superposing the cumulative distribution function for the ambient flow period normalized by (1-% on) ? with the cumulative distribution function of the pumped period normalized by the (% on) and return the probability that a particle input at that position will reach the well. This could reduce the cost of generating these

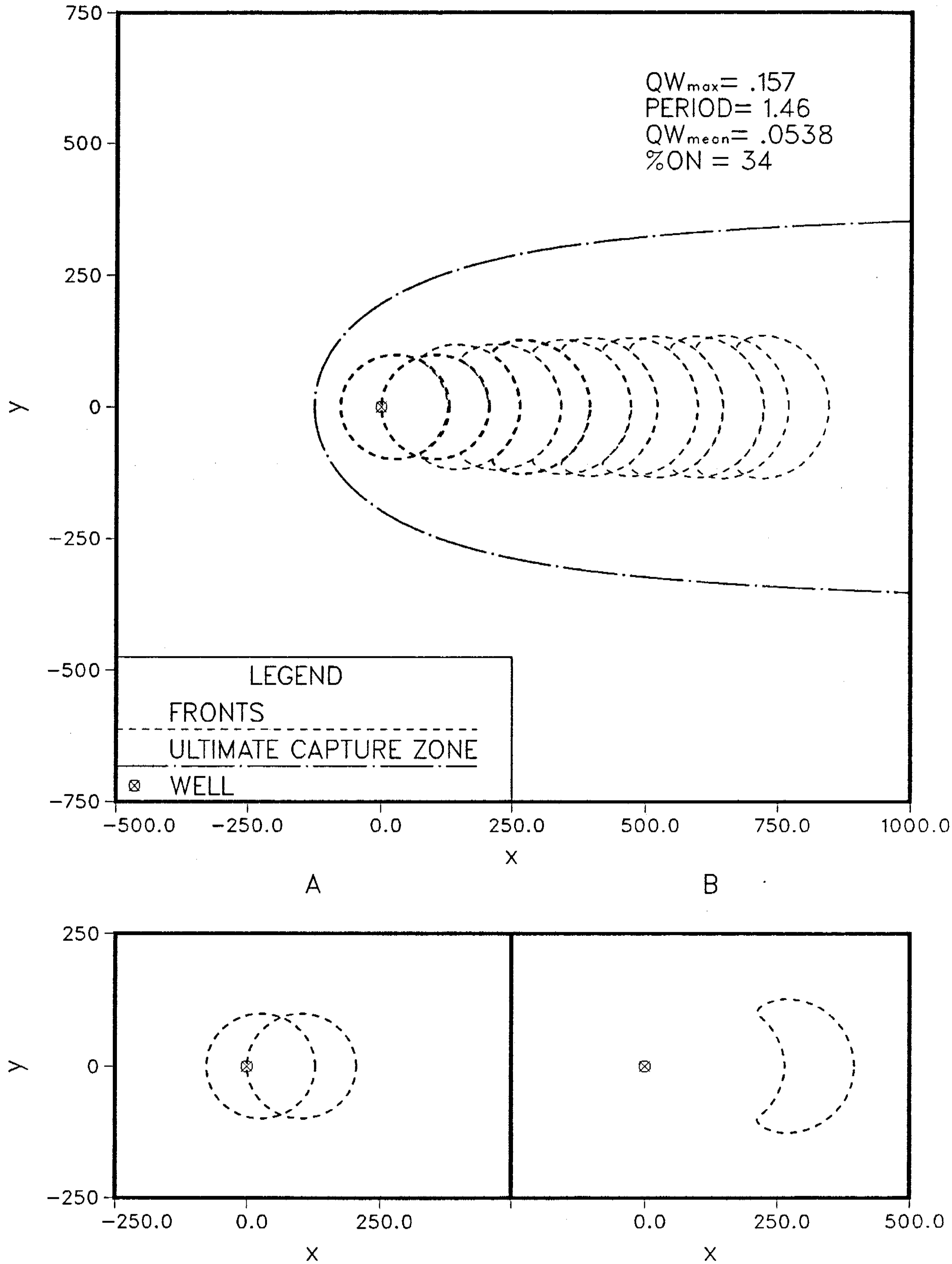
← I think this is for particle input at a point over time !?

? wtf

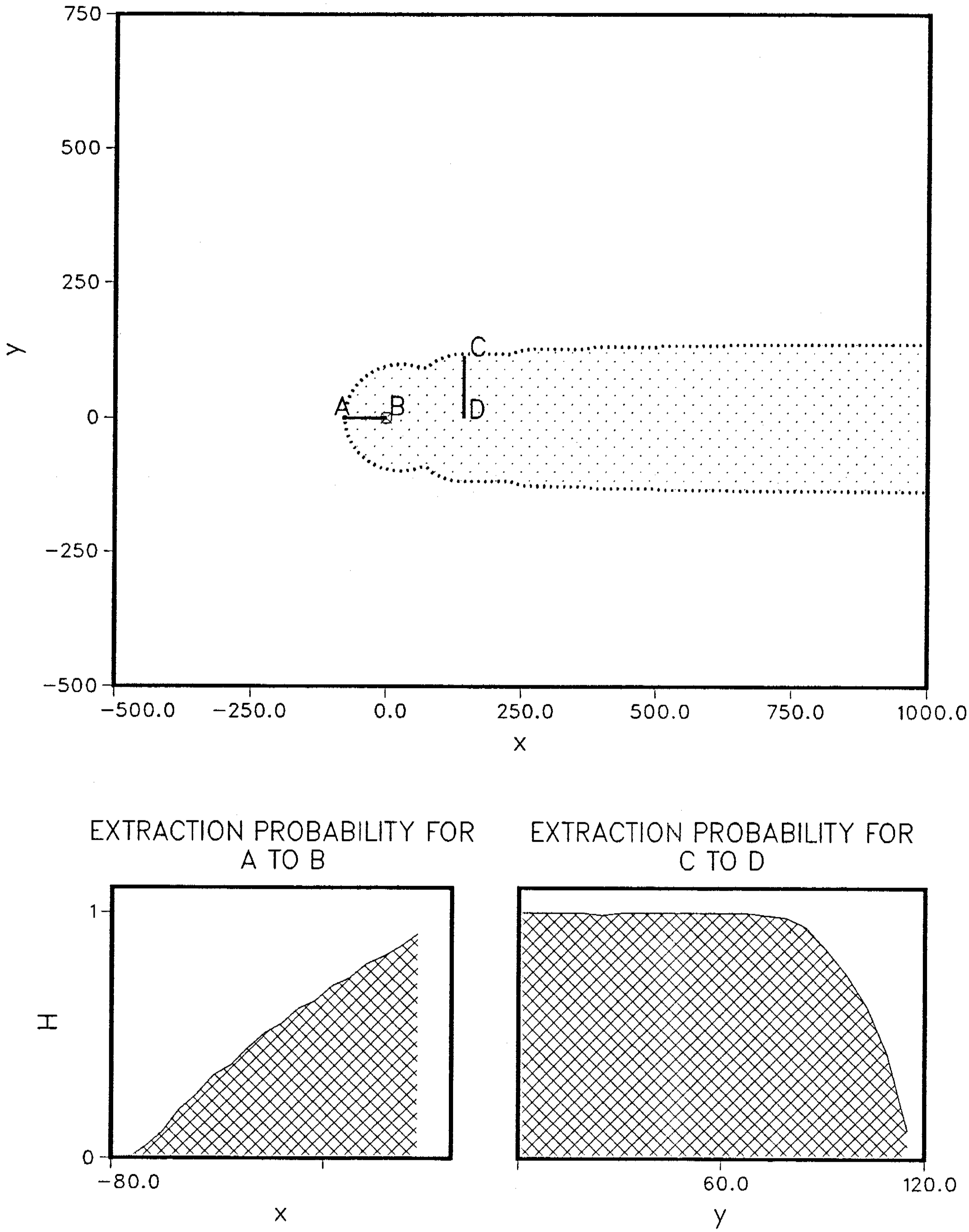


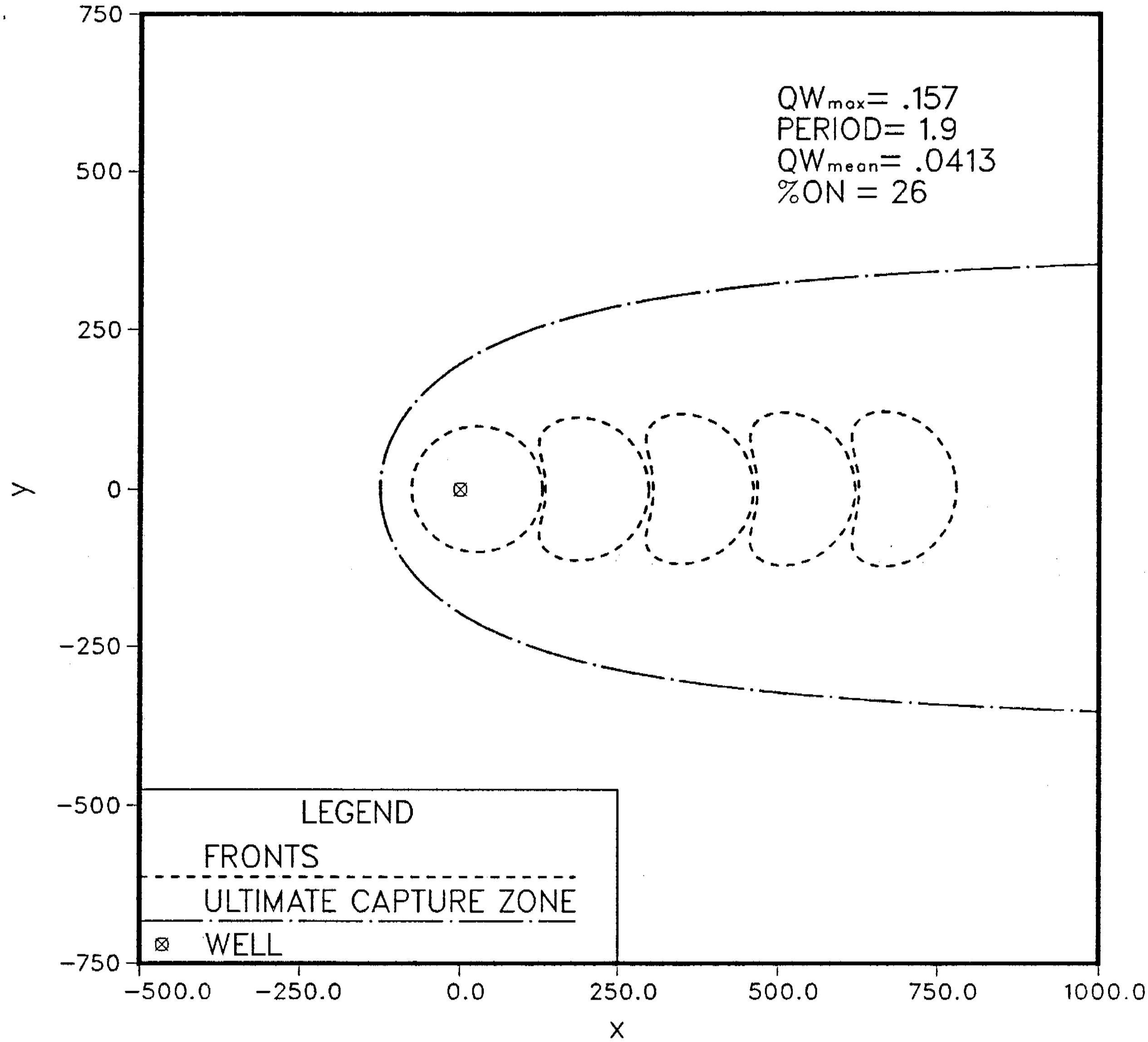

PUMPING SCHEDULE



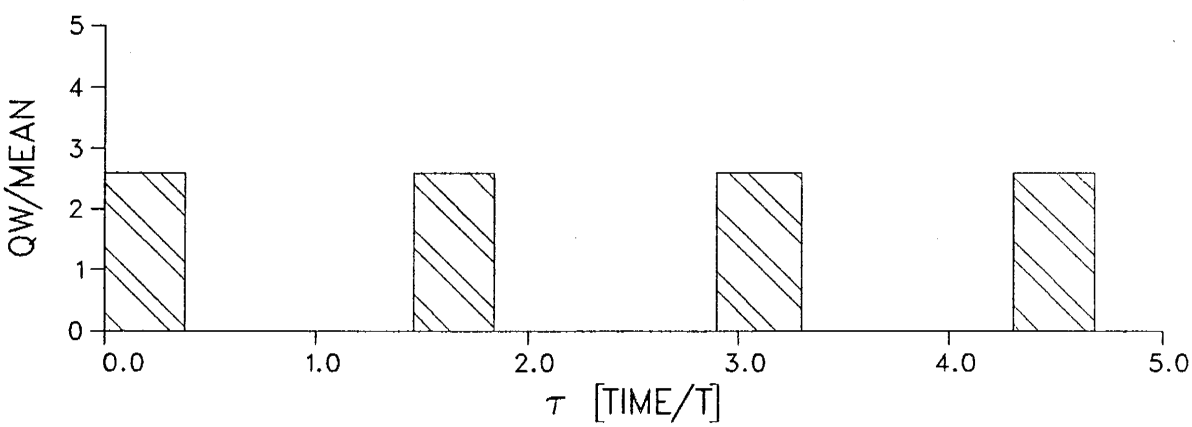


QWmax= .157
PERIOD= 1.46
QWmean= .0538
%ON = 34
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL
y
x
A
B

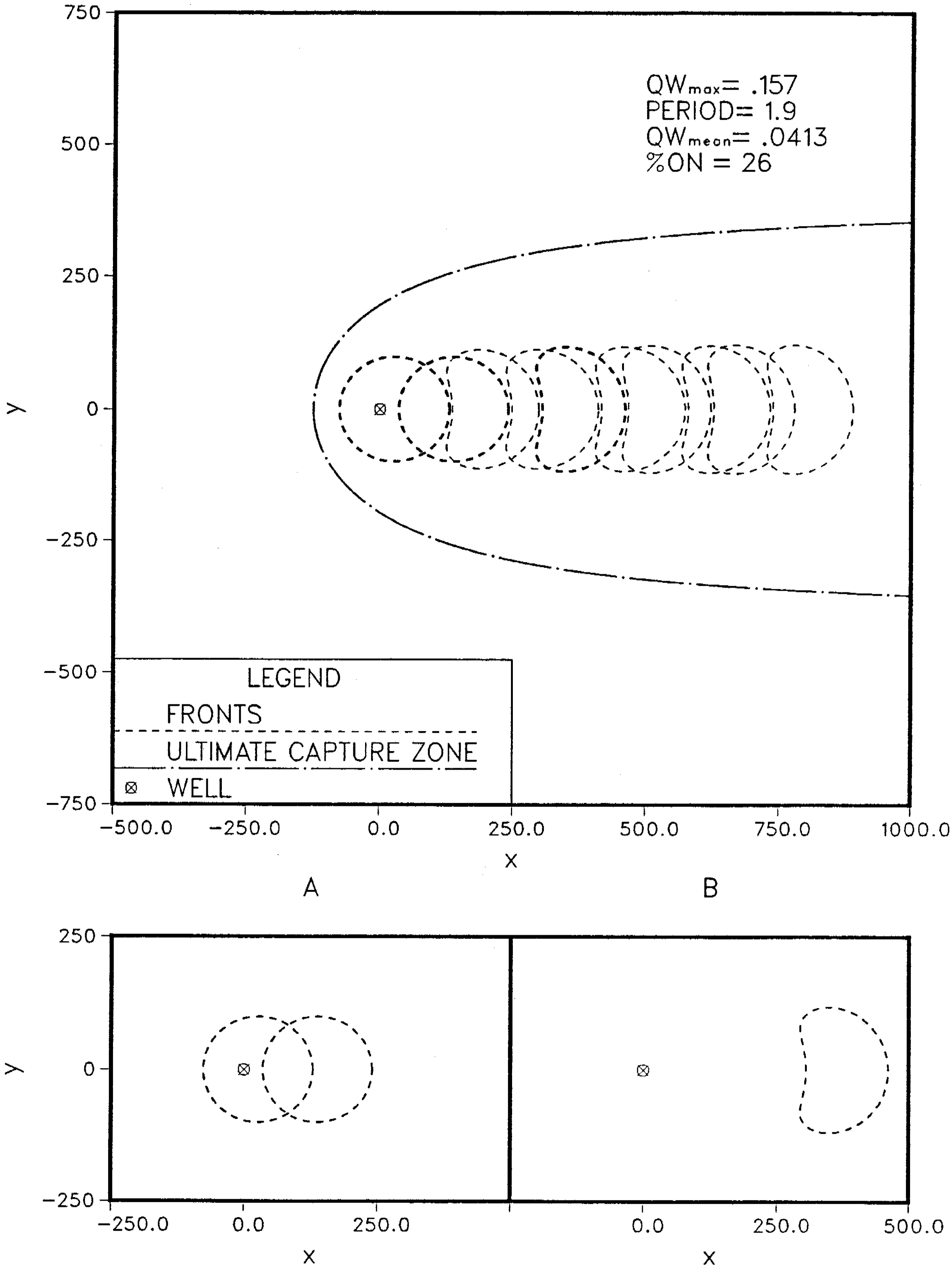

750
500
250
0
−250
−500
y
C
D
A
B
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x
EXTRACTION PROBABILITY FOR
A TO B
H
1
0
−80.0
x
EXTRACTION PROBABILITY FOR
C TO D
60.0
120.0
y



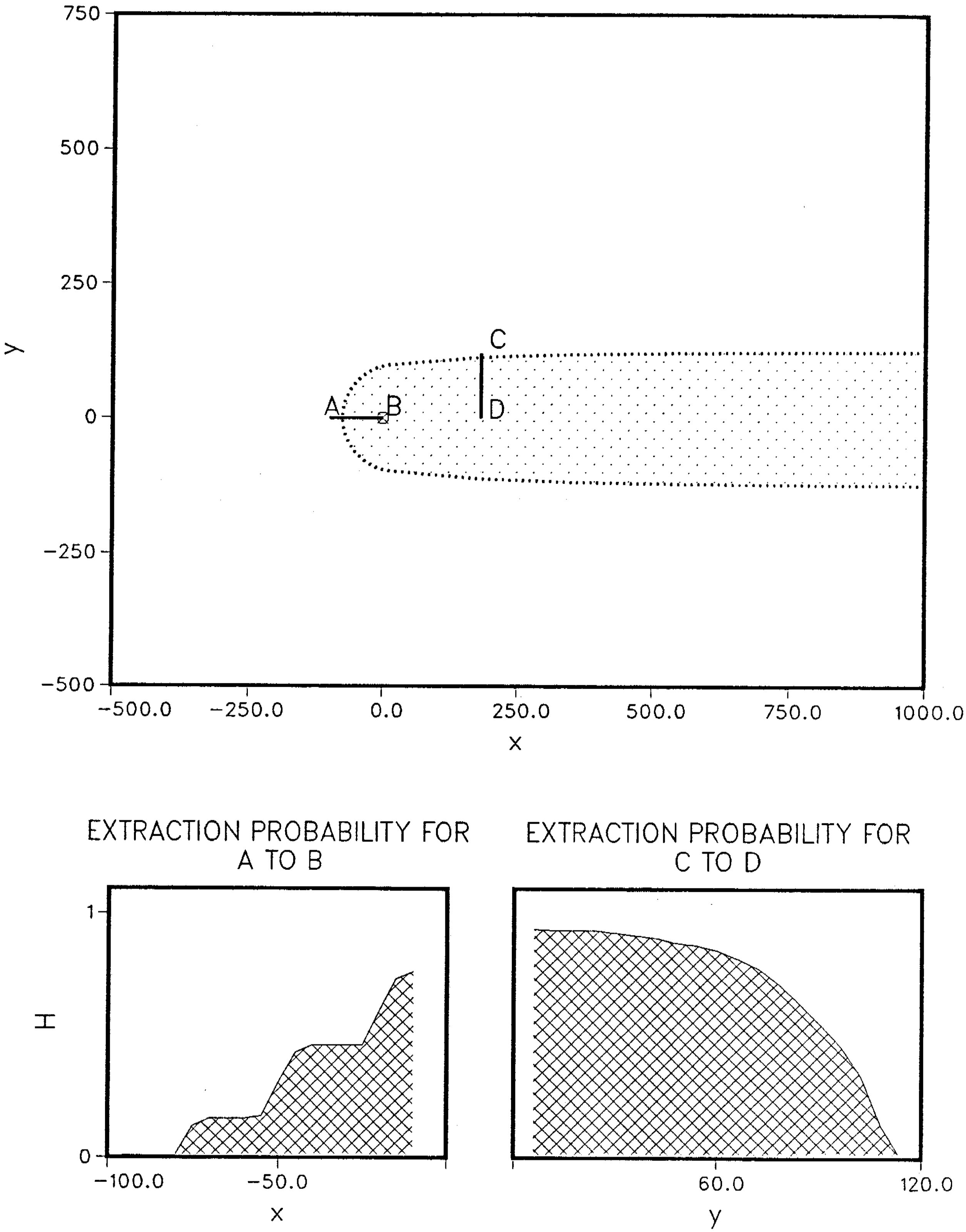

PUMPING SCHEDULE



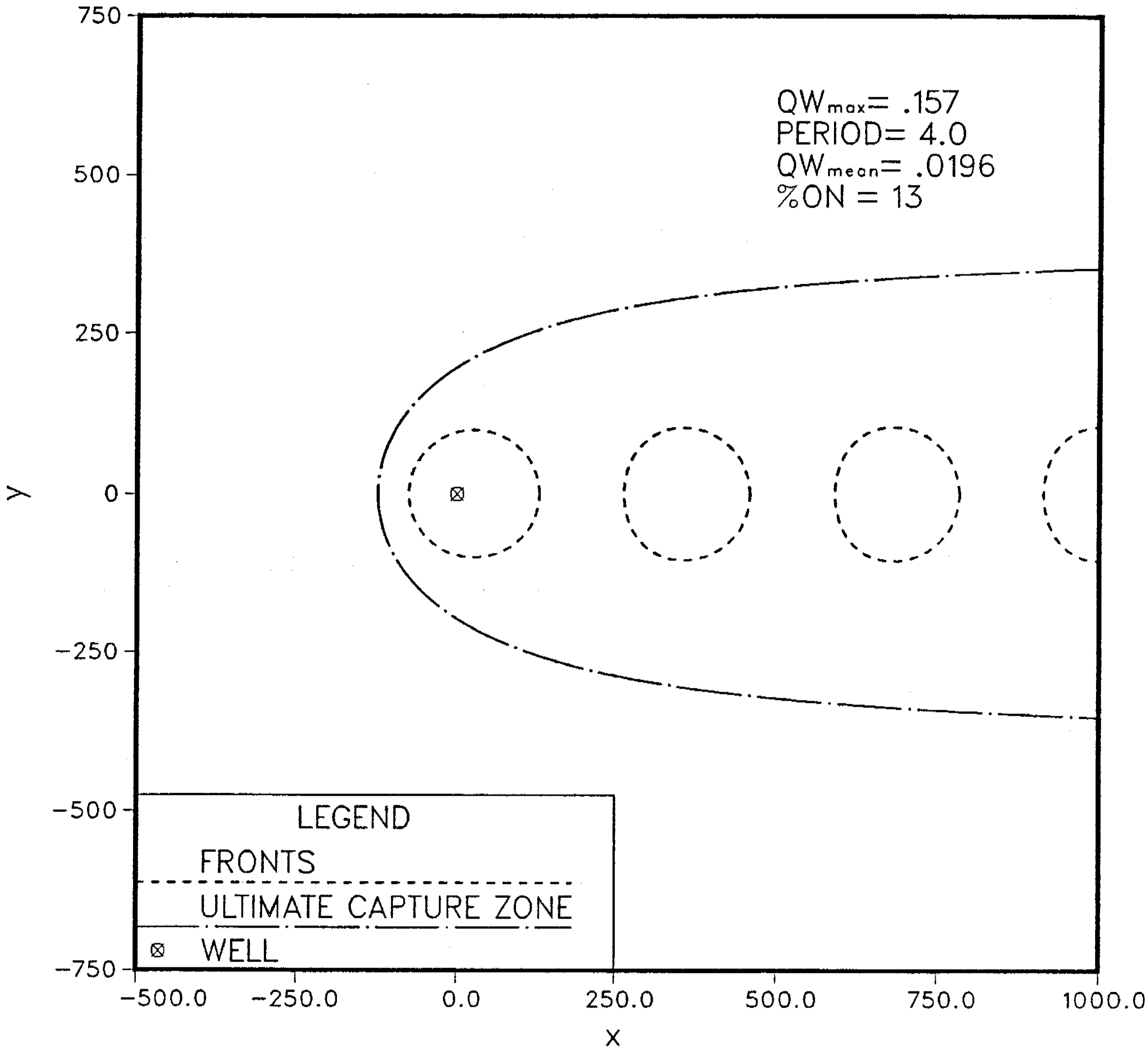

2.7A



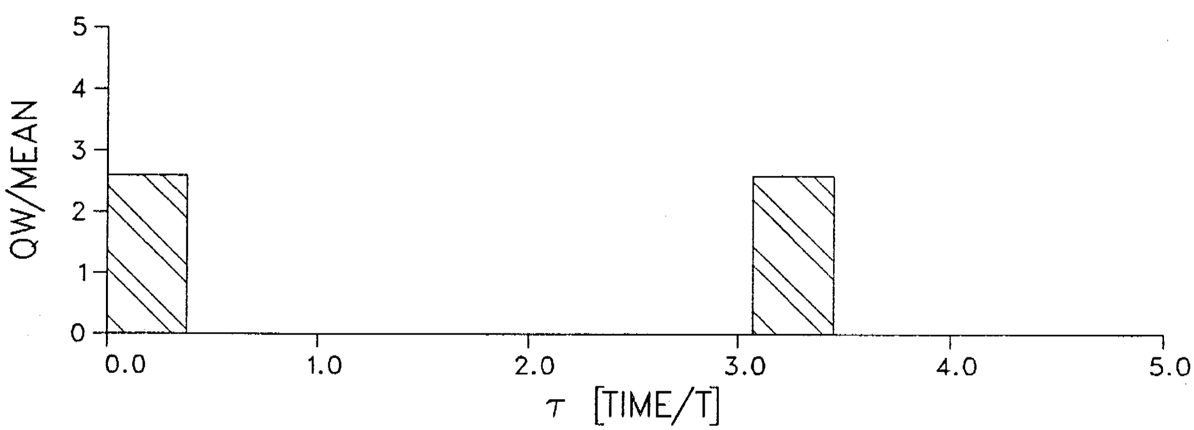
750
500
250
0
−250
−500
−750
y
$QW_{max}$= .157
PERIOD= 1.9
$QW_{mean}$= .0413
%ON = 26
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x
A
B
250
0
−250
y
−250.0
0.0
250.0
x
0.0
250.0
500.0
x

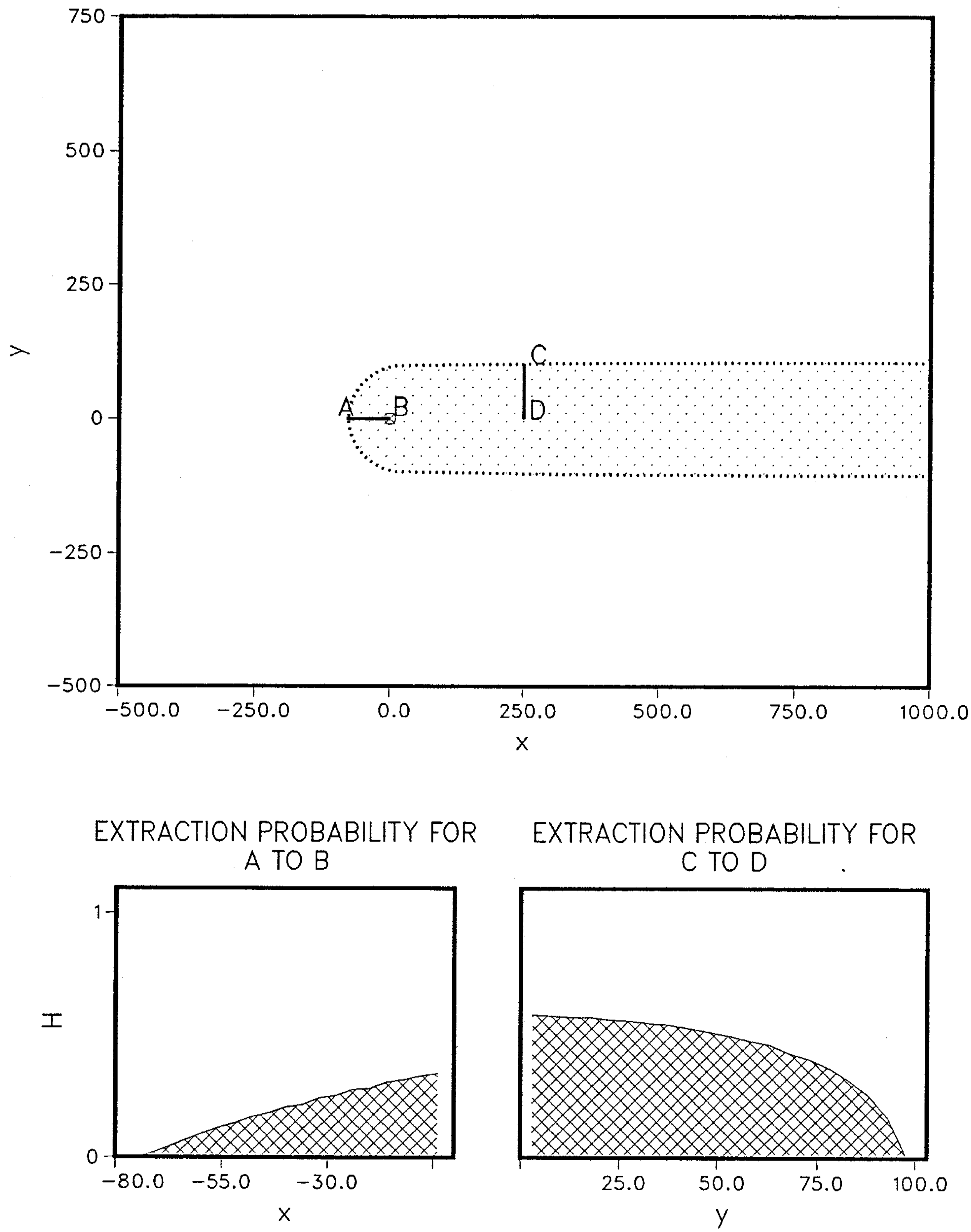


750
500
250
0
−250
−500
y
C
A
B
D
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x
EXTRACTION PROBABILITY FOR
A TO B
EXTRACTION PROBABILITY FOR
C TO D
1
0
H
−100.0
−50.0
x
60.0
120.0
y




PUMPING SCHEDULE





750
500
250
0
−250
−500
y
C
A
B
D
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x
EXTRACTION PROBABILITY FOR A TO B
EXTRACTION PROBABILITY FOR C TO D
1
0
H
−80.0
−55.0
−30.0
x
25.0
50.0
75.0
100.0
y

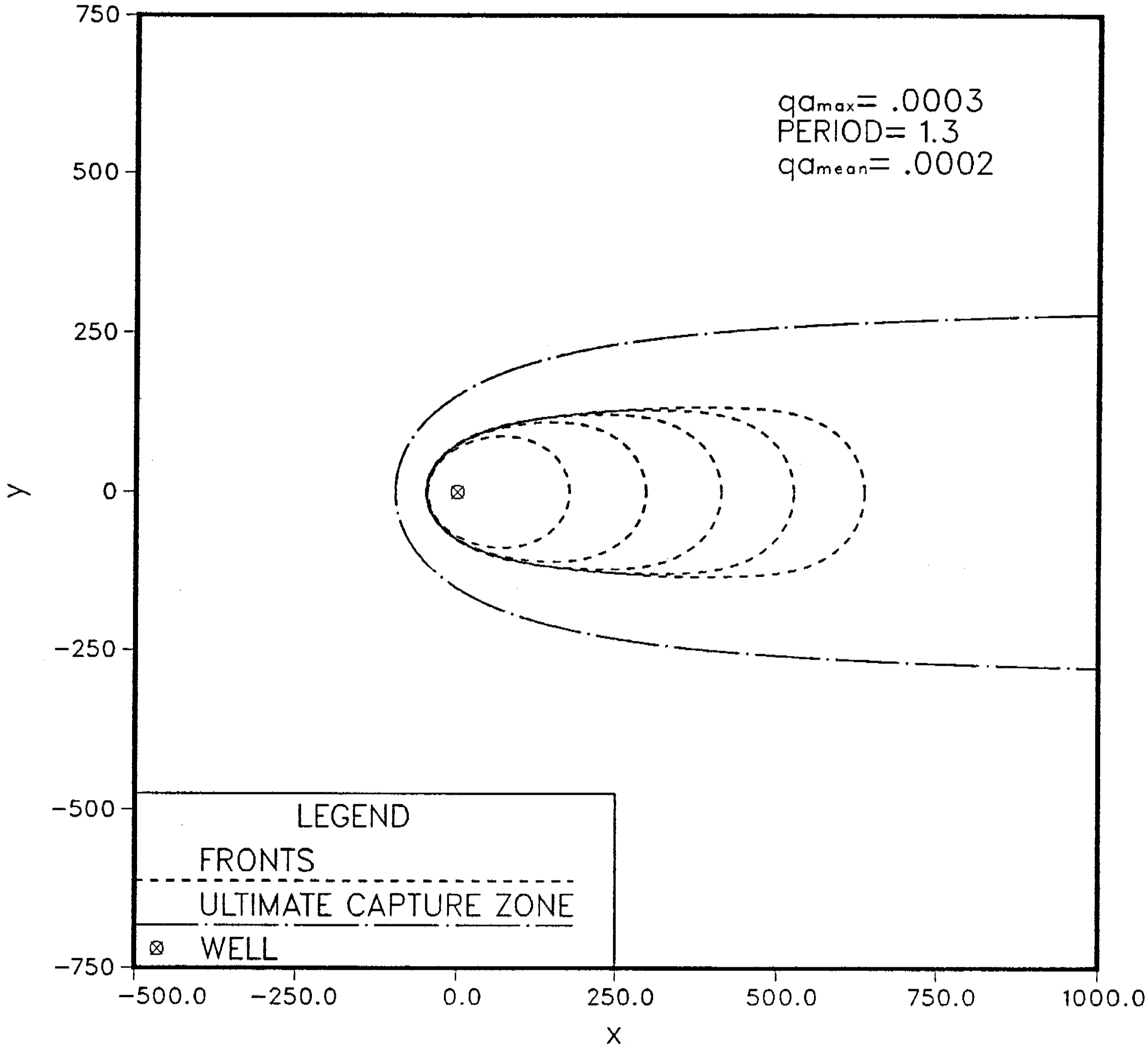

28B

transect probabilities.

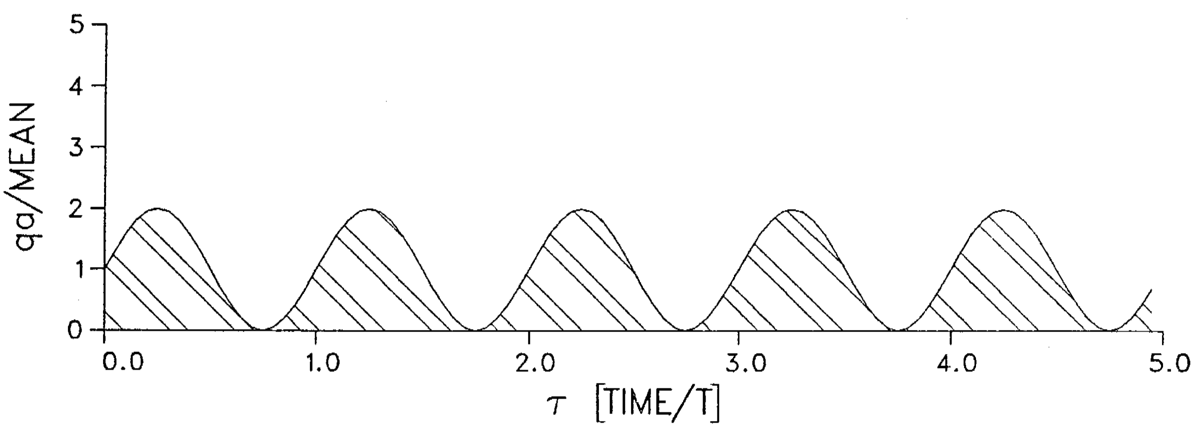
**Ambient Flow Fluctuations**

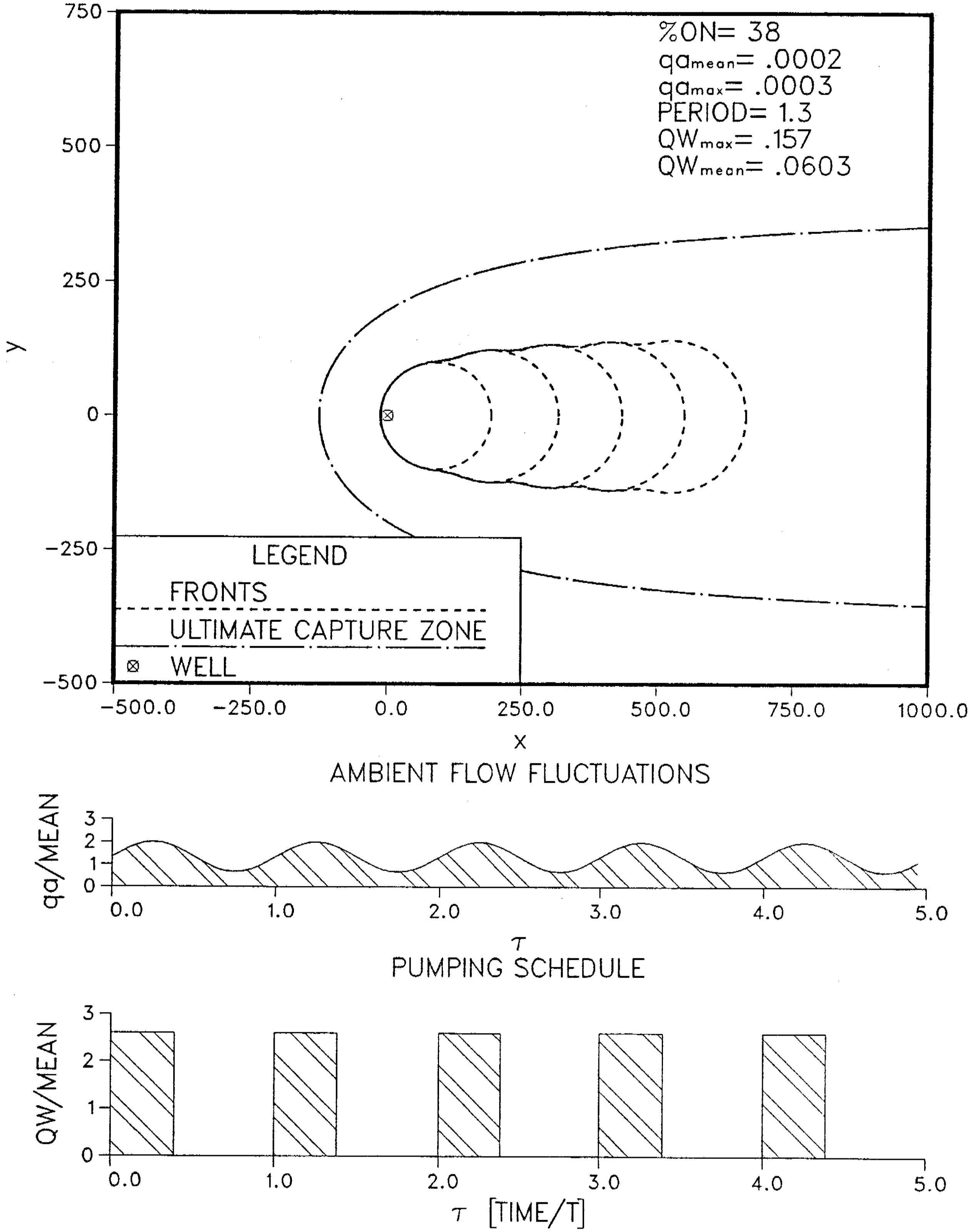
As mentioned in the introduction, recharge can also be transient. This is a brief section demonstrating the effects a sinusoidal ambient flow rate has on a capture zone. The mean ambient flow rate is the same as the ambient flow rate used throughout the previous simulations. The pumping rate is the same as in the first example (0.157). All other parameters stay the same.

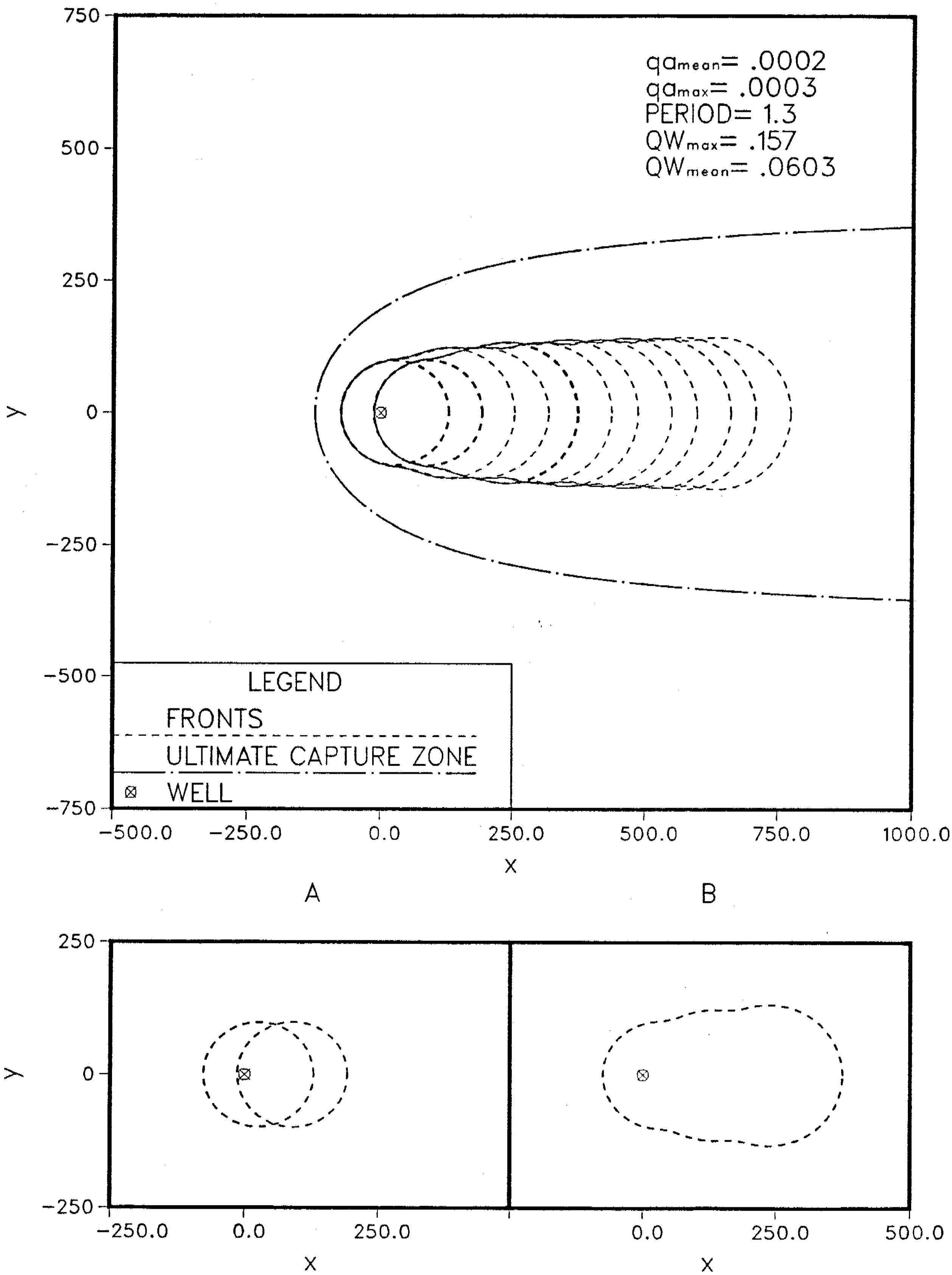
Figure 29 demonstrates the period capture zones for a sine wave that varies from 0 to 0.0003. The simulation exactly reproduces the results of the sine pumping rate simulation. Therefore, no other figures are presented. The results when both ambient flow and pumping rates are transient is presented in figures 30A-B. Here the pumping rate is a square wave with the same conditions that exist in the first example. The pumping rate is coupled with a sinusoidal ambient flow rate as described above. In figure 30A the pumping schedule and the ambient flow rate are illustrated. The period capture zones are exactly the same as those in the first example. In figure 30B there is no difference as compared to what? in the position or shape of the fronts. The transect probabilities are not the same, as what??!! In this example the transect probabilities have an irregular distribution compared for those over the same cross-section in the first example.



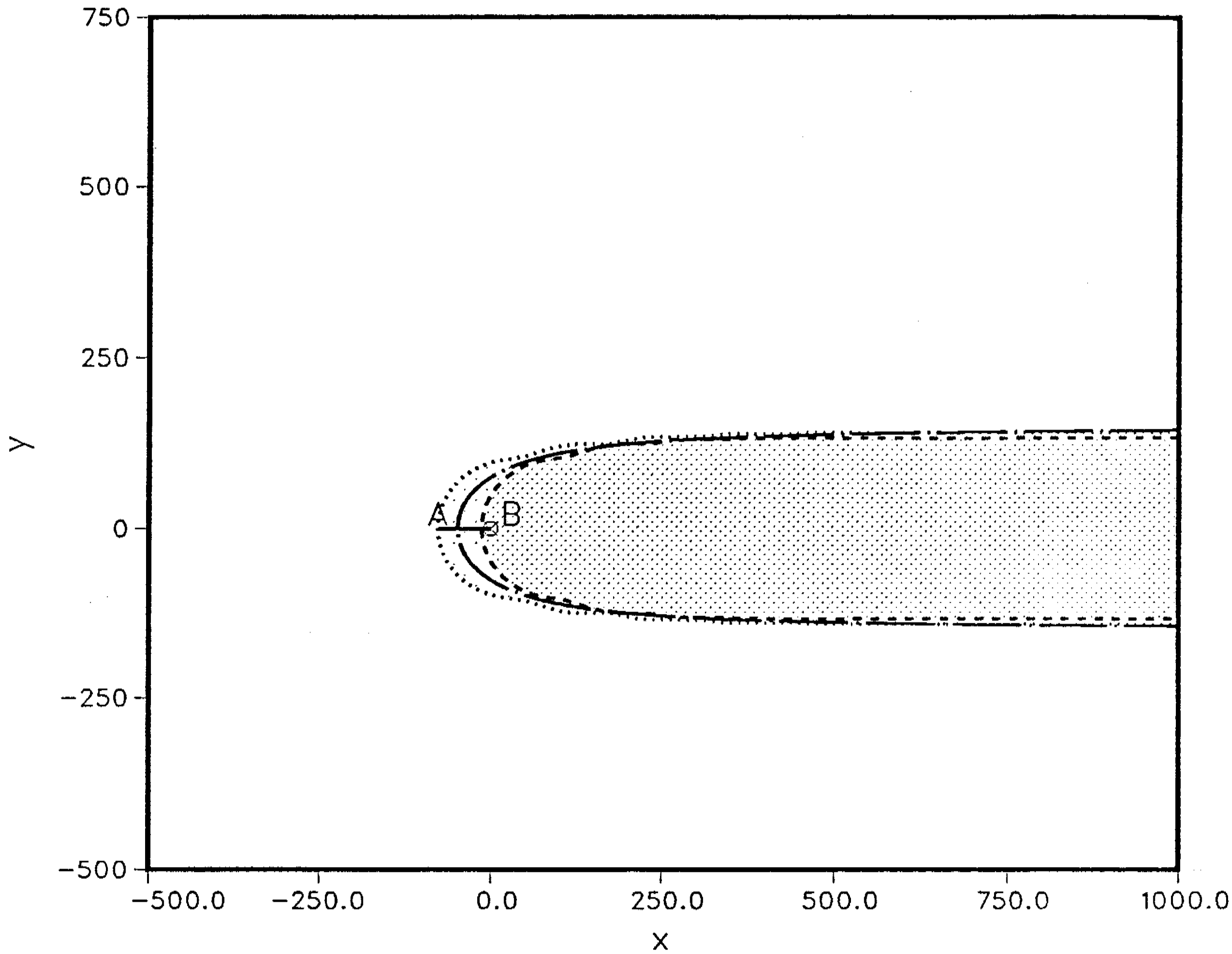

AMBIENT FLOW FLUCTUATIONS



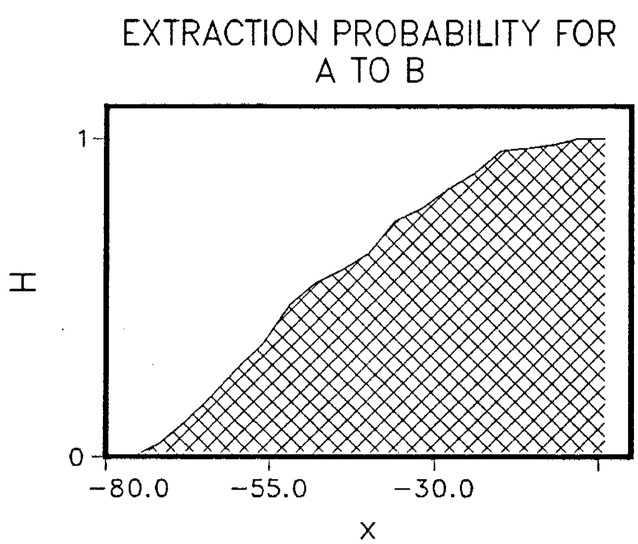

%ON= 38
qa_mean= .0002
qa_max= .0003
PERIOD= 1.3
QW_max= .157
QW_mean= .0603
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL
750
500
250
0
−250
−500
y
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x
AMBIENT FLOW FLUCTUATIONS
qa/MEAN
3
2
1
0
0.0
1.0
2.0
3.0
4.0
5.0
τ
PUMPING SCHEDULE
QW/MEAN
3
2
1
0
0.0
1.0
2.0
3.0
4.0
5.0
τ [TIME/T]


$qa_{mean}$= .0002
$qa_{max}$= .0003
PERIOD= 1.3
$QW_{max}$= .157
$QW_{mean}$= .0603
LEGEND
FRONTS
ULTIMATE CAPTURE ZONE
WELL
y
x
A
B

29.4.95



750
500
250
0
−250
−500
y
A
B
−500.0
−250.0
0.0
250.0
500.0
750.0
1000.0
x

EXTRACTION PROBABILITY FOR
A TO B
1
0
H
−80.0
−55.0
−30.0
x


30e

## CONCLUSIONS

no comparisons !!

The 2-dimensional particle tracking algorithm developed is flexible, efficient, and accurate. Coupled with the dynamic front tracker, equal time fronts for numerous conceptualizations can be accurately delineated. Of particular interest, in this study, is the application of the tracking routine to a transient flow field. The front tracking routine is used to generate capture zones in a quasi-steady state system. The limiting condition is that storage effects are assumed negligible and therefore ignored. An instantaneous response is also assumed.

yuk

The many simulations presented provide extensive insight into the dynamics of transient flow system. The graphical techniques utilized to compare simulations is simple but effective. The techniques included: period capture zone delineation, determination of the effective stagnation point, zonation of the flow field with respect to extraction probability, transect probability generation, and ultimate capture zone comparison.

Extensive examination of the results of the sensitivity analysis provides some interesting conclusion, all of which are fairly intuitive. The conclusions are:

(1) The effective stagnation point is the best predictor for the 0% boundary

(2) There are three basic patterns that the fronts may simulate

  (a) Fully connected - consecutive period capture zones never cross

  (b) Intermediately connected - consecutive PCZs share a portion of their boundary (hummocky patterns)

  (c) Disconnected - consecutive PCZs are completely disconnected

(3) The effective stagnation point can be used to predict the form of the fronts, and therefore can be used in a qualitative sense to determine probabilities.

(4) Sinusoidal variations only effect the transect probabilities, not the shape or position of fronts.

(5) The probability that a particle from a certain position enters the well given enough time can be

( not previously developed.

determined by superposition of the probability it will enter a period capture zone during the ambient flow portion of the period coupled with the probability during the induced flow portion.

Finally, the research done by the author and others most definitely suggests that modeling transient system using a steady state conceptualization may lead to serious problems. Simple inexpensive models such as the one presented could be valuable tools for first cut decision making. They provide gross fluid behavior detail such as where a pumping well is getting its water and what path it takes to get there.

~~Appendix A~~

# REFERENCES


Bear, J., 1972, Dynamics of Fluids in Porous Media, Dover Publications Inc., New York, New York, pp. 312-328.

Bear, J., 1979, Hydraulics of Groundwater,McGraw-Hill Inc., New York, New York, chapters 7-10 pp. 226-299.

Bear, J. and Jacobs, M., 1965, On the Movement of Water Bodies Injected into Aquifers, Journal of Hydrology 3, pp. 37-57.

Javandel, Iraj and Tsang, Chin-Fu, 1988, Capture-Zone Type Curves: a Tool for Aquifer Cleanup, Groundwater 24(5), pp. 616-625.

Lee, Kenrick H. L., 1986, Pollution Capture Zones for Pumping Wells in Aquifers with Ambient Flow, Master Degree Independent Study, New Mexico Institute of Mining and Technology, 76 pp..

Linderfelt, William R., Leppert, Shawn C., and Wilson, John L., 1989, Capture Zones for Well-Head Protection: Effect of Time Dependant Pumping, Saturated Thickness, and Parameter Uncertainty, Presented at the American Geophysical Union's Fall Meeting, San Francisco, California.

Newsom, Joan M. and Wilson, John L., 1988, Flow of Ground Water to a Well Near a Stream - Effect of Ambient Ground-Water Flow Direction, Groundwater 26(6), pp. 703-711.

Scafer-Perini, Annette, Wilson, John L., and Perini, Mathew, 1990, Efficient and Accurate Front Tracking for Two Dimensional Groundwater Flow Methods, Paper submitted for review to Water Resources Research.

Zheng, Chunmiao, 1988, New Solution and Model for Evaluation of Groundwater Pollution Control, PhD. Dissertation, the University of Wisconsin- Madison.

Wilson, John L. and Linderfelt, William R., 1989, The Influence of Streams, Barriers, Groundwater Divides, and Other Hydrologic Features on Well Capture Zones, Presented at the 28th International Geological Congress, Washington, D.C..

## APPENDIX A

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCC  PTHLINE   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C      THIS SUBROUTIMNE GENERATES THE PATHLINE OF A PATICLE IN THE     C
C      SYSTEM.  DRIVES THE 2D CAPTURE ZONE CASES.  EVERYTHING IS IN    C
C      SUBROUTINE FORM.  THE SIZE OF THE ARRAYS IS GOVERN BY NPX       C
C      LOCATED IN THE COMMON BLOCK.  THIS MODEL CAN START THE TRACKING C
C      AT ANY TIME SO AN INITIAL REAL TIME MUST BE SPECIFIED. IT IS ALSO C
C      IMPORTANT TO NOTE THAT BETA CAN BE PASSED INTO THE PROGRAM. THIS C
C      IS ONLY NESSICARY IF THE INPUT PARAMETERS ARE NOT TO STAY CONSTANT. C
C      THIS ROUTINE CAN IMPLEMENT TIME VARYING INPUTS (QW,QA).         C
C                                                                      C
C                  PASS                                                C
C      -----------------------------------------------------------     C
C      BETA = DIMENSIONALESS PUMPING COEFFICIENT                       C
C      XT,YT = POSITION IN SPACE                                       C
C      TIMEIN = THE TIME THAT THE PARTICLE TRACKER IS TURNED ON [T]    C
C      TIMMAX = MAXIMUM ALLOWED RUNNING TIME                           C
C      TM= 1.0 FOR BACKWARD TRACKING                                   C
C         -1.0 FOR FORWARD  TRACKING                                   C
C                                                                      C
C                  FLAGS                                               C
C      -----------------------------------------------------------     C
C      LTRACK = 0 NO VARIATION IN LENGTH CONSTANT (CON)                C
C               1 VARIATION IN LENGTH CONSTANT (CON)                   C
C                                                                      C
C                       INPUT                                          C
C      -----------------------------------------------------------     C
C                                                                      C
C                       OUTPUT                                         C
C      -----------------------------------------------------------     C
C      XT, YT = NEW POSITION IN SPACE                                  C
C      TIMMAX = THE CURRENT TIME IN THE SYSTEM                         C
C                                                                      C
C                LEVEL 3                                               C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE PTHLINE (BETA,XT,YT,TIMEIN,TIMMAX,TM)

      LOGICAL HALT
      INCLUDE 'ALL_COMMON.COM'
      DIMENSION XARAY(NPX),YARAY(NPX),TARRAY(NPX),AN1RAY(NPX),
     *          AN2RAY(NPX)
c      INCLUDE 'ALL_OPEN.COM'
c      INCLUDE 'ALL_VALUE.COM'

C$$$$      TEST BLOCK
C      BETA=.95
C      XT=100.
C      YT=10.
C      TIMMAX=1000000000.
C      TIMEIN=0.0
C      TM=1.

      LHALT=0
      HALT=.FALSE.
      TIME=TIMEIN
      DELT=0.00001
      NUMPTS=1

C$$$$$         START LOCATION COMPUTATION LOOP
      DO 100 I=1,NPX
         CALL TARAY (XT,YT,XARAY,YARAY,NUMPTS)

         IF(LTRACK .EQ. 1)THEN
            XT1=XT
            YT1=YT
            TIME1=TIME
            CALL TRACKER(XT,YT,BETA,DELT,TIME,TIMMAX,TM)
            XT=XT1
            YT=YT1
            TIME=TIME1
         ENDIF

         CALL TVARI(TIME,DELT,AN1,AN2,BETA)
```

```
          CALL VELOCS (TM,XT,YT,VX,VY)
          CALL TMOVE (TIME,TIMMAX,VX,VY,DELT,XT,YT)

          NUMPTS=NUMPTS+1
          CALL TIMARY(TIMMAX,TIME,AN1,AN2,TARRAY,AN1RAY,AN2RAY,NUMPTS,
     *              LHALT)
          CALL TBOUND (XT,YT,LHALT,XARAY,YARAY,AN1RAY,AN2RAY,TARRAY
     *                  ,NUMPTS)

          IF(LHALT .EQ. 1)HALT=.TRUE.
          IF(HALT) GO TO 190
  100   CONTINUE

  190   CALL TWRITE (XARAY,YARAY,AN1RAY,AN2RAY,TARRAY,NUMPTS)

        RETURN
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCC  TMOVE     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                            C
C      THIS SUBROUTIMNE MOVES THE PARTICLR TO THE TRAJECTED                  C
C      POSITION.  ADJUSTS FINAL TIME STEP TO THE TIME LEFT                   C
C      UNTIL MAXIMUM ALLOWED. THIS CODE ALSO CAN GENERATE EQUIPOTENTIALS.    C
C                                                                            C
C                  PASS                                                      C
C      ----------------------------------------------------------            C
C      VX,VY = VELOCITY VETORS                                               C
C      TIME = CURRENT RUNNING TIME                                           C
C      TIMMAX = MAXIMUM ALLOWED RUNNING TIME                                 C
C      XT,YT = POSITION IN SPACE                                             C
C      DELT = TIME STEP                                                      C
C                                                                            C
C                  FLAGS                                                     C
C      ----------------------------------------------------------            C
C      LFRONT IS CLASS TO DETERMINE FRONTS OR NOT                            C
C             = 0 : NO FRONT                                                 C
C             = 1 : FRONT                                                    C
C      LPOT IS THE FLAG FOR EQUIPOTENTIALS                                   C
C             = 0 : NO EQUIPOTENTIAL                                         C
C             = 1 : EQUIPOTENTIALS                                           C
C                                                                            C
C                       INPUT                                                C
C      ----------------------------------------------------------            C
C      RBD = RIGHT BOUNDARY                                                  C
C      CON = LENGTH CONSTANT                                                 C
C                                                                            C
C                       OUTPUT                                               C
C      ----------------------------------------------------------            C
C      XT, YT = NEW POSITION IN SPACE                                        C
C      DELT = NEW TIME STEP                                                  C
C                                                                            C
C                LEVEL 1                                                     C
C                                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE TMOVE (TIME,TIMMAX,VX,VY,DELT,XT,YT)

      INCLUDE 'ALL_COMMON.COM'

      VEL=SQRT(VX**2+VY**2)
      DELT=CON/VEL

      IF(DELT .GE. (TIMMAX-TIME))DELT=TIMMAX-TIME

      IF (LPOT .EQ. 1)THEN
         A=VX
         AA=VY
         VX=AA
         VY=-A
      ENDIF

      TIME=TIME+DELT
      DELX=-1.*DELT*VX
      DELY=-1.0*DELT*VY

      XT=XT+DELX
      YT=YT+DELY

      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCC  TIMARY    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                          C
C      THIS SUBROUTIMNE PLACES THE TIME VARYING INPUTS INTO ARRAY          C
C      FORM. IT ALSO CHECKS THE TIME FOR STOPPING CRITERIUM.               C
C                                                                          C
C                  PASS                                                    C
C      -----------------------------------------------------------        C
C      TIME = CURRENT RUNNING TIME                                         C
C      TARRAY = THE REAL TIME OF EACH POINT [ARRAY]                        C
C      AN1RAY = THE VARIATION OF QWMN W/R TIME [ARRAY]                     C
C      AN2RAY = THE VARIATION OF QAMN W/R TIME [ARRAY]                     C
C      NUMPTS = NUMBER OF POSITIONS                                        C
C      AN1 = PUMPING RATE AS A FUNCTION OF TIME                            C
C      AN2 = AMBIENT FLOW RATE AS A FUNCTION OF TIME                       C
C      TIMMAX = MAXIMUM ALLOWED TIME TO TRAVEL                             C
C      LHALT = WHETHER TO STOP TRACKING OR NOT                             C
C          0 = CONTINUE                                                    C
C          1 = STOP                                                        C
C                                                                          C
C                  FLAGS                                                   C
C      -----------------------------------------------------------        C
C      LTV IS THE INPUT VARIABLE CLASS OF FLAGS                            C
C      LTV0 = 0  : DOES NOT VARY AS A FUNCTION OF BETA                     C
C           = 1 :  QAMN IS THE VARIABLE                                    C
C           = 2 :  DMN IS THE VARIABLE                                     C
C           = 3 :  QWMN IS THE VARIABLE                                    C
C      LTV1 = 0  : PUMPING DOES NOT VARY W/R TO TIME                       C
C           = 1 :  SQUARE WAVE FUNCTION                                    C
C      LTV2 = 0  : AMBIENT FLOW MAGNITUDE DOES NOT VARY                    C
C                  W/R TO TIME                                             C
C           = 1 :  SQUARE WAVE FUNCTION                                    C
C                                                                          C
C                       INPUT                                              C
C      -----------------------------------------------------------        C
C                                                                          C
C                       OUTPUT                                             C
C      -----------------------------------------------------------        C
C                                                                          C
C                LEVEL 1                                                   C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE TIMARY(TIMMAX,TIME,AN1,AN2,TARRAY,AN1RAY,AN2RAY,
     *                  NUMPTS,LHALT)

      INCLUDE 'ALL_COMMON.COM'
      DIMENSION TARRAY(NPX),AN1RAY(NPX),AN2RAY(NPX)

      IF((LTV1 .EQ. 1) .AND. (LTV2 .EQ. 0))THEN
         TARRAY(NUMPTS)=TIME
         AN1RAY(NUMPTS)=AN1
      ELSEIF((LTV2 .EQ. 1) .AND. (LTV1 .EQ. 0))THEN
         TARRAY(NUMPTS)=TIME
         AN1RAY(NUMPTS)=AN2
      ELSEIF((LTV1 .EQ. 1) .AND. (LTV2 .EQ. 1))THEN
         TARRAY(NUMPTS)=TIME
         AN1RAY(NUMPTS)=AN1
         AN2RAY(NUMPTS)=AN2
      ENDIF

      IF (TIME .GE. TIMMAX)THEN
         LHALT=1
            WRITE (6,*)'PARTICLE AT TIME  LIMIT'
      ENDIF

      RETURN
      END
```

```
            IF((DISWL .LE. 1.) .AND. (QWMN .NE. 0.00))THEN
               LHALT=1
C               WRITE(6,*)'PARTICLE AT WELL'
            ELSEIF(XT .GE. RBD)THEN
               XT=RBD
               LHALT=1
C               WRITE(6,*)'PARTICLE AT RIGHT LIMIT'
            ELSEIF(XT .LT. WBD)THEN
               XT=WBD
               LHALT=1
C               WRITE(6,*)'PARTICLE AT LEFT LIMIT'
            ELSEIF(YT .LT. SBD)THEN
               YT=SBD
               LHALT=1
C               WRITE (6,*)'PARTICLE AT LOWER  LIMIT'
            ELSEIF(YT .GT. UBD)THEN
               YT=UBD
               LHALT=1
C               WRITE (6,*)'PARTICLE AT UPPER/LOWER  LIMIT'
            ENDIF

         ENDIF

         IF(LHALT .EQ. 1)THEN
            IF(LBOUND .EQ. 0)THEN
               XARAY(NUMPTS)=XT
               YARAY(NUMPTS)=YT
            ELSEIF(LBOUND .EQ. 1)THEN
               XARAY(NUMPTS)=XT/DMN
               YARAY(NUMPTS)=YT/DMN
            ENDIF
         ENDIF

         RETURN
         END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCC  TBOUND    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C      THIS SUBROUTINE CONTAINS ALL BOUNDARIES.  IT ALSO PUTS THE     C
C      LAST SET OF POSITION INTO ARRAYS (VALUE AT BOUNDARY).          C
C                                                                     C
C                  PASS                                               C
C      ---------------------------------------------------------      C
C      XT,YT = POSITION IN SPACE                                      C
C      LHALT = A FLAG TO DECIDE WHETHER TO STOP PROGRAM IF THE PARTICLE C
C              HAS REACHED A BOUNDARY                                 C
C      XARAY, YARAY = THE POSITION OF THE PATHLINE, [ARRAY]           C
C      TARRAY = THE REAL TIME OF EACH POINT [ARRAY]                   C
C      AN1RAY = THE VARIATION OF QWMN W/R TIME [ARRAY]                C
C      AN2RAY = THE VARIATION OF QAMN W/R TIME [ARRAY]                C
C      NUMPTS = NUMBER OF POSITIONS                                   C
C                                                                     C
C                  FLAGS                                              C
C      ---------------------------------------------------------      C
C      LBOUND BOUNDARY TYPE CLASS OF FLAGS                            C
C             = 0  : INFINITE BOUNDARIES                              C
C             = 1  : SEMI-INFINITE BOUNDARY (RIVER)                   C
C             = 2  : SEMI-INFINITE BOUNDARY (BARRIER)                 C
C             = 3  : STREAM-BARRIER BOUNDARY                          C
C             = 4  : STREAM-STREAM  BOUNDARY                          C
C                                                                     C
C                       INPUT                                         C
C      ---------------------------------------------------------      C
C      WBD = LEFT BOUNDARY                                           C
C      RBD = RIGHT BOUNDARY                                           C
C      UBD = UPPER BOUNDARY                                           C
C      SBD = LOWER BOUNDARY                                           C
C      DMN = DISTANCE TO WELL FROM LEFT BOUNDARY                      C
C                                                                     C
C                       OUTPUT                                        C
C      ---------------------------------------------------------      C
C      LHALT = A FLAG TO DECIDE WHETHER TO STOP PROGRAM IF THE PARTICLE C
C              HAS REACHED A BOUNDARY                                 C
C                                                                     C
C                LEVEL 1                                              C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE TBOUND (XT,YT,LHALT,XARAY,YARAY,AN1RAY,AN2RAY,TARRAY
     *                   ,NUMPTS)

      INCLUDE 'ALL_COMMON.COM'
      DIMENSION XARAY(NPX),YARAY(NPX),AN1RAY(NPX),AN2RAY(NPX),
     *          TARRAY(NPX)

      IF(LBOUND .EQ. 1)THEN
         DISWL=SQRT((XT-DMN)**2+YT**2)

         IF(XT .LE. 0.01)THEN
            XT=WBD
            LHALT=1
C            WRITE(6,*)'PARTICLE AT STREAM'
         ELSE IF((DISWL .LT. 1.) .AND. (QWMN .NE. 0.00))THEN
            LHALT=1
C            WRITE(6,*)'PARTICLE AT WELL'
         ELSE IF(XT .GE. RBD)THEN
            XT=RBD
            LHALT=1
C            WRITE(6,*)'PARTICLE AT RIGHT LIMIT'
         ELSEIF(YT .LT. SBD)THEN
            YT=SBD
            LHALT=1
C            WRITE (6,*)'PARTICLE AT LOWER  LIMIT'
         ELSEIF(YT .GT. UBD)THEN
            YT=UBD
            LHALT=1
C            WRITE (6,*)'PARTICLE AT UPPER/LOWER  LIMIT'
         ENDIF

      ELSEIF(LBOUND .EQ. 0)THEN
         DISWL=SQRT(XT**2+YT**2)
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCC  PLOT_FRNT CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C      THIS SUBROUTIMNE DECIDES WHICH PLOTTING SCHEME IS TO BE USED.   C
C                                                                      C
C                 PASS                                                 C
C      -----------------------------------------------------------     C
C      BETA = DIMENSIONALESS PUMPING COEFFICIENT                       C
C      TIME = CURRENT TIME [T]                                         C
C      LPFRNT= FLAG TO DECIDE WHETHER TO PLOT                          C
C           0=NOT TO PLOT                                              C
C           1=PLOT                                                     C
C      DTYM=DIMENSIONALESS TIME INCREMENT FOR PLOTTING OF FRONT        C
C                                                                      C
C                 FLAGS                                                C
C      -----------------------------------------------------------     C
C      LPFRNT= FLAG TO DECIDE WHETHER TO PLOT                          C
C           0=NOT TO PLOT                                              C
C           1=PLOT                                                     C
C      LPLT GROUP OF FLAGS                                             C
C           1=PLOT ONLY ON/OFF INSTANTANEOUS TIMES                     C
C           2=PLOT INCREMENTAL FRONTS                                  C
C           3=PLOT ALL PUMPING PERIOD FRONTS                           C
C                                                                      C
C                      INPUT                                           C
C      -----------------------------------------------------------     C
C                                                                      C
C                      OUTPUT                                          C
C      -----------------------------------------------------------     C
C      LPFRNT= FLAG TO DECIDE WHETHER TO PLOT                          C
C                                                                      C
C               LEVEL 2                                                C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE PLOT_FRNT (TIME,BETA,LPFRNT,DTYM)

      INCLUDE 'ALL_COMMON.COM'

      IF(LPLT .EQ. 1)THEN
         CALL PLT1(TIME,BETA,LPFRNT)
      ELSEIF(LPLT .EQ. 2)THEN
         CALL PLT2 (TIME,DTYM,BETA,LPFRNT)
      ELSEIF(LPLT .EQ. 3)THEN
         CALL PLT3 (TIME,BETA,LPFRNT)
      ENDIF

      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCC   DYNFRNT      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                        C
C     THIS SUBROUTINE IS THE DYNAMIC FRONT TRACKER.  IT USES A SCHEME    C
C     WHERE IF A PARTICLE IS DIVERGING FROM THE ADJACENT PARTICLE A      C
C     NEW PARTICLE WILL BE PUT HALF WAY BETWEEN THE TWO DIVERGING PARTICLES.  C
C     THE VARIABLE STEP CHANGER SHOULD NOT BE USED HERE.                 C
C                                                                        C
C                  PASS                                                  C
C      -----------------------------------------------------------      C
C      BETA = DIMENSIONALESS PUMPING COEFFICIENT                         C
C      XO,YO = INITIAL POSITION IN SPACE                                 C
C      NPTS = NUMBER OF INITIAL POINTS                                   C
C      SEPMAX = MINIMUM DISTANCE BETWEEN ADJACENT PARTICLES WHEN A NEW   C
C               PARTICLE IS TO BE PLACED                                 C
C      TYMCEK = DIMENSIONALESS TIME BETWEEN CHECKING FOR DIVERGANCE      C
C      TYMTOT = DIMENSIONALESS TIME WHEN THE TRACKING IS TO CEASE        C
C      LSEE = FLAG TO TELL WHETHER TO PLOT THE FRONTS ON THE SCREEN NOW  C
C          1=PLOT                                                        C
C          0=DON'T PLOT                                                  C
C      DTYM = DIMENSIONALESS PERIOD BETWEEN PLOTTING FRONTS(FOR USE ONLY C
C             WHEN A CONSTANT PERIOD BETWEEN PLOTS IS USED)              C
C      TM= 1.0 FOR BACKWARD TRACKING                                     C
C         -1.0 FOR FORWARD  TRACKING                                     C
C      TIMEIN=TIME THE FRONT SHOUOLD BEGIN                               C
C                                                                        C
C                  FLAGS                                                 C
C      -----------------------------------------------------------      C
C      LPLT FAMILY IS USED TO SIGNAL WHAT TYPE OF OUTPUT                 C
C          1=USING PLT1 FOR FRONT PLOTS(PLOTS ON/OFF TRANSITION)         C
C          2=USING PLT2 FOR FRONT PLOTS(PLOTS INCREMENTED FRONTS)        C
C          3=USING PLT3 FOR FRONT PLOTS(PLOTS ALL FRONTS THAT ENTER WELL)  C
C                                                                        C
C                       INPUT                                            C
C      -----------------------------------------------------------      C
C      XO,YO = INITIAL POSITION IN SPACE                                 C
C                                                                        C
C                       OUTPUT                                           C
C      -----------------------------------------------------------      C
C      POINTS ON THE NEEDED FRONTS IN FRNT.DAT                           C
C                                                                        C
C               LEVEL 4                                                  C
C                                                                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE DYNFRNT(BETA,XO,YO,NPTS,SEPMAX,TYMCEK,TYMTOT,
     *                   LSEE,DTYM,TM,TS)

      INCLUDE 'ALL_COMMON.COM'
      DIMENSION XO(NPX),YO(NPX)

C$$$$      INITIAL CONSTANTS
      TIME=TS
      LFRONT=1
      NPL=0
      DDMN=250.

      IF(LSEE .EQ. 1)THEN
         DO 10 JJ=1,NPTS
            WRITE(6,*)XO(JJ),YO(JJ)
 10      CONTINUE

         XMAX=RBD
         XMIN=WBD
         YMIN=SBD
         YMAX=UBD

         CALL PLOTEM(XMIN,XMAX,YMIN,YMAX)
      ENDIF

      TIMCEK=(TYMCEK*DDMN*POR*B/(PI*QAMIN*BETA))
      TIMTOT=(TYMTOT*DDMN*POR*B/(PI*QAMIN*BETA))
      ITT=0

      DO 200 NTIM=1,NPX
         ITT=ITT+1
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCC  SUBROUTINE BETAC_LMVANG  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C          THIS SUBROUTINE ITERATES TO FIND THE CRITICAL BETA           C
C          (THE BETA IN WHICH THE CAPTURE ZONE JUST TOUCHES THE STREAM) C
C          FOR GIVEN CONDITIONS WITH AMBIENT FLOW TO AN ANGLE           C
C          USING SEMI-INFINITE BOUNDARY CONTITIONS.                     C
C          MAY BE QUESTIONABLE ABOVE ANGLES OF 88 DEGREES.              C
C                                                                       C
C                   PASS                                                C
C       ---------------------------------------------------------      C
C       BETAC = CRITICAL DIMENSIONALESS PUMPING COEFFICIENT             C
C                                                                       C
C                   FLAGS                                               C
C       ---------------------------------------------------------      C
C             LTVO = 0  : DOES NOT VARY AS A FUNCTION OF BETA           C
C                  = 1 :  QAMN IS THE VARIABLE                          C
C                  = 2 :  DMN IS THE VARIABLE                           C
C                  = 3 :  QWMN IS THE VARIABLE                          C
C                                                                       C
C                        INPUT                                          C
C       ---------------------------------------------------------      C
C       ALPHA = ANGLE OF FLOW IN RADIANS                                C
C       DMN = IMAGE DISTANCE FROM STREAM TO WELL                        C
C                                                                       C
C                        OUTPUT                                         C
C       ---------------------------------------------------------      C
C       BETAC = CRITICAL DIMENSIONALESS PUMPING COEFFICIENT             C
C                                                                       C
C                 LEVEL 2                                               C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE BETAC_LMVANG (BETAC)

      DIMENSION DIFF(2)

      INCLUDE 'ALL_COMMON.COM'

      NFLAG=0
      BMAX=1.2
      CONVER=.0000001

      ALPHA=DEGREE*PI/180.
      BETA=(1.0*COS(ALPHA))+0.00001
      TERM=BETA/(COS(ALPHA))

      IF(LTVO .EQ. 1)QAMN=QWMN/(PI*DMN*BETA)
      IF(LTVO .EQ. 2)DMN=QWMN/(PI*QAMN*BETA)
      IF(LTVO .EQ. 3)QWMN=QAMN*(PI*DMN*BETA)

C$$$        FIND THE FIRST STAGNATION POTENTIAL
      CALL STGPNT_LMVANG(BETA,YPNT1,XPNT1,XPNT2,YPNT2)

      XPNT=XPNT2
      YPNT=YPNT2
      STAGPOT= FLMV_POTEN (XPNT,YPNT)

C$$$ FIND THE POTENTIAL AT X=0.0
      YA=DMN*SQRT((BETA/COS(ALPHA))-1.0)
      XPNT=0.0
      YPNT=YA

      YAPOT= FLMV_POTEN (XPNT,YPNT)

C$$$        CALCULATE THE DIFFERENCE BETWEEN THOSE POTENTIALS
      DIFF(1)=YAPOT-STAGPOT
      BMIN=BETA

C$$$      KICK OUT IF THEY ARE THE SAME
      IF(ABS(DIFF(1)) .LT. CONVER)GOTO 50

C$$$$  ESTIMATE THE CRITICAL BETA BY ITERATION
      DO 100 I=1,100
         NFLAG=NFLAG+1
         BETA=((BMAX-BMIN)/2.0)+BMIN
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCC  SUBROUTINE STGNTN_STRMLN  CCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          THIS SUBROUTINE CREATES THE CAPTURE ZONE USING                      C
C          ANY OF THE BOUNDARIES.  IT MIRRORS NON ANGLED FLOW AND              C
C          ACTUALLIY TRACKS ALL THE STREAMLINE FOR ANGLED FLOW.                C
C                  PASS                                                        C
C      ------------------------------------------------------------            C
C      BETA = DIMENSIONALESS PUMPING COEFFICIENT                               C
C                                                                              C
C                  FLAGS                                                       C
C      ------------------------------------------------------------            C
C      LBOUND BOUNDARY TYPE CLASS OF FLAGS                                     C
C             = 0  : INFINITE BOUNDARIES                                       C
C             = 1  : SEMI-INFINITE BOUNDARY (RIVER)                            C
C             = 2  : SEMI-INFINITE BOUNDARY (BARRIER)                          C
C             = 3  : STREAM-BARRIER BOUNDARY                                   C
C             = 4  : STREAM-STREAM  BOUNDARY                                   C
C      LANGLE ANGLE CLASS OF FLAGS                                             C
C             = 0  : PERPENDICULAR FLOW                                        C
C             = 1  : ANGLE FLOW                                                C
C                                                                              C
C                       INPUT                                                  C
C      ------------------------------------------------------------            C
C      BETA = DIMENSIONALESS PUMPING COEFFICIENT                               C
C      DEGREE= ANGLE OF FLOW IN DEGREES                                        C
C                                                                              C
C                       OUTPUT                                                 C
C      ------------------------------------------------------------            C
C      STGNTN_STRMLN.DAT                                                       C
C                                                                              C
C                LEVEL 4                                                       C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE STGNTN_STRMLN(BETA)

      INCLUDE 'ALL_COMMON.COM'
C      INCLUDE 'ALL_OPEN.COM'
C      INCLUDE 'ALL_VALUE.COM'

      LFRONT=0
      LPOT=0

      IF(LBOUND .EQ. 1)THEN
         IF(LANGL .EQ. 0)THEN
            CALL PERFLOW_LMV (BETA)
         ELSE
            ALPHA=DEGREE*PI/180.
            CALL ANGFLOW_LMV (BETA)
         ENDIF
      ELSEIF(LBOUND .EQ. 0)THEN
         CALL PERFLOW_INF (BETA)
      ENDIF

      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCC  TRACKER   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C       THIS SUBROUTINE ITERATES TO FIND THE BEST CON WHICH IS THE    C
C       CONSTANT USED TO DETERMINE THE TIME STEP.  IT IS DONE BY TRACKING C
C       A PARTICLE 1 STEP AT THE CURRENT CON, THEN DIVIDE THE CON IN HALF C
C       AND SEE WHETHER THE DIFFERENCE IS WIHTIN TOLERANCE.  IF IT ISN'T  C
C       DIVIDE THIS CON IN HALF AGAIN AND ITERATE UNTIL IT DOES.  IF THE  C
C       CURRENT CON IS WITHIN THE TOLL THEN INCREASE THE CON BY 2 AND     C
C       CHECK TO SEE IF THIS IS WITHIN THE GIVEN TOLL ITERATE.            C
C                                                                     C
C                   PASS                                              C
C       -------------------------------------------------------------C
C       BETA = DIMENSIONALESS PUMPING COEFFICIENT                     C
C       XT,YT = POSITION IN SPACE                                     C
C       TIME = CURRENT TIME [T]                                       C
C       TIMMAX = MAXIMUM ALLOWED RUNNING TIME                         C
C       TM= 1.0 FOR BACKWARD TRACKING                                 C
C          -1.0 FOR FORWARD  TRACKING                                 C
C       DELT= TIME STEP                                               C
C                                                                     C
C                   FLAGS                                             C
C       -------------------------------------------------------------C
C                                                                     C
C                        INPUT                                        C
C       -------------------------------------------------------------C
C       CON = CURRENT CONSTANT OF PROPORTIONALITY FOR TIME STEP [L]   C
C                                                                     C
C                        OUTPUT                                       C
C       -------------------------------------------------------------C
C       CON = CONSTANT OF PROPORTIONALITY FOR TIME STEP [L]           C
C                                                                     C
C                 LEVEL 3                                             C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE TRACKER(XT,YT,BETA,DELT,TIME,TIMMAX,TM)

      INCLUDE 'ALL_COMMON.COM'

C$$$$          FIND THE NEW POSITION AT THE CURRENT CON
      TSTART=TIME
      X=XT
      Y=YT
      CON1=CON

      CALL TVARI(TIME,DELT,AN1,AN2,BETA)
      CALL VELOCS (TM,XT,YT,VX,VY)
      CALL TMOVE (TIME,TIMMAX,VX,VY,DELT,XT,YT)
      XFIRST=XT
      YFIRST=YT

      XTONE=XT
      YTONE=YT

C$$$$    CHECK TO SEE IF A SMALLER TIME STEP IS NEEDED
      DO 100 I=1,20
         CON=CON1
         TIME=TSTART
         XTX=X
         YTY=Y
         CON=CON/(2.**I)

C$$$$    PARTICLE TRACK AT A CON 1/2 OF THE LAST CON
         DO 200 J=1,2**I
            CALL TVARI(TIME,DELT,AN1,AN2,BETA)
            CALL VELOCS (TM,XTX,YTY,VX,VY)
            CALL TMOVE (TIME,TIMMAX,VX,VY,DELT,XTX,YTY)
 200     CONTINUE

         RAD=SQRT((XTONE-XTX)**2+(YTONE-YTY)**2)

C$$$     IF THE CURRENT CON IS WITHIN TOLL CHECK TO SEE IF CAN GO HIGHER
         IF((RAD .LT. TOLL) .AND. (I .EQ. 1))THEN

            DO 210 II=1,10
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCC  STGPNT_INF   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C      THIS SUBROUTINE CALCULATES THE STAGNATION POINT FOR THE        C
C      INFIINITE BOUNDARY CONDITIONS.                                 C
C                                                                     C
C                 PASS                                                C
C      ---------------------------------------------------------      C
C      XT,YT = POSITION IN SPACE                                      C
C                                                                     C
C                 FLAGS                                               C
C      ---------------------------------------------------------      C
C                                                                     C
C                      INPUT                                          C
C      ---------------------------------------------------------      C
C      QWMN = PUMPING RATE                                            C
C      QAMN = AMBIENT RATE                                            C
C                                                                     C
C                      OUTPUT                                         C
C      ---------------------------------------------------------      C
C      XT,YT = POSITION IN SPACE                                      C
C                                                                     C
C               LEVEL 1                                               C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE STGPNT_INF (XT,YT)

      INCLUDE 'ALL_COMMON.COM'

      YT=2.0
      XT=-QWMN/(QAMN*2.0*PI)

      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCC  STGNT_PLOT   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C     THIS SUBROUTINE FINDS THE POSITION OF THE TWO PARTICLES NEAR      C
C     THE STAGNATION POINT THAT WILL BEST DELINEATE THE CAPTURE ZONE.   C
C     THIS IS DONE BY FINDING THE ANGLE OF THE LINE THAT CONNECTS THE   C
C     WELL WITH THE STAGNATION POINT.  USING THIS ANGLE THE POINTS ARE  C
C     SET 90 DEGREES PLUS AND MINUS TO SOME SPECIFIED SMALL DISTANCE (0.01). C
C                                                                     C
C                PASS                                                 C
C     ------------------------------------------------------------   C
C     XPNT2,YPNT2=POSITION OF STAGNATION POINT                        C
C     XPNT3,YPNT3=POSITION TO START THE TRACKING                      C
C     XPNT4,YPNT4=OTHER POSITION TO START THE TRACKING                C
C                                                                     C
C                FLAGS                                                C
C     ------------------------------------------------------------   C
C                                                                     C
C                     INPUT                                           C
C     ------------------------------------------------------------   C
C     DMN = IMAGE DISTANCE BETWEEN THE WELL AND THE STREAM            C
C                                                                     C
C                     OUTPUT                                          C
C     ------------------------------------------------------------   C
C     XPNT3,YPNT3=POSITION TO START THE TRACKING                      C
C     XPNT4,YPNT4=OTHER POSITION TO START THE TRACKING                C
C                                                                     C
C              LEVEL 1                                                C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE STGPNT_PLOT (XPNT2,YPNT2,XPNT3,YPNT3,XPNT4,YPNT4)

      DIMENSION XSTAG(2),YSTAG(2)

      INCLUDE 'ALL_COMMON.COM'

      ANGCON=90.
      R=0.01

      XWELL=DMN
      YWELL=0.0

      RISE=ABS(YPNT2)
      RRUN=XPNT2-XWELL
      SLOP=RISE/RRUN

      THETA=ATAN(SLOP)

      ANGCON=ANGCON*PI/180.0

      ANG1=THETA+ANGCON
      ANG2=THETA-ANGCON

      XSTAG(1)=R*COS(ANG1)
      YSTAG(1)=R*SIN(ANG1)
      XSTAG(2)=R*COS(ANG2)
      YSTAG(2)=R*SIN(ANG2)

      XPNT3=XPNT2+XSTAG(1)
      YPNT3=-(ABS(YPNT2)+YSTAG(1))
      XPNT4=XPNT2+XSTAG(2)
      YPNT4=-(ABS(YPNT2)+YSTAG(2))

      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCC  TARAY     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C      THIS SUBROUTINE MANIPULATES THE PARTICLE POSITIONS INTO          C
C      ARRAY FORM.                                                      C
C                                                                       C
C                  PASS                                                 C
C      ----------------------------------------------------------      C
C      XT,YT = POSITION IN SPACE                                        C
C      XARAY, YARAY = THE POSITION OF THE PATHLINE, [ARRAY]             C
C      NUMPTS = NUMBER OF POSITIONS                                     C
C                                                                       C
C                  FLAGS                                                C
C      ----------------------------------------------------------      C
C                       INPUT                                           C
C      ----------------------------------------------------------      C
C      DMN = DISTANCE TO WELL                                           C
C                                                                       C
C                       OUTPUT                                          C
C      ----------------------------------------------------------      C
C                                                                       C
C               LEVEL 1                                                 C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE TARAY (XT,YT,XARAY,YARAY,NUMPTS)

      INCLUDE 'ALL_COMMON.COM'
      DIMENSION XARAY(NPX),YARAY(NPX)

      IF(LBOUND .EQ. 1)THEN
         XARAY(NUMPTS)=XT/DMN
         YARAY(NUMPTS)=YT/DMN
      ELSEIF(LBOUND .EQ. 0)THEN
         XARAY(NUMPTS)=XT
         YARAY(NUMPTS)=YT
      ENDIF

      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCC   BETA_EST  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                         C
C          THIS SUBROUTINE ESTIMATES THE STAGNATION STREAMLINE            C
C          FOR A GIVEN POINT IN SPACE.  IT BACK TRACKS THE POINT          C
C          UNTIL IT HITS A BOUNDARY.  THEN IT DOES A BISECTIONAL          C
C          TO FIND THE STAGNATION STREAMLINE WITH THE SAME BOUNDARY       C
C          INTERSECTION.  THE GIVEN POINTS' BOUNDARY INTERSECTION CHANGES C
C          WITH THE CHANGING FLOW FIELD.                                  C
C                                                                         C
C                  PASS                                                   C
C      ------------------------------------------------------------       C
C      CRITBET = THE PARAMETER VALUE THAT WILL DEFINE THE ENCOMPASSING CZ C
C      XGIVEN,YGIVEN = INITIAL POSITION IN SPACE                          C
C                                                                         C
C                  FLAGS                                                  C
C      ------------------------------------------------------------       C
C      LBOUND BOUNDARY TYPE CLASS OF FLAGS                                C
C             = 0  : INFINITE BOUNDARIES                                  C
C             = 1  : SEMI-INFINITE BOUNDARY (RIVER)                       C
C             = 2  : SEMI-INFINITE BOUNDARY (BARRIER)                     C
C             = 3  : STREAM-BARRIER BOUNDARY                              C
C             = 4  : STREAM-STREAM  BOUNDARY                              C
C                                                                         C
C                       INPUT                                             C
C      ------------------------------------------------------------       C
C      RBD = RIGHT BOUNDARY                                               C
C      UBD = UPPER BOUNDARY                                               C
C                                                                         C
C                       OUTPUT                                            C
C      ------------------------------------------------------------       C
C      CRITBET = THE PARAMETER VALUE THAT WILL DEFINE THE ENCOMPASSING CZ C
C                                                                         C
C                LEVEL 5                                                  C
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE BETA_EST (XGIVEN,YGIVEN,CRITBET)

      INCLUDE 'ALL_COMMON.COM'

      DIMENSION VALUE(50),BBETA(50),XT(NPX),YT(NPX),
     *          XTT(NPX),YTT(NPX)

C$$$$  INITIALIZATION BLOCK

      TM=1.0
      TIMMAX=10000000000.
      TIMEIN=0.0
      LTIME=0
      LFRONT=0

      IF(LBOUND .EQ. 0)THEN
         DQW=QWMN/2.
         QWMN=QWMN/100.
         BETA=1.
      ELSEIF(LBOUND .EQ. 1)THEN
         BETA=0.005
         DBETA=0.5
      ENDIF

      X1=XGIVEN
      Y1=YGIVEN

C$$$  MAIN PROGRAM

      DO 1000 K=1,50
C$$$$$      CALL THE SUBROUTINES
         XGIVEN=X1
         YGIVEN=Y1

         CALL PTHLINE (BETA,XGIVEN,YGIVEN,TIMEIN,TIMMAX,TM)

         REWIND 50

         READ(50,*)NUMPTS
```

```
      SUBROUTINE CONTAM (XI,YI,DTYME,TYMMAX,NPTS,BETA,PERHIT)

      INCLUDE 'ALL_COMMON.COM'
      DIMENSION X1(10000),Y1(10000)

      TM=-1.
      NHIT=0
      NCOUNT=0

      DTIME=DTYME*250.*POR*B/(PI*QAMIN)
      TIMMAX=TYMMAX*250.*POR*B/(PI*QAMIN)
      TIMEIN=0.0

      DO 100 I=1,NPTS
         XT=XI
         YT=YI

         CALL PTHLINE (BETA,XT,YT,TIMEIN,TIMMAX,TM)
         REWIND 50
         REWIND 51
         DISWL=SQRT(XT**2+YT**2)
         IF(DISWL .LE. 1.)THEN
            NHIT=NHIT+1
         ELSE
            NCOUNT=NCOUNT+1
            X1(NCOUNT)=XT
            Y1(NCOUNT)=YT
         ENDIF

         TIMEIN=DTIME+TIMEIN
 100  CONTINUE

      WRITE(6,*)NHIT,NPTS
      PERHIT=FLOAT(NHIT)/FLOAT(NPTS)
C      WRITE(55,*)NCOUNT
C      DO 200 I=1,NCOUNT
C         WRITE(55,*)X1(I),Y1(I)
C 200  CONTINUE
C      WRITE(55,*)' '

      RETURN
      END
```

```
          CALL PLOT_FRNT (TIME,BETA,LPFRNT,DTYM)

          IF(LPFRNT .EQ. 1)THEN

             NPL=NPL+1
             IF(LSEE .EQ. 1)THEN
                CALL CURVE(XO,YO,NPTS,-1)
             ENDIF

             WRITE(53,*)TIME
             WRITE(53,*)NPTS
             DO 50 K=1,NPTS
                WRITE(53,*)XO(K),YO(K)
 50          CONTINUE
             WRITE(53,*)' '
          ENDIF

          IF(TIME .GT. TIMTOT)THEN
             GOTO 1000
          ENDIF

          TIMEIN=TIME
          TIMMAX=TIME+TIMCEK
C$$$ move the particles from the current time to time+deltat
          DO 300 I=1,NPTS
             CALL PTHLINE (BETA,XO(I),YO(I),TIMEIN,TIMMAX,TM)
 300      CONTINUE

          IF(NPTS.EQ.NPX)THEN
            GOTO 200
          ENDIF

C$$$  check the separation distance between adjacent particles
          NADD=0
          II=1

          DO 400 I=1,NPTS-1
             XDEL=(XO(II+1)-XO(II))
             YDEL=(YO(II+1)-YO(II))

             IF(XDEL**2+YDEL**2 .GT. SEPMAX)THEN
                NADD=NADD+1
                INEW=NPTS+NADD

C$$$    bump the arrays up one
                DO 500 J=INEW,II+2,-1
                   XO(J)=XO(J-1)
                   YO(J)=YO(J-1)
 500            CONTINUE

C$$$$   assign new coordinates for the particle number ii+1
                XO(II+1)=XO(II)+XDEL/2.
                YO(II+1)=YO(II)+YDEL/2.

C$$$    increment ii so that you don't check between the ones just added
                II=II+1
             ENDIF

C$$$$   now the next particle set to be checked is ii+1,ii+2 so let ii = ii + 1
             II=II+1
 400      CONTINUE
          NPTS=NPTS+NADD
          TIME = TIMEIN+TIMCEK

 200  CONTINUE
 1000 WRITE(53,*)' '
      WRITE(53,*)NPL

      IF(LSEE .EQ. 1)THEN
         CALL ENDPL(0)
         CALL DONEPL
      ENDIF

      RETURN
      END
```

```
      DO 101 III=1,NUMPTS
         READ(50,*)XT(III),YT(III)
101   CONTINUE
      REWIND 50

      XDN=XT(NUMPTS)
      YDN=YT(NUMPTS)

      IF(XDN .GE. RBD)THEN
         NTEST=1
      ELSEIF(YDN .GE. UBD)THEN
         NTEST=2
      ELSEIF(XDN .EQ. 0.0)THEN
         GOTO 999
      ENDIF

      CALL STGNTN_STRMLN(BETA)

      REWIND 50
      REWIND 52

C$$$$$   READ IN THE DATA FOR STREAMLINE AND FIND THE PLACES IT HITS
C$$$$$   THE BOUNDARY

      READ(52,*)NU
      DO 100 I=1,NU
         READ(52,*)XTT(I),YTT(I)
100   CONTINUE
      REWIND 52

      XTTUP=XTT(1)
      YTTUP=YTT(1)
      XTTDN=XTT(NU)
      YTTDN=YTT(NU)

C$$$$    FIND ORIENTATION OF STAGNATION CURVE
      IF((XTTUP .GE. RBD) .AND. (XTTDN .GE. RBD))THEN
         NAXIS=1
         YTDN=MIN(YTTUP,YTTDN)
         YTUP=MAX(YTTUP,YTTDN)
         XTUP=RBD
         XTDN=RBD
C         WRITE(6,*)'BOTH ENDS OF THE CURVE ARE ON THE Y-BNDRY'
      ELSEIF((XTTUP .EQ. 0.0) .OR. (XTTDN .EQ. 0.0))THEN
         NAXIS=4
         XTUP=MIN(XTTUP,XTTDN)
         XTDN=MAX(XTTUP,XTTDN)
      ELSEIF((XTTUP .GE. RBD) .OR. (XTTDN .GE. RBD))THEN
         XTUP=MIN(XTTDN,XTTUP)
         YTDN=MIN(YTTUP,YTTDN)
         NAXIS=2
C         WRITE(6,*)'ONLY ONE END OF THE CURVE IS ON X-BNDRY'
      ELSEIF((YTTUP .GE. UBD) .AND. (YTTDN .GE. UBD))THEN
         XTUP=MIN(XTTUP,XTTDN)
         XTDN=MAX(XTTUP,XTTDN)
         YTUP=UBD
         YTDN=UBD
         NAXIS=3
C         WRITE(6,*)'BOTH ENDS OF THE CURVE ARE ON X-BNDRY'
      ENDIF

C$$$$$ TEST THIS ITERARION
      IF(K .EQ. 1)THEN
         IF((NTEST .EQ. 1) .AND. (NAXIS .LE. 2))THEN
            DIFF1=ABS(YDN-YTDN)
            DIFF2=ABS(YDN-YTUP)
            IF(DIFF1 .LT. DIFF2)THEN
               NNN=-1
            ELSE
               NNN=1
            ENDIF
         ELSEIF((NTEST .EQ. 2) .AND. (NAXIS .GE. 2))THEN
            DIFF2=ABS(XDN-XTUP)
            DIFF1=ABS(XDN-XTDN)
            IF(DIFF1 .LT. DIFF2)THEN
               NNN=-1
```

```
          TERM=BETA/(COS(ALPHA))

          IF(LTVO .EQ. 1)QAMN=QWMN/(PI*DMN*BETA)
          IF(LTVO .EQ. 2)DMN=QWMN/(PI*QAMN*BETA)
          IF(LTVO .EQ. 3)QWMN=QAMN*(PI*DMN*BETA)

          CALL STGPNT_LMVANG(BETA,YPNT1,XPNT1,XPNT2,YPNT2)

          XPNT=XPNT2
          YPNT=YPNT2

          STAGPOT= FLMV_POTEN (XPNT,YPNT)

          YA=DMN*SQRT((BETA/COS(ALPHA))-1.0)
          XPNT=0.0
          YPNT=YA

          YAPOT= FLMV_POTEN (XPNT,YPNT)

          DIFF(2)=YAPOT-STAGPOT
          CON1=DIFF(1)*DIFF(2)

C$$$        CONVERGANCE CRITERIUM
          IF(ABS(DIFF(2)) .LE. CONVER)THEN
 50          BETAC=BETA
             GOTO 101
          ELSEIF(CON1 .LT. 0.0)THEN
             BMAX=BETA
          ELSE
             BMIN=BETA
          ENDIF

          DIFF(2)=DIFF(1)
 100  CONTINUE

 101  WRITE(6,*)' '

      RETURN
      END
```

```
                PRECON=CON
                CON=CON1
                TIME=TSTART
                XTX=X
                YTY=Y

                DO 220 JJ=1,2**II
                   CON=CON1
                   CALL TVARI(TIME,DELT,AN1,AN2,BETA)
                   CALL VELOCS (TM,XTX,YTY,VX,VY)
                   CALL TMOVE (TIME,TIMMAX,VX,VY,DELT,XTX,YTY)
 220            CONTINUE
                XTFST=XTX
                YTFST=YTY

                XTX=X
                YTY=Y
                CON=CON*(2.**II)
                CALL TVARI(TIME,DELT,AN1,AN2,BETA)
                CALL VELOCS (TM,XTX,YTY,VX,VY)
                CALL TMOVE (TIME,TIMMAX,VX,VY,DELT,XTX,YTY)

                RAD=SQRT((XTFST-XTX)**2+(YTFST-YTY)**2)

                IF(RAD .GT. TOLL)THEN
                   CON=PRECON
                   IF(II .EQ. 1)CON=CON1
                   GOTO 201
                ENDIF
 210         CONTINUE
          ENDIF

          XTONE=XTX
          YTONE=YTY
          IF(RAD .LT. TOLL)GOTO 201
 100   CONTINUE

 201   RETURN
       END
```

```
            ELSE
               NNN=1
            ENDIF
         ENDIF
      ENDIF

      IF((NAXIS .EQ. 1) .AND. (NNN .LT. 0))THEN
         DIFF=YTDN-YDN
      ELSEIF((NAXIS .EQ. 1) .AND. (NNN .GT. 0))THEN
         DIFF=YDN-YTUP
      ELSEIF((NAXIS .EQ. 2) .AND. (NNN .LT. 0))THEN
         DIFF=YTDN-YDN
      ELSEIF((NAXIS .EQ. 2) .AND. (NNN .GT. 0))THEN
         DIFF=XDN-XTUP
      ELSEIF((NAXIS .EQ. 3) .AND. (NNN .LT. 0))THEN
         DIFF=XDN-XTDN
      ELSEIF((NAXIS .EQ. 3) .AND. (NNN .GT. 0))THEN
         DIFF=XTUP-XDN
      ELSEIF((NAXIS .EQ. 4) .AND. (NNN .LT. 0.0))THEN
         DIFF=XDN-XTDN
      ELSEIF((NAXIS .EQ. 4) .AND. (NNN .GT. 0.0))THEN
         DIFF=-1.
      ENDIF

      CHK=ABS(DIFF)

      IF(CHK .LE. 0.005)THEN
         GOTO 1001
      ELSEIF(DIFF .GT. 0.0)THEN
         IF(LBOUND .EQ. 0)THEN
            QWMN=QWMN+DQW
         ELSEIF(LBOUND .EQ. 1)THEN
            BETA=BETA+DBETA
         ENDIF
      ELSEIF(DIFF .LT. 0.0)THEN
999      IF(LBOUND .EQ. 0)THEN
            QWMN=QWMN-DQW
            DQW=DQW*.5
         ELSEIF(LBOUND .EQ. 1)THEN
            BETA=BETA-DBETA
            DBETA=DBETA*.5
         ENDIF
      ENDIF
1000 CONTINUE

1001 IF(LBOUND .EQ. 1)THEN
        CRITBET=BETA
        WRITE(6,*)'ESTIMATED BETA IS ',CRITBET
     ELSEIF(LBOUND .EQ. 0)THEN
        CRITBET=QWMN
        WRITE(6,*)'ESTIMATED PUMPING IS ',CRITBET
     ENDIF

     RETURN
     END
```