An automatic, stagnation point based algorithm for the delineation of Wellhead Protection Areas

Tiziana Tosco,[1] Rajandrea Sethi,[1] and Antonio Di Molfetta[1]

Received 10 September 2007; revised 8 February 2008; accepted 10 March 2008; published 26 July 2008.

[1] Time-related capture areas are usually delineated using the backward particle tracking method, releasing circles of equally spaced particles around each well. In this way, an accurate delineation often requires both a very high number of particles and a manual capture zone encirclement. The aim of this work was to propose an Automatic Protection Area (APA) delineation algorithm, which can be coupled with any model of flow and particle tracking. The computational time is here reduced, thanks to the use of a limited number of nonequally spaced particles. The particle starting positions are determined coupling forward particle tracking from the stagnation point, and backward particle tracking from the pumping well. The pathlines are postprocessed for a completely automatic delineation of closed perimeters of time-related capture zones. The APA algorithm was tested for a two-dimensional geometry, in homogeneous and nonhomogeneous aquifers, steady state flow conditions, single and multiple wells. Results show that the APA algorithm is robust and able to automatically and accurately reconstruct protection areas with a very small number of particles, also in complex scenarios.

Citation: Tosco, T., R. Sethi, and A. Di Molfetta (2008), An automatic, stagnation point based algorithm for the delineation of Wellhead Protection Areas, *Water Resour. Res.*, *44*, W07419, doi:10.1029/2007WR006508.

1. Introduction

[2] A large number of models for capture zone delineation has been developed thus far. These models can be divided into deterministic methods, which yield Wellhead Protection Areas expressed as defined perimeters, and probability methods, which result in capture areas expressed in terms of probability maps. Deterministic capture areas are based only on advection phenomena. They can be generated using analytical solutions [*Bear and Jacobs*, 1965; *Javandel and Tsang*, 1986; *Ceric and Haitjema*, 2005], semianalytical methods [*Pollock*, 1989; *Blandford and Huyakorn*, 1991; *Fienen et al.*, 2005] or numerical methods. While analytical solutions can be applied only in a few cases and under several simplifying assumptions [*Kinzelbach et al.*, 1992], numerical methods allow us to take account also of complex situations (strong heterogeneities, transient flow, complex configuration of flow sources and sinks). In some cases, stream functions have been used in order to compute the flow field and to calculate time-related capture zones, both with analytical and numerical solutions [*Zheng and Bennett*, 2002]. The use of probability capture zones is justified by the uncertainty of the spatial distribution of aquifer parameters, which results in a strong unpredictability regarding the true position of well catchment perimeters [*Varljen and Shafer*, 1991; *Stauffer et al.*, 2002]. *Vassolo et al.* [1998] determined probability distributions for capture areas by means of stochastic inverse modeling, in order to take account of the spatial variability in the value of the hydraulic conductivity and of areal recharge. Monte Carlo analysis techniques, with both conditional and nonconditional simulations [*Van Leeuwen et al.*, 2000; *Guadagnini and Franzetti*, 1999; *Feyen et al.*, 2001; *Kunstmann and Kinzelbach*, 2000], have been also widely implemented. A backward probability model, based on the adjoint of the transport equation derived by *Neupauer and Wilson* [1999], describes the spreading course of a capture probability generated in the pumping well and its backward movement along the flow direction, according to advection-dispersion phenomena [*Frind et al.*, 2002, 2006; *Tosco et al*, 2006, 2007]. Uncertainty in the aquifer parameter distribution can be included in the macrodispersion coefficient, thus a time-related capture probability map can be obtained with only one backward simulation, significantly reducing the computing time.

[3] For practical applications, the deterministic methods, and in particular the backward particle tracking method, are, at the moment, the most widely used [*Pollock*, 1989]. For backward particle tracking, a number of particles are located around the flow sinks, and then traced backwards in the reversed flow direction. The time-related capture zones can be therefore computed with only one simulation, and the perimeter of the time-related capture zone is identified by the points reached by the particles after a simulation time equal to the fixed traveltime. However, as the traveltime increases, the distance between the end positions of the particles increases, resulting in a poor resolution of the capture zone perimeter, and a very high number of particles around the pumping wells may be necessary [*Strack*, 1989].

[4] The main problem arising from the use of the available routines for backward particle tracking is connected with the correct determination of the particle starting posi-

[1] DITAG-Dipartimento di Ingegneria del Territorio, dell'Ambiente e delle Geotecnologie, Politecnico di Torino, Torino, Italy.

0043-1397/08/2007WR006508

tions. Circles of equally spaced particles can be located around each pumping well. Nevertheless, a very high number of particles is here required for an accurate delineation of the perimeter of capture zones, even in homogeneous and isotropic aquifers, and especially downgradient the well. Tracing and managing thousands of pathlines can develop into a time-demanding operation when the perimeter of the capture zone has to be manually identified, but even when an automatic capture zone encirclement is desired. Furthermore, regardless of the use of a very large number of particles, it is impossible to avoid the exclusion (or inclusion) of particles with times of travel lower (or higher) than the fixed one, even in simple scenarios and using a basic linear interpolation. To circumvent the particle allocation problem, a dynamic particle allocation algorithm can be used. *Schafer-Perini and Wilson* [1991] have added some particles during the backward tracking simulation, when the distance between the existing particles exceeds an assigned value. *Bakker and Strack* [1996] have used a similar dynamic particle allocation, using a control on the smoothness of capture zone perimeters. The capture zone boundaries for the desired traveltimes are obtained employing an automatic iterative procedure, embedded in an analytic element code. Furthermore, the authors identify the stagnation point associated with a well, in order to define the ultimate capture zone envelope and to better delineate the time-related capture zone perimeter in the proximity of the stagnation point.

[5] An automatic delineation can be useful when defining capture zones in practical applications, but it can also be successfully applied when defining capture zones with stochastic methods, if a high number of simulations is required. The aim of this work is to present an Automatic Protection Area (APA) delineation algorithm, a new method for the identification of the particles starting positions, and an automatic delineation of the capture zone perimeter. The Automatic Protection Area method consists of three steps, implemented into subalgorithms referred as APA-I through APA-III (Figure 1). The preprocessing algorithm (APA-I) estimates the position of the stagnation points and places circles of forward particles around them, while backward particles are located around the pumping wells. The output of the two particle tracking simulations are processed by an intermediate-processing algorithm (APA-II), in order to define new starting positions for backward particles around each pumping well, which are not equally spaced. The postprocessing algorithm (APA-III) interpolates the pathlines and automatically delineates closed capture zone perimeters for fixed traveltimes. APA-III is implemented in order to exclude from the capture zone all the points with a traveltime higher than the fixed one, and to maximize the capture area.

[6] The APA algorithm runs under 2D geometry in the presence of anisotropic and nonhomogeneous domains, steady state flow conditions, and multiple pumping wells. Further extensions and modifications of the procedure are required for it to be suitable in a 3D geometry.

2. APA Algorithm

2.1. Preprocessing Algorithm (APA-I)

[7] The APA-I algorithm identifies the stagnation points, and places circles of equally spaced forward particles around them, while a circle of equally spaced backward particles is placed around each pumping well. It also calculates the most suitable radius for each circle of particles around the wells and the stagnation points.

[8] The position of the stagnation points can be determined on the basis of the discharge vector Q, or the mean velocity vector v, searching for points of the model domain where their modulus is equal to zero:

$$\sqrt{v_x^2 + v_y^2} = 0 \tag{1}$$

where v_x is the component of the flow velocity along the x axis and v_y is the flow velocity component along the y axis. The stagnation point is therefore numerically identified by looking for a local minimum in the discharge field, or in the flow velocity field (Figure 2) using a minimization procedure based on the Nelder-Mead search method [*Nelder and Mead*, 1965]. However, this procedure is not suitable for weak sinks (a sink that does not capture all the flow entering the cell), when the stagnation point is located inside the well cell itself, and thus it cannot be identified. The problem can be solved by refining the grid of the model domain in the proximity of the weak wells. Once found, stagnation points are automatically associated to their well: the user visually chooses an approximation of the stagnation points on the map reporting the flow velocity, and the algorithm builds a local subset of the domain inside which the minimum search is performed. The described way of searching for the stagnation points is alternative to the one proposed by *Bakker and Strack* [1996], in which stagnation points are identified looking for velocity minima along the pathlines discharging into a pumping well. This latter method has the drawback of requiring an overly fine grid discretization in order to be effective.

[9] The most suitable radii of the particle circles for wells and stagnation points are automatically calculated by the APA-I algorithm. The choice of the circle radius around each stagnation point is based on the model domain discretization. In order to ensure that a sufficient number of forward particles will move in every direction, and in particular from the stagnation points to the pumping wells, the radius of the circle r_0 is set equal to the diagonal of the stagnation cell:

$$r_0 = \sqrt{\Delta x_0^2 + \Delta y_0^2} \tag{2}$$

where Δx_0 is the dimension of the stagnation cell along the x axis, and Δy_0 is the dimension of the stagnation cell along the y axis.

[10] The particle tracking code developed by *Pollock* [1994] employs flow velocities derived from the discharge values calculated at the cell interfaces. As a consequence, the flow velocities v_x and v_y are defined at every cell face, not in the center of it, and are linearly interpolated within each cell (Figure 2). For this reason, in strong sink cells v_x and v_y change linearly from positive (negative) values to negative (positive) values, and the modulus of the flow velocity is fictitiously decreasing from the cell boundaries to a point where it is equal to zero. In this paper we define such a point as the “shadow point” S of a strong sink. If a circle of particles is located around a pumping well and traced backwards, the pathlines inside the well cell will diverge

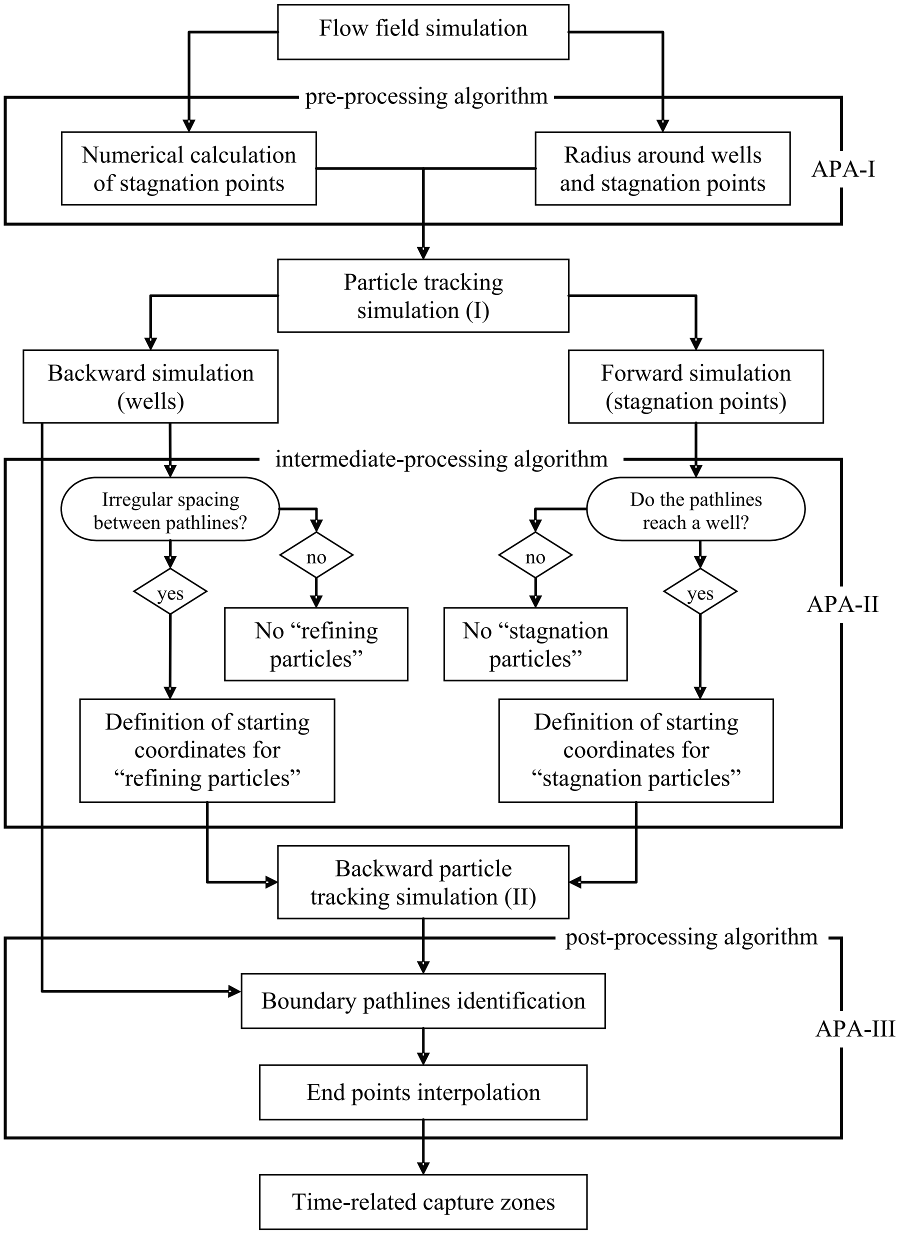

Flow field simulation
pre-processing algorithm
Numerical calculation of stagnation points
Radius around wells and stagnation points
APA-I
Particle tracking simulation (I)
Backward simulation (wells)
Forward simulation (stagnation points)
intermediate-processing algorithm
Irregular spacing between pathlines?
no
yes
No "refining particles"
Do the pathlines reach a well?
no
yes
No "stagnation particles"
APA-II
Definition of starting coordinates for "refining particles"
Definition of starting coordinates for "stagnation particles"
Backward particle tracking simulation (II)
post-processing algorithm
Boundary pathlines identification
APA-III
End points interpolation
Time-related capture zones

Figure 1. Flowchart for the structure of the APA algorithm.

from the shadow point, rather than from the well. If the radius of the circle is too small and the shadow point is located outside the circle, the pathlines will pass through the pumping well and the capture area will be meaningless. For these reasons, the APA-I algorithm places the center of the circle above the pumping well, and defines its radius as twice the distance between the well and the shadow point (Figure 2):

$$r_W = 2 \cdot d_S = 2 \cdot \sqrt{(x_W - x_S)^2 + (y_W - y_S)^2} \qquad (3)$$

where r_w is the radius of the circle around the well W, d_S is the distance between the well W and the shadow point S, x_w and y_w are the coordinates of the pumping well, x_s and y_s are the coordinates of the shadow point. We observe that, as a linear interpolation of flow velocity leads to lower values inside the sink cells, a more correct distribution of backward particles could allocate them at the cell boundaries. In this way we can avoid incorrect calculations of flow velocities along the pathlines, although it is common practice to use circles of particles inside the well cells.

[11] If the APA method is to be extended to a 3D geometry, it is necessary to look for a stagnation point in every layer of the model domain, which would generate a ''stagnation line'' associated to each pumping well.

2.2. Intermediate-Processing Algorithm (APA-II)

[12] After the position and radius of each circle have been determined, the particles around each pumping well are traced backwards, while the circles around each stagnation point are traced forwards. The results are used to define new starting positions of additional particles, which are then traced in the second particle tracking simulation. The

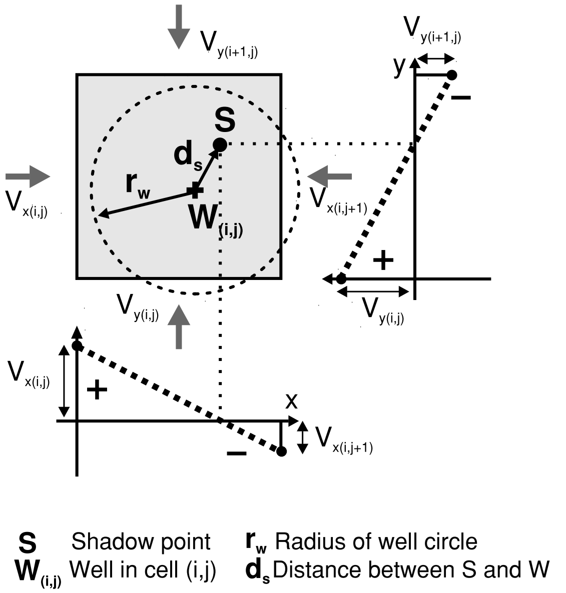


Figure 2. Preprocessing algorithm (APA-I). Identification of the radius around pumping wells: flow velocity vectors in well cells and "shadow point" generated by linear interpolation of the flow velocity.

intermediate algorithm structure consists of the following steps.

[13] 1. Finding which of the forward particles originated around a stagnation point reach a pumping well.

[14] 2. Defining the intersection of the pathlines with the circle around the reached well, in order to use these as starting points for new backward particles around the wells (the "stagnation particles").

[15] 3. Identifying the starting positions of further "refining particles", when the calculated backward pathlines are too distant.

[16] As for the forward pathlines originated around the stagnation points, some of them reach a pumping well, some of them do not (Figure 3a). Furthermore, in case of multiple pumping wells, some particles originated from the stagnation point of the well W_A can be captured by the well W_B, if the capture area of W_B includes the capture area of W_A. For particles reaching a well W, the APA algorithm calculates the intersection between the forward pathline and the circle of radius r_w around W. In the second particle tracking simulation, this point is used as the starting point of a stagnation particle. In case of strong sinks, most of the semianalytical particle tracking codes (e.g., MODPATH) stop the particles from entering a strong sink cell at the boundary of the cell itself. In this case the last point of the pathline is connected to the well point, and the intersection between this line and the circle is considered. Further particles are also located near the stagnation particles, in order to define a higher number of pathlines close to the boundary of the capture area. In the work of *Bakker and Strack* [1996], on the contrary, only two particles are added close to the ultimate capture zone envelope which had been previously determined with four particles released around the stagnation point.

[17] As for the backward pathlines leaving the well (the well particles), they are processed in order to define a second group of additional particles, the refining particles. Sometimes (e.g., in the case of strong heterogeneous aquifers, or overlapping capture areas) the distance between adjacent pathlines inside the capture area may be too large, thus leading to an inaccurate delimitation of the time-related capture zones. One or more refining particles are added when the distance between two adjacent pathlines of the first backward simulation is higher than a fixed value (Figure 3b).

[18] Backward tracking from stagnation points is not considered in the process because it would not provide "complete" pathlines from the pumping well. Backward pathlines from stagnation points would have to be combined with the forward segment starting from the same point, but

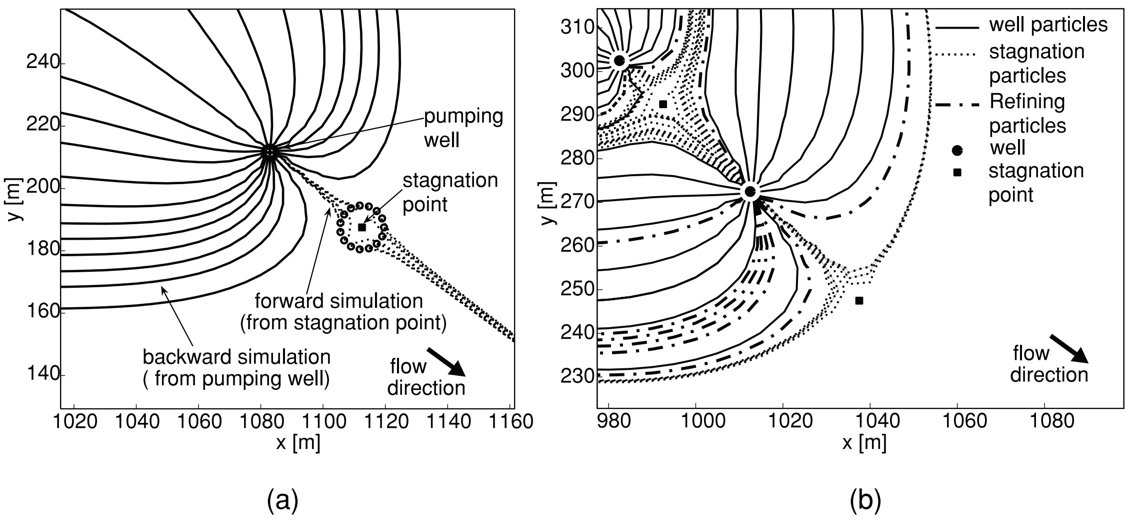


Figure 3. Particle tracking simulations: (a) First simulation, with backward pathlines from pumping wells and forward pathlines from stagnation points. (b) Second simulation: well, stagnation and refining particles.

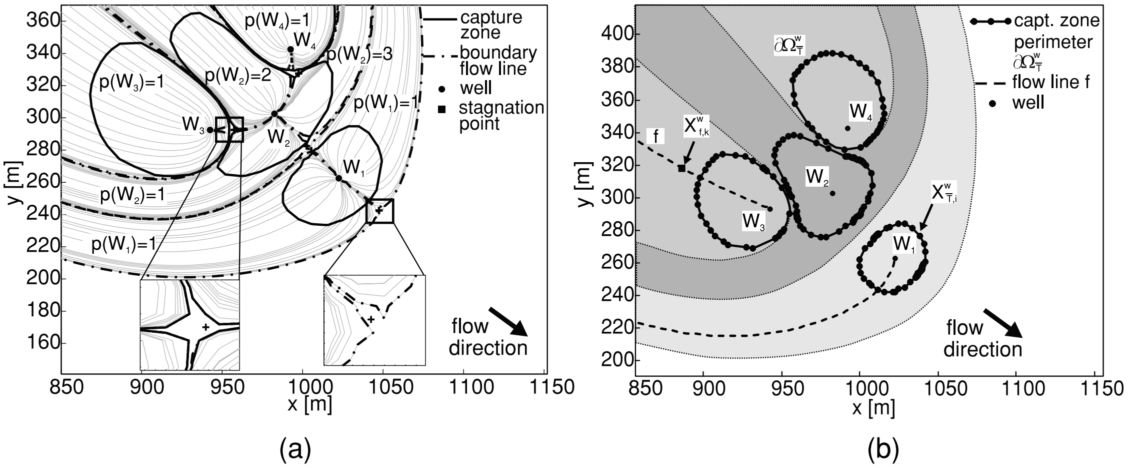


Figure 4. Post processing algorithm (APA-III). (a) Identification of boundary flow lines. (b) Discretized flow lines and capture zones perimetration.

the results would not be completely consistent with backward circles starting from the pumping well: semianalytical particle tracking codes, like MODPATH, stop the forward particles at the boundary of the strong sink cells. From a computational point of view, it is simpler to define a new starting point around the well for each pathline which flows near the stagnation point, and to trace it backwards.

2.3. Post-Processing Algorithm (APA-III)

[19] When the starting points of the stagnation particles and the refining particles have been determined, the second backward particle tracking simulation is performed. The output is post-processed by the APA-III algorithm, in order to automatically delineate the time-related capture zones. The APA-III is structured so that the perimeter of the capture zone for a given time neither intersects any flow line nor includes or excludes any point with a traveltime respectively higher or lower than the fixed one. This topic is an improvement to the work of *Bakker and Strack* [1996], who dynamically allocated some particles during the delineation of the capture zone perimeter, when the existing pathlines are too distant, but did not use controls regarding the intersection between the pathlines and the perimeter, or the inclusion or exclusion of points with traveltimes higher or lower than the fixed one.

[20] The capture area $\omega_{\bar t}^{w}$ of a generic pumping well *w*, and its boundary $\partial_{\bar t}^{w}$, for a traveltime $\bar t^{w}$, can be identified as

$$\omega_{\bar t}^{w} = \left\{x_t^w : t \le \bar t\right\} \quad \text{with} \quad t \in R^+, x_t^w \in R^2 \tag{4a}$$

$$\partial\omega_{\bar t}^{w} = \left\{x_t^w : t = \bar t\right\} \text{ with } \mathrm{t} \in \mathrm{R}^+, \mathrm{x}_\mathrm{t}^\mathrm{w} \in \mathrm{R}^2 \tag{4b}$$

where x_t^w is a generic point of the model domain inside the steady state capture area of the pumping well *w*. A water particle that moves along flow lines starting from x_t^w reaches *w* after a time of travel equal to *t*.

[21] For multiple pumping wells, the total capture area at time *t* is defined as

$$\omega_t = \bigcup_w \omega_t^w \quad \text{with} \quad w \in \{1, \cdots, n_w\} \tag{5}$$

where n_w is the number of pumping wells. If the capture area of a pumping well *w* includes the capture area of another one, it can be further divided into a number of partitions (Figure 4a). From a mathematical point of view, a partition of a set is a division into a number of nonempty and nonoverlapping subsets, which cover the whole set. In this article, the term partition identifies the largest subarea of a capture zone in which, for any couple of flow lines, all the flow lines that can be traced between them reach the pumping well *w*. The number of partitions $n_p(w)$ for a capture area of a pumping well *w* is equal to $n_p(w) = \overline{n_S} + 1$, where $\overline{n_S}$ is the number of stagnation points (except the one associated to *w*) close to the flow lines of the well *w*. In this way, if the capture area of a well does not include the capture area of another well (e.g., W_3 and W_4 in Figure 4a), the number of its partitions will be $n_p(w) = 1$, while in case of included wells it will be $n_p(w) > 1$ (e.g., W_2, with $n_p(w_2) = 3$). The area of a generic partition *p* of a pumping well *w*, for a time of travel *t*, is identified as $\omega_{p,t}^{w}$, and its boundary is $\partial\omega_{p,t}^{w}$. The capture area of a pumping well *w* at the time *t* is the union of all the $n_p(w)$ partitions belonging to the pumping well:

$$\omega_t^w = \bigcup_p \omega_{p,t}^w \tag{6}$$

[22] Although partitions, capture areas and their boundaries can be defined as continuous regions of the model domain, their delineation with numerical models and algorithms requires a discrete definition. In this paper, discrete capture areas and their boundaries are identified using uppercase letters (Ω and ∂Ω). The capture zone perimeter is defined on the basis of the discretized flow lines calculated in the second particle tracking simulation. Flow lines are classified depending on the origination well and the partition they belong to. "Outer" and eventually "inner" boundary flow lines are identified for each pumping well (Figure 4a). They define the perimeter of the partitions. Inner boundary flow lines are generated in case of multiple wells, when the capture zone of each pumping well includes the capture zone of another one. Therefore the APA algorithm identifies the following kinds of flow lines.

[23] 1. Outer boundary flow lines, two for each pumping well. They represent the two lines that flow the closest to the stagnation point of the well they belong to.

[24] 2. Inner boundary flow lines, two for each included pumping well. They represent the two flow lines of the including well, that flow the closest to the stagnation point of the included well (respectively, W_2 and W_3 in Figure 4a).

[25] 3. Inner flow lines, all the others.

[26] Every discrete steady state capture area is made up of a finite number of simulated flow lines. Flow lines calculated with particle tracking simulations result in a discretized set of points $X_{p,f,T}$, where

[27] 1. w is the well identifier: $w \in \{1,\ldots, n_w\} w \in \mathrm{N}$.

[28] 2. p is the partition identifier; partitions are numbered for each pumping well: $p \in \{1,\ldots, n_p(w)\} p \in \mathrm{N}$

[29] 3. f is the flow line identifier; flow lines are numbered for each partition:$f \in \{1,\ldots, n_f(w, p)\}$, $f \in \mathrm{N}$ $n_f(w, p) = N_f$ will be used for simplicity, when the corresponding well can be clearly identified. The flow lines are ordered clockwise, so that the boundary flow lines of a partition p are identified by $f = 1$ and $f = N_f$.

[30] 4. T is an element of the sequence of times of travel related to the points $X^w_{f,T} : T \in \{0,\ldots, T_{\max}\}$, $T \in R^+$

[31] Every point $X^w_{p,f,T}$ is related to a traveltime T, which defines the time required for a particle located in $X^w_{p,f,T}$ to reach the pumping well w. The finite sequence of these times ranges from zero (starting points of particles around the pumping wells) to a maximum time $T_{\max}$, which represents the time of a backward particle tracking simulation at which the last particle exits the model domain (or reaches an inner flow sink). In addition, discrete capture areas are calculated at fixed traveltimes $\overline{T} : \overline{T} \in \{\overline{T}_1 ,\ldots,\overline{T}_N\}$, where N is the number of traveltimes at which capture areas are calculated. The points with a time of travel equal to $\overline{T}$ are identified as $X^w_{p,f,\overline{T}}$.

[32] The capture zone boundary $\partial\Omega^{w}_{p,\overline{T}}$ of a partition p, for a traveltime $\overline{T}$, is identified using a sequence of points $X^w_{p,\overline{T},i}$. The $X^w_{p,\overline{T},i}$ are ordered clockwise, starting from the stagnation point of the pumping well, and are numbered for $i = 1,\ldots, N_i (w,p, \overline{T})$. The boundary $\partial\Omega^{w}_{p,\overline{T}}$ is defined using an interpolation of the $X^w_{p,\overline{T}i}$: it is a closed perimeter, so that $X^w_{p,\overline{T}1} \equiv X^w_{p,\overline{T}Ni+1}$ and N_i distinct points are considered. If a linear interpolation is used, the boundary is defined by a series of N_i straight segments $l^w_{p,\overline{T},i}$ connecting adjacent points $X^w_{p,\overline{T},i}$ and $X^w_{p,\overline{T},i+1}$. To generalize, a cubic spline interpolation for closed curves can be used, e.g., the cubic spline interpolation described by *Bartels et al.* [1987]. In this case, the interpolating curve is expressed parametrically, and the coordinates x and y of the partition boundary are calculated separately. In both cases, the boundary of the partition is defined as

$$\partial\Omega^w_{p,\overline{T}} = \bigcup_i l^w_{p,\overline{T},i} \qquad (7)$$

[33] As the complexity of the flow field increases, the complexity of the interpolation algorithm increases, and the identification of the points $X^w_{p,\overline{T},i}$ requires more calculations. Further details are listed in the Appendix.

[34] The APA-III algorithm is designed to work in a 2D geometry. If the method is to be extended to the 3 dimensions, this last discussed part should be also extended, considering intersections of the traced flow lines not with segments connecting the end points, but with triangles, and a new ordering algorithm should be used.

3. Applications and Comparison

[35] The APA algorithm was applied to a set of synthetic test cases, and the results were compared to the output of the "classic" particle tracking method. In particular, the procedure was applied to the following models.

[36] Case 1: one pumping well in a confined, homogeneous, isotropic aquifer, with the main flow direction along the y axis.

[37] Case 2: four pumping wells in a confined, homogeneous, isotropic aquifer.

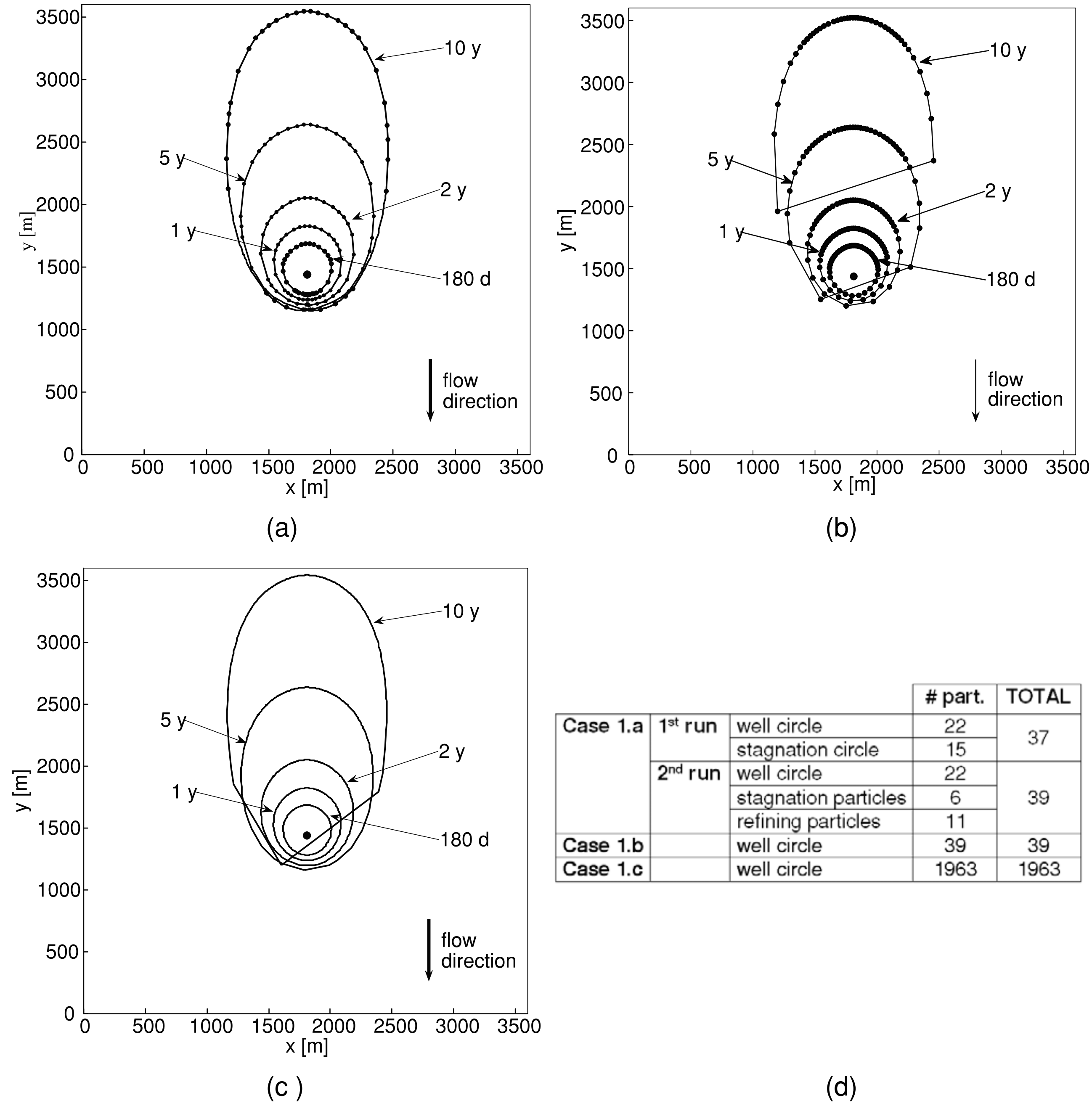
[38] Case 3: one pumping well in a confined, nonhomogeneous, isotropic aquifer, with the main flow direction oriented along the diagonal of the domain.

[39] The model domain is a 3600 m × 3600 m square domain, divided into 144 square 25 m × 25 m cells. The origin of the axes is located in the lower left corner. The aquifer is 10 m thick. In case 1 and 2, the main flow direction is opposed to the y axis, i.e., from north to south, and the flow boundary conditions are two Dirichlet conditions, applied at the upper boundary (300 m constant head) and at the lower boundary (240 m constant head), resulting in a regional gradient equal to $1.67 \cdot 10^{-3}$. The left and the right boundaries are no flow boundaries. In case 3, four Diriclet conditions are applied at the boundaries, and a linearly changing constant-in-time head is applied along them. In case 1 and 3, the pumping well is located at x = 1812.5 m, y = 1437.5 m. The pumping rate is equal to $2.0 \cdot 10^{-2}$ m^3/s. The steady state flow field was simulated by MODFLOW 2000 [*Harbaugh et al.*, 2000], the backward particle tracking by the APA algorithm and MODPATH [*Pollock*, 1989, 1994]. The time-related capture zones were reconstructed for five traveltimes (180 d, 1, 2, and 5 years in all cases, 10 years in cases 1 and 2, 8 years in case 3).

[40] The results obtained using the APA algorithm are presented and compared to the capture areas calculated using equally spaced particles around the pumping wells. For each case study, three different capture zone encirclements are presented. The first one is the result of the APA algorithm (cases 1.a, 2.a, and 3.a), while the second and the third ones employ equally spaced particles from pumping wells and simply connect the end points at the fixed traveltime (the $X^w_{p,\overline{T},i}$ are simply defined as in equation (A1), see Appendix). More in details, the second capture zone encirclement employs the same number of particles defined in APA-II, (cases 1.b, 2.b and 3.b) and the third one uses a number of equally spaced particles which corresponds to the minimum spacing between particles of the APA algorithm (cases 1.c, 2.c, 3.c), typically in the proximity to the stagnation point. The number of the different kinds of particles used for each case is reported in Figure 5d, 6d and 7d. The computational time for each case is reported in Table 1.

3.1. Case 1

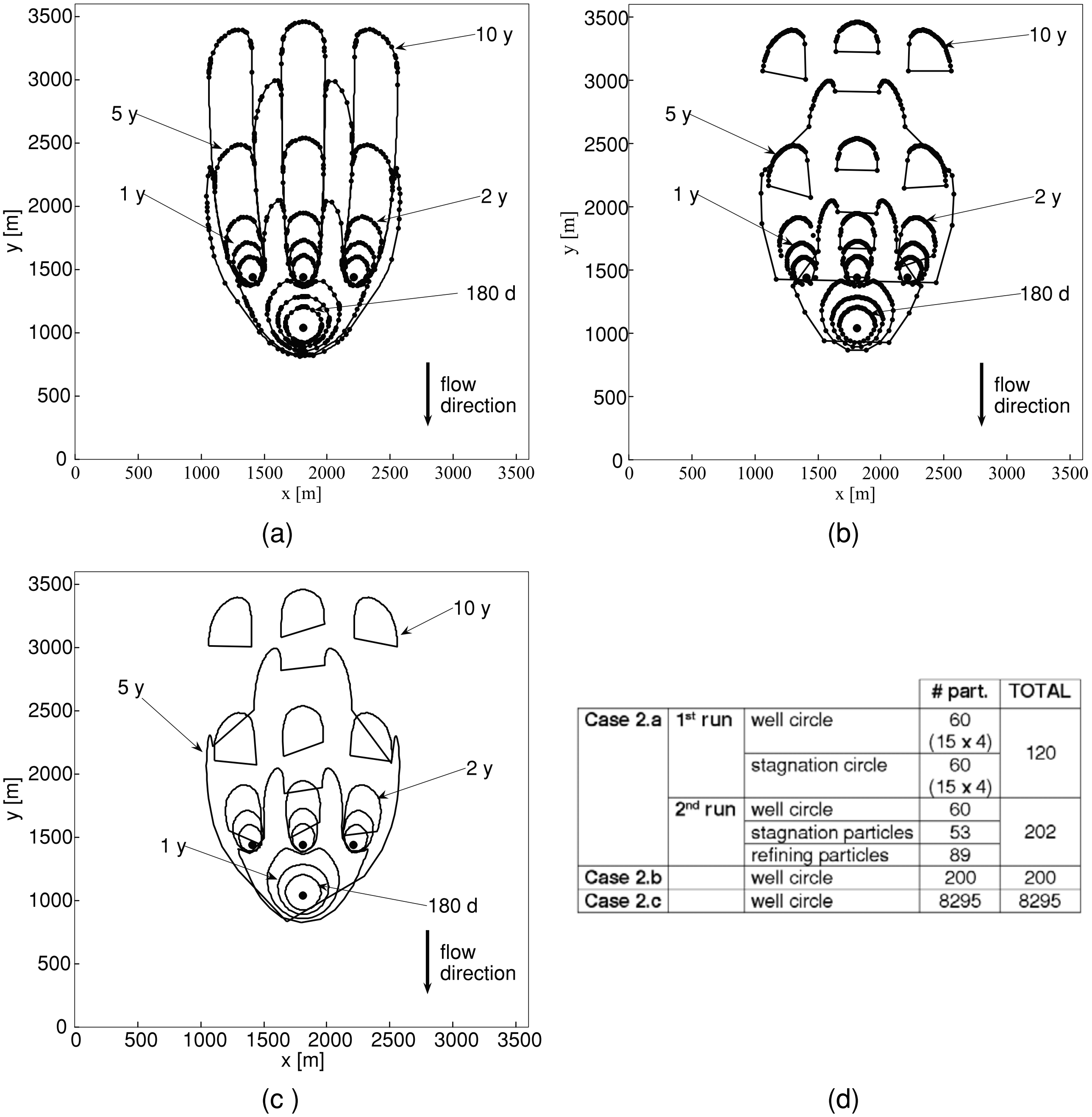
[41] The time-related capture areas, for the traveltimes described above, are calculated in an homogeneous, isotropic aquifer, with an hydraulic conductivity equal to



			# part.	TOTAL
Case 1.a	1st run	well circle	22	37
		stagnation circle	15	
	2nd run	well circle	22	39
		stagnation particles	6	
		refining particles	11	
Case 1.b		well circle	39	39
Case 1.c		well circle	1963	1963

Figure 5. Case 1: time related capture areas for 180 d, 1, 2, 5, and 10 years, for a single pumping well in a confined, homogeneous, isotropic aquifer. (a) Case 1.a: capture areas for the APA algorithm. (b) Case 1.b: capture areas for a circle of 39 equally spaced particles. (c) Case 1.c: capture areas for a circle of 1963 equally spaced particles. (d) Number of particles used in Case 1.a, 1.b, and 1.c.

$7.0{\cdot}10^{-5}$ m/s. Capture areas delineated using the APA algorithm (Case 1.a, Figure 5a) were determined by locating 22 particles around the well and 15 around the stagnation point, which results in the location of a total amount of 39 particles in the second MODPATH run (Figure 5d). As a comparison, the time related capture areas for the same traveltimes were calculated using a circle of 39 (Case 1.b, Figure 5b) and 1963 (Case 1.c, Figure 5c) equally spaced particles around the pumping well. The results show that the APA algorithm allows a good delineation of the capture zone downstream of the pumping well, for every traveltime. The flow line interpolation defines regular and smoothed perimeters. The identification of boundary flow lines is absolutely necessary: for long traveltimes, the capture zones obtained in Case 1.b and 1.c move upstream and do not encompass the pumping well, although the minimum spacing between particles is the same as in the APA algorithm. In this case, even if a very high number of particles is used, a (significant) part of the capture zone is neglected (i.e., many points with a time of travel lower than the fixed one are excluded), resulting in an incorrect capture zone encirclement.



			# part.	TOTAL
Case 2.a	1st run	well circle	60 (15 x 4)	120
		stagnation circle	60 (15 x 4)	
	2nd run	well circle	60	202
		stagnation particles	53	
		refining particles	89	
Case 2.b		well circle	200	200
Case 2.c		well circle	8295	8295

Figure 6. Case 2: time related capture areas for 180 d, 1, 2, 5, and 10 years, for four pumping wells in a confined, homogeneous, isotropic aquifer. (a) Case 2.a: capture areas for the APA algorithm. (b) Case 2.b: capture areas for four circles of 50 equally spaced particles. (c) Case 2.c: capture areas for four circles of equally spaced particles, for a total amount of 8295 particles. (d) Number of particles used in Case 2.a, 2.b, and 2.c.

3.2. Case 2

[42] The time related capture areas, for traveltimes equal to 180 d, 1, 2, 5, and 10 years, are calculated in a homogeneous, isotropic aquifer, with an hydraulic conductivity equal to $7.0 \cdot 10^{-5}$ m/s, and with four pumping wells. Three pumping wells (W_1, W_2, and W_3) are lined up perpendicularly with respect to the flow direction; the fourth well (W_4) is located downgradient. The distance between the wells is 400 m, and their coordinates and discharge rates are:

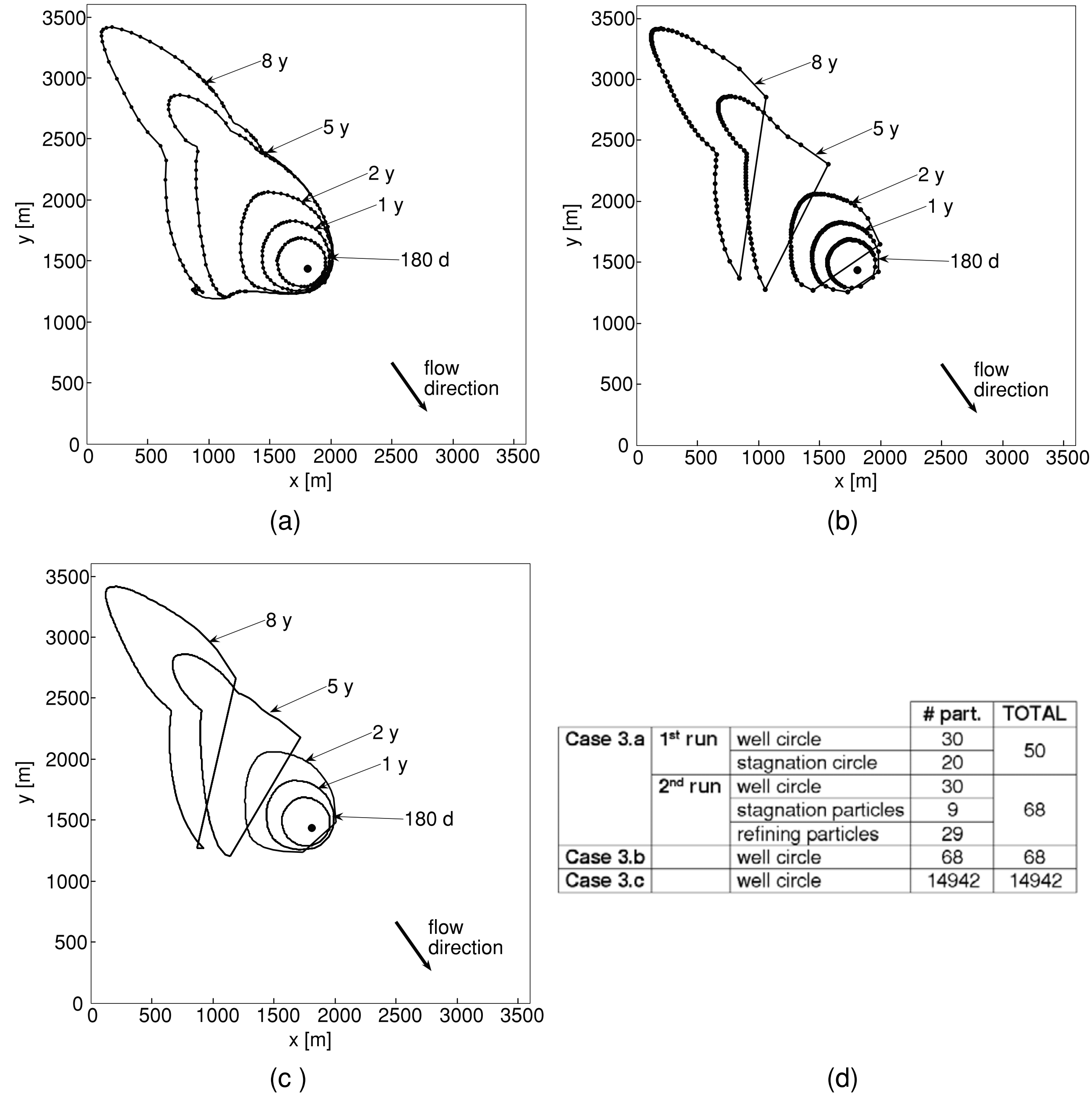
[43] well W_1: $x = 1812.5$ m, $y = 1437.5$ m, $Q = 5.0 \cdot 10^{-3}$ m^3/s;

[44] well W_2: $x = 1412.5$ m, $y = 1437.5$ m, $Q = 5.0 \cdot 10^{-3}$ m^3/s;

[45] well W_3: $x = 2212.5$ m, $y = 1437.5$ m, $Q = 5.0 \cdot 10^{-3}$ m^3/s;

[46] well W_4: $x = 1812.5$ m, $y = 1037.5$ m, $Q = 1.0 \cdot 10^{-2}$ m^3/s;

[47] The total discharge is equal to $2.5 \cdot 10^{-2}$ m^3/s. Capture areas delineated with the APA algorithm were calculated using circles of 15 particles around each pumping well



			# part.	TOTAL
Case 3.a	1st run	well circle	30	50
		stagnation circle	20	
	2nd run	well circle	30	68
		stagnation particles	9	
		refining particles	29	
Case 3.b		well circle	68	68
Case 3.c		well circle	14942	14942

Figure 7. Case 3: time related capture areas for 180 d, 1, 2, 5 and 8 years, for a single pumping well in a confined, nonhomogeneous, isotropic aquifer, flow direction along the diagonal of the model domain. (a) Case 3.a: capture areas for the APA algorithm. (b) Case 3.b: capture areas for a circle of 68 equally spaced particles. (c) Case 3.c: capture areas for a circle of 14942 equally spaced particles. (d) Number of particles used in Case 3.a, 3.b and 3.c.

Table 1. Run Times for the Test Cases, Determined Using a Pentium IV 3.2 GHz PC

		Case 1		Case 2		Case 3	
		Run Time, s	TOTAL, s	Run Time, s	TOTAL, s	Run Time, s	TOTAL, s
Case X.a: APA	APA-I + APA-II	2.5	15.5	6.0	70.1	3.5	32.1
	MODPATH	5		5		5	
	APA-III	8.1		59.1		23.6	
Case X.b: few equally spaced particles	MODPATH	5	11.2	5	28.2	5	16.5
	perimeter delineation	6.2		23.2		11.5	
Case X.c: many equally spaced particles	MODPATH	21	419.5	55	902.8	74	2438
	perimeter delineation	398.5		852.8		2364	

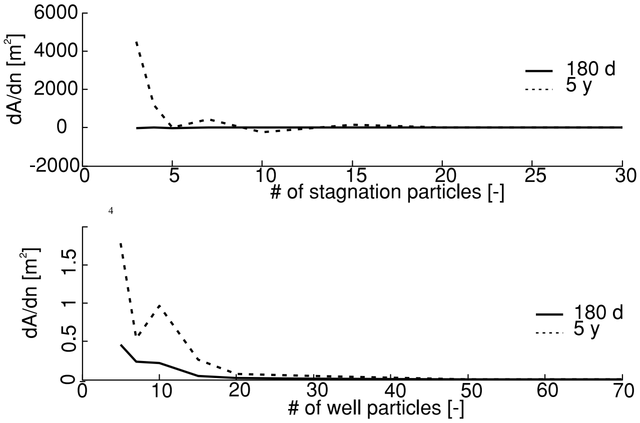


Figure 8. Variation in the enclosed capture area as a function of the number of particles traced from the well and the stagnation point, for short (180 d) and long (5 years) traveltimes. The results are calculated for the application of Case 3.

and each stagnation point. The routine resulted in a total number of 202 particles for the second MODPATH run (Figure 6d). Figure 6a shows that the APA algorithm (Case 2.a) is able to accurately define the capture zone perimeters for all the pumping wells, and in particular for the well which is located furthest downgradient (W_4, whose capture area encompass the other ones). The identification of stagnation points for the encompassed wells leads to the definition of additional stagnation particles, which originate around W_4 and run near the stagnation points of the others. Furthermore, the APA algorithm identifies inner boundary flow lines for W_4 (a couple of flow lines for each encompassed well). In Figures 6b and 6c, the capture areas calculated with equally spaced particles are presented. They were determined using circles of 50 equally spaced particles around each pumping well (Case 2.b, Figure 6b), and using circles with a uniform spacing equal to the minimum spacing of each well, for a total amount of 8295 particles (Case 2.c, Figure 6c). In both cases, for long traveltimes the capture area of W_4 is overestimated, and overlaps the others. The capture area extends on the other ones, including all the wells, and its boundary is therefore meaningless, while many points downgradient the pumping well are neglected, although they are associated to a traveltime lower than the fixed one.

3.3. Case 3

[48] The time related capture areas, for the traveltimes described above, are calculated in a nonhomogeneous, isotropic aquifer. The domain is divided into nine (3 × 3) regions of 1200 m × 1200 m, which alternate two hydraulic conductivity values of $K_1 = 2.0 \cdot 10^{-4}$ m/s (in the four corners and in the middle of the domain) and $K_2 = 5.0 \cdot 10^{-5}$ m/s. The flow direction is along the diagonal of the model domain (and, as a consequence, of the cells). 30 backward particles around the well and 20 forward particles around the stagnation point were used; a total amount of 68 particles is obtained for the second MODPATH run (Figure 7d). The minimum space between particles located by the APA algorithm corresponds to a circle of 14,942 equally spaced particles, used in Case 3.c. The results are shown in Figure 7. As shown in the previous cases, increasing the time and regardless of the number of particles used, the capture zone encirclement fails to include points with traveltimes lower than the fixed one.

4. Discussion and Conclusions

[49] The aim of this paper is to present a method for an automatic delineation of capture zones, working with any finite difference flow and particle tracking solvers (in this case, MODFLOW and MODPATH were used). For this reason, the APA methodology can be considered an absolutely general algorithm for an automatic capture zone delineation, while other methods are always related to a specific tracking code [e.g., *Bakker and Strack*, 1996]. In addition, it gives an automatic perimetration, whereas, if many equally spaced particles are used, a post-run manual perimetration is very time-demanding.

[50] The equal spacing of starting positions requires a large amount of particles for a good delineation downgradient the pumping well. The test cases show that the computational time required by the APA algorithm is almost equal to the time consumed by a MODPATH run of a number of equally spaced particles corresponding to the minimum space between the particles allocated by the APA algorithm (Table 1, the times refer to a Pentium IV 3.2 GHz PC). Furthermore, if many equally spaced particles are used, the time required for a manual capture zone encirclement, or for simply connecting the end points with an external code, should be added. This can be relevant and much higher than the APA + MODPATH running time. If the APA algorithm was to be embedded in the same software which calls the MODFLOW and MODPATH runs, and treats their output, its time efficiency could be further improved.

[51] The application of the APA algorithm showed that, in APA-I, a number of 20 to 30 particles for every circle (around each well and stagnation point) is sufficient to obtain good quality results from the algorithm. A lower number (e.g., 10) may lead to quite irregular perimeters, while a higher number is unnecessary and does not significantly improve the smoothness of the area. An example of the changes in the capture areas as the number of particles in the well and stagnation circles increases is shown in Figure 8, which refers to the test Case 3. Similar results can be obtained in almost every application.

[52] The APA algorithm presented in this work is based on the identification of the stagnation points. Better starting positions of the particles around the pumping wells are identified, in order to obtain an adequate number of flow lines in the proximity of the stagnation points. Furthermore, a postprocessing algorithm for the interpolation of the stop position of particles at the fixed traveltime is proposed, in order to get automatic smoothed perimeters. For a fixed traveltime $\overline{T}$, the algorithm maximizes the capture area, avoiding any intersection of the perimeter with the flow lines, neither for $T < \overline{T}$ nor for $T > \overline{T}$. The points of the flow lines included in the capture zone perimeter always have traveltimes $T \leq \overline{T}$, so that the definition of time-related capture zone is obeyed. The proposed method shows that the identification of boundary flow lines is extremely important for correct capture perimeter delineation. If no

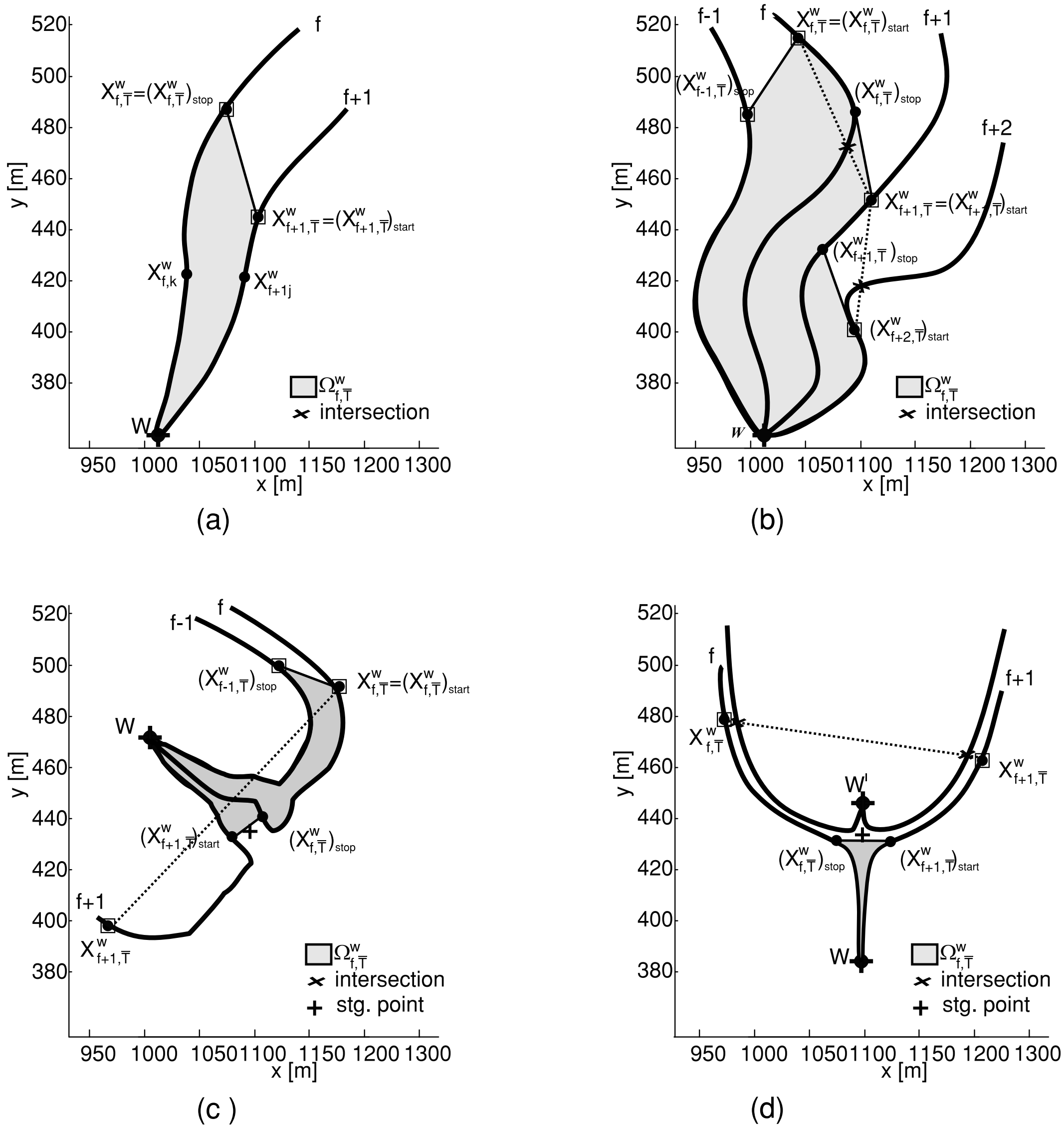


Figure A1. Interpolation schemes: (a) Interpolation in case of no intersection. (b) Interpolation in case of inner flow lines and one or more intersections. (c) Interpolation in case of outer boundary flow lines. (d) Interpolation in case of inner boundary flow lines between different partitions.

boundary flow lines are used, the time-related capture area is irregular and, for long traveltimes, it does not encompass the sink, and can neglect large regions with $T < \overline{T}$. Errors become more evident when the capture areas approach to the ultimate capture zone envelope. Furthermore, in case of multiple pumping wells and encompassing capture zones, the wrapping area perimeter can be meaningless. The identification of stagnation points allows allocating further particles (stagnation particles) which define the capture area downgradient of the pumping well. In addition, the interpolating postprocessing algorithm avoids the intersections between the capture zone perimeter and the flow lines, or between capture zones of different pumping wells.

[53] The algorithm is here presented for 2D applications, but it can be extended to 3D geometry. The APA method is not suggested if stagnation points cannot be univocally identified, which means that pure radial flow and weak sinks wells cannot be solved. For weak sinks, a grid refining can overcome the problem. As for radial flow, this condition is not very common, and the good quality results of the perimetration methods could justify its use in the other cases.

Appendix A: Insight of APA-III Interpolation Algorithm

[54] The APA-III algorithm defines the closed boundary of time-related capture zones in a completely automatic way, avoiding intersections of the perimeter with flow lines for traveltimes both shorter and longer than the fixed one. It also includes inside the boundary only points with a traveltime lower than or equal to the time at which the capture area is designed. According to equation (4b), the simplest choice for the X_p, $\overline{T},^{w}_{i}$ would be

$$X^{w}_{p,\overline{T},i} \in \left\{X^{w}_{p,f,T} : T = \overline{T}\right\} \qquad \text{(A1)}$$

[55] In this case, $N_i = N_f$. Results can be satisfactory only in case of homogeneous isotropic aquifers and short traveltimes (i.e., if the capture zone boundary has not yet reached the proximity to the stagnation point). In the other cases, if the $X^w_{p,\overline{T},i}$ are chosen in this same way and directly connected without further controls, the capture zone perimeter intersects the flow lines and, for quite long times of travel, would not include the pumping well itself (see Figures 5b–5c, 6b–6c, and 7b–7c for equally spaced circles of backward particles). Such intersections have no physical meaning. They are expected near boundary flow lines, if the capture zone perimeter is traced close to the stagnation point, and in case of abrupt changes in the conductivity field.

[56] The APA scheme employs an algorithm to maximize the protection area avoiding intersections. In order to accomplish this task, the APA-III calculates the points of equation (A1) and works on couples of flow lines, distinguishing different cases (Figure A1). The boundary $\partial\Omega^w_{p,\overline{T}}$ is identified by groups of points resting on flow lines (one or more points for every flow line). The ordered clockwise sequence of $X^w_{p,\overline{T},i}$ starts on f at a point $X^w_{p,f,T_{start}}$, moves backwards or forwards along f till to the point $X^w_{p,f,T_{stop}}$, "jumps" on f + 1 at $X_{p,f+1,T_{start}}^{\,w}$, and so on (Figure A1). Besides, for every couple of flow lines f and f + 1, the algorithm identifies $X^w_{p,f,T_{stop}}$ on f and $X^w_{p,f+1,T_{start}}$ on f + 1. The interpolation (in this paper, linear interpolation) between them guarantees that the "slice" of capture area between the two flow lines is also the largest one, if intersections are avoided and points with a time of travel higher than $\overline{T}$ are excluded. The times T_{start} and T_{stop} are identified by the interpolation algorithm and are different for every flow line and $\overline{T}$ at which the capture zone is calculated.

[57] For each couple of flow lines f and f + 1, belonging to the same well w and partition p, the algorithm defines:

[58] $\Omega^w_{f,\overline{T}}$, the local "slice" of capture area between flow lines f and f + 1, which maximizes the area and avoids intersections. It will be added to the capture area $\Omega^w_{p,\overline{T}}$. In order to use an easier notation, when dealing with local capture areas and their boundaries, the index p is removed and replaced by f, because the algorithm always works on f and f + 1, and the partition is always the same for the two flow lines.

[59] ● $\partial\Omega^w_{f,\overline{T}}$, the corresponding piece of capture zone boundary. As a rule, $\partial\Omega^w_{f,\overline{T}}$ is defined by the linear interpolation of the points

$$X^w_{f,\overline{T},i} \in \left\{ \left\{ X^w_{p,f,T} : T = T_{start}, \ldots, T_{stop} \right\} \cup \left\{ X^w_{p,f+1,T} : T = T_{start} \right\} \right\} \tag{A2}$$

[60] ● $\Omega^{\,w}_{f,T_1,T_2} = \Omega_{f,\overline{T}}^{\,w}$ $(X^w_{p,f,T_1}, X_{p,f+1,T_2}^{\,w})$, the generic "slice" of capture area between flow lines f and f + 1, identified by the points X^w_{p,f,T_1} and $X^w_{p,f+1,T_2}$. T_1 and T_2 are the traveltimes associated with the points that identify Ω^w_{f,T_1,T_2}, and are lower that or equal to $\overline{T}$. If $T_1 = T_{start}$ on flow line f and $T_2 = T_{stop}$ on flow line f + 1, the slice of capture area is identified as $\Omega^w_{f,\overline{T}}$.

[61] ● $I^{\,w}_{f,T_1,T_2} = I^w_f$ $(X^{\,w}_{p,f,T_1}$,$X^w_{p,f+1,T_2}$), the number of intersections between the flow line f and the line connecting X_{p,f,T_1} and $X^w_{p,f+1,T_2}$; $I^w_{f+1,T_1,T_2} = I^w_{f+1}$ $(X^w_{p,f,T_1}$, $X^w_{p,f+1,T_2}$) defines the number of intersections with the flow line f + 1. If $T_1 = T_{start}$ on flow line f and $T_2 = T_{stop}$ on flow line f + 1, the number of intersections are identified, respectively, as $I^w_{f,\overline{T}}$ and $I^w_{f+1,\overline{T}}$.

[62] The APA-III scheme distinguishes the following cases.

[63] ● Both f and f + 1, or at least one of them, are inner flow lines: the points $X^w_{p,f,\overline{T}}$ and $X^w_{p,\,f+1,\overline{T}}$ are connected with a straight line and the number of intersections with f and f + 1 (respectively, $I^w_{f,\overline{T}}$ and $I^w_{f+1,\overline{T}}$) are calculated:

[64] ● If $I_{f,\overline{T}}^{\,w} = 0$ and $I^w_{f+1,\overline{T}} = 0$, $X^w_{p,f,\overline{T}}$ and $X^w_{p,f+1,\overline{T}}$ directly define the local capture area (Figure A1a):

$$\begin{aligned} X^w_{p,f,T_{stop}} &\equiv X^w_{p,f,\overline{T}} \\ X^w_{p,f+1,T_{start}} &\equiv X^w_{p,f,\overline{T}} \end{aligned} \tag{A3}$$

[65] ● if $I^w_{f,\overline{T}} > 0$ and/or $I^w_{f+1,\overline{T}} > 0$, the algorithm searches for the couple of $X^W_{p,f,T1}$ and $X^W_{p,f+1,T2}$, with $T_1,T_2 \le \overline{T}$, that does not generate any intersection and gives the highest local area $\Omega^W_{f,T1,T2}$ (Figure A1b):

$$X^w_{p,f,T_{stop}} \equiv X^w_{p,f,T_1,},$$

$$X^w_{p,f+1,T_{start}} \equiv X^w_{p,f+1,T_2} : \begin{cases} T_1, T_2 \le \overline{T} \\ I^w_{f,T_1,T_2} = 0; I^w_{f+1,T_1,T_2} = 0 \\ \Omega^w_{f,T_1,T_2} = \max\limits_{T_1,T_2} \left(\Omega^w_{f,T_1,T_2} \right). \end{cases} \tag{A4}$$

[66] ● Both f and f + 1 are outer boundary flow lines: the points $X^w_{f,\overline{T}}$ and $X^w_{f+1,\overline{T}}$ are connected with a straight line and intersections $I_{f,\overline{T}}^{\;w}$ and $I^w_{f+1,\overline{T}}$ are calculated (Figure 5c):

[67] ● if no intersection occur, the stagnation point of the pumping well w is considered. If the stagnation point is not included in $\Omega^w_{f,\overline{T}}$, the points $X^w_{p,f,\overline{T}}$ and $X_{f+1,\overline{T}}^{\,w}$ directly define the local capture area. In this case $X^w_{p,f,\overline{T}stop}$ and $X_{p,f+1,\overline{T}start}^{\,w}$ are defined as in equation (A3).

[68] ● If any intersection occur, and/or the stagnation point S_t is included in $\Omega^w_{f,\overline{T}}$, the algorithm searches for the couple of $X_{p,f,\overline{T}_1}^{\,w}$ and $X_{p,f+1,\overline{T}_2}^{\,w}$, with $T_1,T_2 \le \overline{T}$, that does not generate any intersection, does not include the stagnation point in Ω_{f,T_1,T_2} and gives the highest local area:

$$X^w_{p,f,T_{stop}} \equiv X^w_{p,f,T_1,},$$

$$X^w_{p,f+1,T_{start}} \equiv X^w_{p,f+1,T_2} : \begin{cases} T_1, T_2 \le \overline{T} \\ I^w_{f,T_1,T_2} = 0; I^w_{f+1,T_1,T_2} = 0 \\ \Omega^w_{f,T_1,T_2} = \max\limits_{T_1,T_2} \left(\Omega^w_{f,T_1,T_2} \right) \\ S_t \notin \Omega^w_{f,T_1,T_2}. \end{cases} \tag{A5}$$

[69] In case of a couple of inner boundary flow lines (which do not belong to the same partition), the APA scheme interpolates them using an algorithm similar to the one described for the outer boundary flow lines (Figure A1d). The stagnation point of the included well is left outside the capture area of the including well and intersections with the flow lines of the included well are avoided.

[70] **Acknowledgments.** The authors wish to thank Henk Haitjema and Roseanna Neupauer for their useful suggestions and comments for the

improvement of the work. We also acknowledge Alberto Tiraferri for final revision of the article.

References

Bakker, M., and D. L. Strack (1996), Capture zone delineation in two-dimensional groundwater flow models, *Water Resour. Res.*, *32*(5), 1309–1315.

Bartels, R. H., J. C. Beatty, and B. A. Barsky (1987), An Introduction to Splines for use in Computer Graphics and Geometric Modeling, p. 476, Morgan Kaufmann Publishers, Inc., Los Altos, California.

Bear, J., and M. Jacobs (1965), On the movement of water bodies injected into aquifers, *J. Hydrol.*, *3*(1), 37–57.

Blandford, T. N., and P. S. Huyakorn (1991), *WHPA: A Modular Semi-Analytical Model for the Delineation of Wellhead Protection Areas*, U.S. Environ. Prot. Agency, Off. of Ground-Water Prot., Washington, D.C., March.

Ceric, A., and H. M. Haitjema (2005), On using simple time-of-travel capture zone delineation methods, *Ground Water*, *43*(3), 408–412.

Feyen, L., K. J. Beven, F. de Smedt, and J. Freer (2001), Stochastic capture zone delineation within the generalized likelihood uncertainty estimation methodology: Conditioning on head observations, *Water Resour. Res.*, *37*(3), 625–638.

Fienen, M. N., J. Jian Luo, and P. K. Kitanidis (2005), Semi-analytical homogeneous anisotropic capture zone delineation, *J. Hydrol.*, *312*, 39–50.

Frind, E. O., D. S. Muhammad, and J. W. Molson (2002), Delineation of three-dimensional well capture zones for complex multi-aquifer system, *Ground Water*, *40*(6), 586–598.

Frind, E. O., J. W. Molson, and D. L. Rudolph (2006), Well vulnerability: A quantitative approach for source water protection, *Ground Water*, *44*(5), 732–742.

Guadagnini, A., and S. Franzetti (1999), Time-related capture zones for contaminants in randomly heterogeneous formations, *Ground Water*, *37*(2), 253–260.

Harbaugh, A. W., E. R. Banta, M. C. Hill, and M. G. McDonald (2000), MODFLOW-2000, the US Geological Survey modular Groundwater model-User guide to modularization concepts and the groundwater process, *Open-File Report 00-92*, U.S. Geol. Surv., Reston, Va.

Javandel, I., and C. F. Tsang (1986), Capture-zones type curves: A tool for aquifer cleanup, *Ground Water*, *24*(5), 616–625.

Kinzelbach, W., M. Marburger, and M. Chiang (1992), Determination of groundwater catchment areas in two and three spatial dimensions, *J. Hydrol.*, *134*, 221–246.

Kunstmann, H., and W. Kinzelbach (2000), Computation of stochastic wellhead protection zones by combining the first-order second-moment method and Kolmogorow backward equation analysis, *J. Hydrol.*, *237*, 127–146.

Nelder, J. A., and R. Mead (1965), A simplex method for function minimization, *Comp. J.*, *7*, 308–313.

Neupauer, R. M., and J. L. Wilson (1999), Adjoint method for obtaining backward-in-time location and travel time probabilities of a conservative groundwater contaminant, *Water Resour. Res.*, *35*(11), 3389–3398.

Pollock, D. (1989), Documentation of computer programs to compute and display pathlines using results from the U. S. Geological Survey modular three-dimensional finite-difference ground-water model, *Open-File Report 89-381*, U.S. Geol. Surv., Reston, Va.

Pollock, D. (1994), User's Guide for MODPATH/MODPATH-PLOT, Version 3: A particle tracking post-processing package for MODFLOW, the U. S. Geological Survey finite-difference ground-water flow model, *Open-File Report 94-464*, U.S. Geol. Surv., Reston, Va.

Schafer-Perini, A., and J. L. Wilson (1991), Efficient and accurate front tracking for two-dimensional groundwater flow models, *Water Resour. Res.*, *27*(7), 1471–1485.

Stauffer, F., S. Attinger, S. Zimmermann, and W. Kinzelbach (2002), Uncertainty estimation of well catchments in heterogeneous aquifers, *Water Resour. Res.*, *38*(11), 1238, doi:10.1029/2001WR000819.

Strack, O. D. L. (1989), *Groundwater Mechanics*, p. 732, Prentice-Hall, Englewood Cliffs, N. J.

Tosco, T. A. E., R. Sethi, and Di A. Molfetta (2006), A probabilistic method for delineation of wellhead protection areas, paper presented at Fifteenth International Symposium on Mine Planning & Equipment Selection, MPES, Turin, Italy, 20–22 Sept.

Tosco, T. A. E., R. Sethi, and A. Di Molfetta (2007), A backward probabilistic model to calculate well head protection areas, paper presented in International Conference on WAter POllution in natural POrous media at different scales. Assessment of fate, impact and indicators, WAPO^2, Barcelona, Spain, 11–13 April.

Van Leeuwen, M., A. P. Butler, C. B. M. te Stroet, and J. A. Thompkins (2000), Stochastic determination of well capture zones conditioned on regular grids of transmissivity measurements, *Water Resour. Res.*, *36*(4), 949–957.

Varljen, M. D., and J. M. Shafer (1991), Assessment of uncertainty in time related capture zones using conditional simulation of hydraulic conductivity, *Ground Water*, *29*(5), 737–748.

Vassolo, S., W. Kinzelbach, and W. Shafer (1998), Determination of a well head protection zone by stochastic inverse modelling, *J. Hydrol.*, *206*, 268–280.

Zheng, C., and G. D. Bennett (2002), *Applied Contaminant Transport Modelling*, John Wiley, New York.

A. Di Molfetta, R. Sethi, and T. Tosco, DITAG-Dipartimento di Ingegneria del Territorio, dell'Ambiente e delle Geotecnologie, Politecnico di Torino, corso Dulca degli Abruzzi 24, 10129 Torino, Italy. (antonio.dimolfetta@polito.it; rajandrea.sethi@polito.it; tiziana.tosco@polito.it)