GEOPHYSICS, VOL. 56, NO. 12 (DECEMBER 1991); P. 2057–2069, 14 FIGS.

Ray tracing in 3-D complex isotropic media: An analysis of the problem

Jean Virieux* and Véronique Farra*

ABSTRACT

Procedures for accurate ray tracing in complex three-dimensional media with interfaces are proposed. The ray tracing equations and the associated paraxial linear equations are solved either by a numerical solver or by an analytical perturbation approach. Interfaces are described with an explicit representation or an implicit representation using B-spline interpolation. For the implicit representation, we exploit two important properties of B-splines, the convex hull and subdivision properties, in order to determine the intersection of the ray with the interface.

At the free surface where the recording system is located, a sampling strategy is proposed: limits of branches at caustics, shadow zones, and medium boundaries are detected for a fixed azimuth while the take-off angle is automatically adjusted in order to have a roughly homogeneous spacing between end points of the rays. The same strategy is also possible for a fixed take-off angle. The assumed continuity of the traveltime surface between two adjacent azimuths enables one to obtain the initial condition of a ray arriving at any station located on the portion of surface delimited by these two azimuths. This procedure allows for the classification of rays arriving at a given station as we show on different synthetic examples.

INTRODUCTION

Tracing rays in a three-dimensional medium is a formidable problem which has many applications in seismology. Introducing complicated interfaces considerably increases the difficulty. In order to reduce the problem to a tractable one, different simplifications are usually made in the parameterization of either the velocity or the interfaces. Chiu et al. (1986) or Haas et al. (1987), among others, assumed constant velocities between interfaces. These interfaces are simply described by explicit sinusoidal or polynomial functions of the horizontal coordinates and do not allow for complex shapes. These different simplifications result in fast ray-tracing methods, but their impact on the tomographic image, for example, are difficult to analyze. Several research groups (Cerveny and Psencik, 1983; Cerveny, 1987; Pereyra et al., 1980; Pereyra, 1988; Gjoystdal et al., 1984; Hanyga, 1988) have attempted to go one-step further and solve with good accuracy the ray-tracing problem in three-dimensional media with interfaces.

In this article, we analyze the classes of difficulties met in ray tracing and we investigate methods to solve them in an efficient way. The first step is to solve the ray-tracing equations. Analytical perturbation techniques proposed by Farra (1990) or Virieux (1991) are emphasized as an efficient and accurate alternative to numerical solvers. We must also check intersections with interfaces and compute them when they occur. The intersection problem is strongly related to the geometrical description of interfaces and the assumed complex shape of interfaces leads us to introduce implicit representations. Finally, we investigate a first attempt to find rays arriving at a station for different branches of the traveltime surface.

Before addressing these problems, let us summarize briefly the ray theory and the associated first-order linear technique called the paraxial approximation.

RAY-TRACING THEORY AND THE PARAXIAL APPROXIMATION

Tracing rays inside a medium is a powerful tool for extracting information, because the computed quantities (traveltime, slowness vector, polarization vectors, and amplitudes) are related to simple quantities in a seismogram and are perfectly associated with different features of the medium. In this section, we only introduce the notations we use in this paper.

In a three-dimensional medium, the rays can be found by solving the eikonal equation, $(\nabla T)^2 = u^2 = c^{-2}$, where c is the wave speed and u is the corresponding slowness. The

Manuscript received by the Editor September 11, 1990; revised manuscript received June 5, 1991.
*Laboratoire de Sismologie, Institut de Physique du Globe de Paris, 4 Place Jussieu, F.75252 PARIS cedex 05.
© 1991 Society of Exploration Geophysicists. All rights reserved.

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

vector $\mathbf{p} = \nabla T$ orthogonal to the wavefront represents the slowness vector. The traveltime $T(\mathbf{x})$ is deduced afterwards by simple integration along the ray.

The eikonal equation can be recast into a Hamiltonian formalism. The conjugate quantity (corresponding to momentum) of the position is the slowness vector $\mathbf{p}$. Introducing the Hamiltonian proposed by Burridge (1976),

$$H(\mathbf{x}, \mathbf{p}) = \frac{1}{2}[\mathbf{p}^2 - u^2(\mathbf{x})], \tag{1}$$

we observe that the eikonal equation implies that $H = 0$ along a ray. Solving the relation (1) can be performed by the method of characteristics. The canonical equations,

$$\begin{aligned} \dot{\mathbf{x}} &= \nabla_{\mathbf{p}} H \\ \dot{\mathbf{p}} &= -\nabla_{\mathbf{x}} H, \end{aligned} \tag{2}$$

define the evolution of the ray. In relation (2), the dot denotes derivative with respect to the sampling parameter τ defined by

$$dT = \mathbf{p} \cdot d\mathbf{x} = u^2(\mathbf{x})\, d\tau \tag{3}$$

Moreover, $\nabla_{\mathbf{x}}$ and $\nabla_{\mathbf{p}}$ denote the gradients with respect to vector $\mathbf{x}$ and $\mathbf{p}$, respectively. Let us mention that equations (2) are not independent, as discussed by Virieux et al. (1988).

Using first-order perturbation techniques (Luneberg, 1944; Farra and Madariaga, 1987), one can investigate what occurs around a reference drawn ray. The deduced trajectories are often called paraxial rays. Let us introduce the canonical vector $\mathbf{y}_0(\tau)^t = (\mathbf{x}_0(\tau), \mathbf{p}_0(\tau))$ of the reference ray, where the superscript t denotes the transposition operation. A paraxial ray is defined by $\mathbf{x}(\tau) = \mathbf{x}_0(\tau) + \delta\mathbf{x}(\tau)$ and $\mathbf{p}(\tau) = \mathbf{p}_0(\tau) + \delta\mathbf{p}(\tau)$. The perturbation of the canonical vector $\mathbf{y}$, given by $\delta\mathbf{y}^t = (\delta\mathbf{x}, \delta\mathbf{p})$, satisfies the paraxial ray tracing equations,

$$\delta\dot{\mathbf{y}} = \underset{\sim}{\mathbf{A}}\delta\mathbf{y}, \tag{4}$$

deduced by linearization of (2) with the matrix

$$\underset{\sim}{\mathbf{A}} = \begin{bmatrix} \nabla_{\mathbf{x}}\nabla_{\mathbf{p}} H & \nabla_{\mathbf{p}}\nabla_{\mathbf{p}} H \\ -\nabla_{\mathbf{x}}\nabla_{\mathbf{x}} H & -\nabla_{\mathbf{p}}\nabla_{\mathbf{x}} H \end{bmatrix}. \tag{5}$$

Solutions of linear system (4) can be found by standard propagator techniques (Aki and Richards, 1980, p 273). From a given value $\delta\mathbf{y}(\tau_0)$, the paraxial canonical vector

$$\delta\mathbf{y}(\tau) = \mathcal{P}(\tau, \tau_0)\delta\mathbf{y}(\tau_0) \tag{6}$$

is expressed with the help of the propagator $\mathcal{P}(\tau, \tau_0)$. The obtained trajectory is a ray if the canonical vector $\delta\mathbf{y}$ satisfies the extra condition,

$$\delta H = \nabla_{\mathbf{p}} H \cdot \delta\mathbf{p} + \nabla_{\mathbf{x}} H \cdot \delta\mathbf{x} = 0, \tag{7}$$

derived from first order perturbation of the Hamiltonian. Inversely, relation (7) implies that the Hamiltonian is zero to first order along paraxial rays. Moreover, using relations (2), one can verify that δH is constant along any solution of the system (4). Therefore, it is sufficient to guarantee $\delta H = 0$ at a given position on the paraxial ray. Practically, the relation (7) will be set either at the beginning or at the end of the ray.

Paraxial rays are very important in many applications of ray theory, because they provide stable information around a given ray [see Cerveny (1985) for description of many applications]. We use them for detecting any contact of the ray with a caustic and for two-point ray tracing.

Looking for caustics is very important because the shape of the propagating signal is modified when the ray touches a caustic. For a two-dimensional medium, rays belong to either a forward branch or a reverse branch. These branches are separated by caustics and shadow zones: partition in branches is easy to build. In three-dimensional media, we have two initial angles, the azimuthal and take-off angles, and the pattern of caustics and shadow zones at the free surface is much more complicated than in a two-dimensional medium. In any case, determination of branches requires a detection of caustics with the computation of index KMAH (Ziolkowski and Deschamps, 1980).

We trace the three paraxial independent trajectories corresponding to the following point source initial conditions $\delta\mathbf{y}_1(0) = (0, 0, 0, 1, 0, 0)$, $\delta\mathbf{y}_2(0) = (0, 0, 0, 0, 1, 0)$ and $\delta\mathbf{y}_3(0) = (0, 0, 0, 0, 0, 1)$. These trajectories are not necessarily paraxial rays, because they do not generally satisfy the equation (7). They must be combined in order to generate paraxial rays. Let us denote $\delta\mathbf{q}_1$, $\delta\mathbf{q}_2$, and $\delta\mathbf{q}_3$ the perturbations of position of the three corresponding paraxial trajectories with respect to the reference ray. Let us consider a ray with initial slowness vector $\mathbf{p}_i$ and two independent paraxial rays with initial point source conditions $\delta\mathbf{y}(0) = (0, 0, 0, 0, p_{zi}, -p_{yi})$ and $\delta\mathbf{y}'(0) = (0, 0, 0, p_{zi}, 0, -p_{xi})$, respectively. These paraxial rays are obtained by linear combinations of the three paraxial elementary trajectories $\delta\mathbf{y}_1$, $\delta\mathbf{y}_2$ and $\delta\mathbf{y}_3$. The cross-section of the ray tube at any position $\mathbf{x}$ along the ray is $\mathbf{p}.(\delta\mathbf{x} \times \delta\mathbf{x}')$, where $\delta\mathbf{x}$ and $\delta\mathbf{x}'$ are position perturbations associated to the initial perturbations $\delta\mathbf{y}(0)$ and $\delta\mathbf{y}'(0)$, respectively, and $\mathbf{p}$ is the slowness vector at $\mathbf{x}$. Developing the mixed product, we obtain the equivalent global determinant

$$\begin{vmatrix} p_x & \delta q_{1x} & \delta q_{2x} & \delta q_{3x} \\ p_y & \delta q_{1y} & \delta q_{2y} & \delta q_{3y} \\ p_z & \delta q_{1z} & \delta q_{2z} & \delta q_{3z} \\ 0 & p_{xi} & p_{yi} & p_{zi} \end{vmatrix} \tag{8}$$

for every point along the ray. Selecting other independent initial paraxial rays will end up with a global determinant equivalent to the one defined by equation (8). When this determinant changes its sign, the index KMAH is incremented by 1. The minor determinants with respect to p_{xi}, p_{yi} and p_{zi} are also estimated to detect a possible focal point. In that case, the KMAH index ray is incremented by two.

When the medium has interfaces, the parameters of paraxial rays follow a generalized transformation proposed initially by Deschamps (1972) and reformulated by different workers (see Cerveny et al. (1974) for a differential geometry approach, Farra (1987) for a Hamiltonian approach in general coordinate system and Farra et al. (1989) for a Hamiltonian approach in Cartesian coordinate system). This transformation can be obtained by two equivalent approaches: the matching procedure for traveltimes [see Cerveny (1985)

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

for references] or the perturbation of Snell's law (Chapman, 1985; Farra, 1987; Farra et al., 1989).

Basically, two elementary transformations have to be performed on the paraxial canonical vectors when the central ray hits an interface. A first-order linear transformation denoted by $\underline{\underline{\Pi}}$ gives the canonical vector $\mathbf{dy}$ of any paraxial ray along the interface. This paraxial canonical vector $\mathbf{dy}$ is transformed by another linear transformation $\underline{\underline{T}}$ into the paraxial canonical vector $\mathbf{d\hat{y}}$ of the converted paraxial ray along the interface. The canonical vector $\mathbf{d\hat{y}}$ is used as the initial condition of the converted paraxial ray [see Farra et al. (1989) for further discussions]. The transformation matrices $\underline{\underline{\Pi}}$ and $\underline{\underline{T}}$ involve the structure of the interface (normal vector and curvature matrix $\underline{\underline{C}}$) at the intersection point, as well as the velocity distribution of the incident medium and the medium associated with the converted ray. Computing matrices $\underline{\underline{\Pi}}$ and $\underline{\underline{T}}$ depends on the interface representation, as we shall see later in the section on that subject.

In addition to the estimation of the index KMAH, the paraxial rays are also used in order to obtain convergence at a given station once an initial guess is given. This technique has been discussed by Cerveny (1985). For the selected Hamiltonian of this article, a previous description has been given by Virieux et al. (1988): we outline the procedure in the following. Let us assume that we have traced a ray with initial azimuthal and take-off angles ϕ and θ. This ray hits the free surface at a position $\mathbf{x}_f$ away from the observer position $\mathbf{x}_{\text{obs}}$. We consider the two paraxial rays with initial point source conditions, i.e., $\delta\mathbf{x}(0) = 0$, such that one has an initial slowness perturbation $\delta\mathbf{p}_\phi(0)$ associated to a variation of angle ϕ and the other one has an initial slowness perturbation $\delta\mathbf{p}_\theta(0)$ associated to a variation of angle θ. These paraxial rays are obtained by linear combinations of the three paraxial elementary trajectories $\delta\mathbf{y}_1$, $\delta\mathbf{y}_2$ and $\delta\mathbf{y}_3$ previously introduced. Using these two paraxial vectors as well as the projection matrix $\underline{\underline{\Pi}}$ at the free surface, we may construct the following linear system

$$\begin{bmatrix} x_{\text{obs}} - x_f \\ y_{\text{obs}} - y_f \end{bmatrix} = [Q_\phi \;\; Q_\theta] \begin{bmatrix} \delta\phi \\ \delta\theta \end{bmatrix}, \tag{9}$$

with an obvious definition for Q_ϕ and Q_θ. The deduced variations $\delta\phi$ and $\delta\theta$ are then used to update shooting angles for another iteration to reach for the station: the nonlinear problem of shooting at a given station is linearized around the hitting point and a few iterations (typically between three and ten) are needed in order to converge.

SOLVING RAY TRACING EQUATIONS

Solving ray equations (2) and (4) in heterogeneous media is one of the two crucial points we address in this paper. The second point, the intersection of rays with an interface separating two different media, will be discussed in the following section. We shall concentrate here on three different strategies to integrate the ray equations: a numerical solver, a finite element approach with analytical rays and a finite element approach with perturbed analytical rays.

Numerical ray tracing approach

Solving the ray tracing equations (2) and (4) requires an integration of a system of ordinary differential equations, problem widely discussed in textbooks on numerical analysis (Conte and de Boor, 1983). Simple numerical schemes as second-order or fourth-order numerical integrations are sufficient and, basically, Runge-Kutta and predictor-corrector numerical integrations are used in the seismological literature.

These two solvers with very different philosophies (Stone and Forbes, 1990) require the computation of partial derivatives of the velocity distribution up to the second order. While the predictor-corrector scheme uses one evaluation at each integration step whatever its order of precision may be, the Runge-Kutta algorithm demands two evaluations for the second-order scheme and four for the fourth-order scheme. In spite of that, the Runge-Kutta approach has been widely used because it is self-starting (second-order predictor-corrector algorithm requires a Runge-Kutta integration at the first step) and easy to program. More sophisticated solvers have been discussed by Gjoystdal et al. (1984).

Estimating the integration step used by the numerical solvers is quite difficult. Although self-adapting solvers exist generally based on the comparison of the solutions for two different orders of integration, most of the approaches require the selection of this parameter and lead to a very inefficient integration when rays are nearly straight lines over large distances. One might compute the Hamiltonian (1), which must be equal to zero, and verify a posteriori the adequate choice of the integration step for the numerical scheme. We found a drift from 10^{-8} to 10^{-6} during integration when integration steps give accurate rays. More important drifts are the warning for incorrect integration step. In our trial tests to evaluate this drift, we have checked the accuracy of the ray with the one obtained for half the selected integration step. A discussion about the precision of numerical solvers can be found in Sambridge and Kennett (1990).

Programming efficient estimations of partial derivatives of the velocity can speed up dramatically the ray tracing. We have found that the representation of the square of slowness as a three-dimensional tensorial product of cardinal B-splines of order 4 (de Boor, 1978) gives very simple expressions for partial derivatives (see below for a more detailed description of B-splines which are also used in the description of interfaces).

Analytical ray-tracing approach

Numerical integration of ray trajectories is a rather lengthy procedure, especially in three-dimensional media. Moreover, choosing the sampling parameter is always a difficult problem. The ray tracing scheme can be simplified as proposed by different workers (Müller, 1984; Chapman, 1985; Cerveny, 1987; Virieux et al., 1988). The medium is divided into elementary cells where analytical solutions of ray tracing are available. A constant gradient of the square of slowness inside triangles or tetrahedra leads to the simplest solutions for ray trajectories, paraxial rays and traveltimes. We refer the reader to Cerveny (1987) or Virieux et al. (1988) for a complete description and we shall only quote the final results in this article.

Let us consider an element where the square of slowness,

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

$$u^2(\mathbf{x}) = u_0^2 + \boldsymbol{\gamma} \cdot \mathbf{x}, \tag{10}$$

presents a constant gradient $\boldsymbol{\gamma}$. The ray-tracing equations (2) have a very simple solution (Virieux et al., 1988),

$$\begin{aligned} \mathbf{p}_0(\tau) &= \frac{1}{2}\,\boldsymbol{\gamma}(\tau - \tau_0) + \mathbf{p}_0(\tau_0) \\ \mathbf{x}_0(\tau) &= \frac{1}{4}\,\boldsymbol{\gamma}(\tau - \tau_0)^2 + \mathbf{p}_0(\tau_0)(\tau - \tau_0) + \mathbf{x}_0(\tau_0), \end{aligned} \tag{11}$$

where the position $\mathbf{x}_0(\tau_0)$ and the slowness of $\mathbf{p}_0(\tau_0)$ are given at parameter τ_0. The traveltime is obtained by direct integration of the relation (3). This velocity distribution is very interesting because the propagator matrix $\mathcal{P}_0(\tau, \tau_0)$ associated to the paraxial system (4) turns out to have a very simple analytical expression

$$\begin{bmatrix} \underline{\mathbf{I}} & (\tau - \tau_0)\underline{\mathbf{I}} \\ 0 & \underline{\mathbf{I}} \end{bmatrix}, \tag{12}$$

where $\underline{\mathbf{I}}$ is the identity matrix. The jump of the gradient between cells requires projections of the paraxial canonical vectors at each boundary [see Farra et al. (1989)]. However, because the gradient of the square of slowness suffers from abrupt jumps between cells, an anomalous behavior of paraxial quantities is likely to occur. Perturbation methods can be used in order to obtain ray trajectories in media with more complex parameterizations.

Perturbed analytical ray-tracing approach

Considering quadratic interpolation in each element makes the gradient of the square of slowness continuous in the whole medium. Moreover, increasing the degree of the interpolation will also reduce the number of elements needed to describe a given velocity distribution. Unfortunately, no analytical solution for ray tracing is available for such an interpolation. We have to resort to perturbed analytical solutions as proposed by Farra (1990) or Virieux (1991). We refer to them for details and present here an outline of the proposed strategy.

The perturbation approach allows us to consider simpler elements as parallelepipeds instead of tetrahedra. In each element, the linear part of the development of the square of slowness is used in order to define a reference medium with a constant gradient of the square of slowness. In this simple medium, rays have the simple analytical expressions (11). The higher order polynomial terms in the development of $u^2(\mathbf{x})$ are considered as a perturbation Δu^2 and produce a corresponding perturbation of the Hamiltonian $\Delta H = -1/2\Delta u^2$.

We assume that a ray has already been traced in the reference medium. To first order in Δu^2, it is possible to obtain rays of the perturbed medium that deviate slightly from this reference ray (Farra and Madariaga, 1987). Following Farra et al. (1989), we introduce the perturbed canonical vector $y(\tau) = y_0(\tau) + \Delta y(\tau)$ of these rays. The perturbation $\Delta y(\tau) = (\Delta x, \Delta p)$ contains the perturbation in position and slowness vector at τ; $\mathbf{y}_0(\tau)$ is the canonical vector of the reference ray. Expressions of rays in the perturbed medium are given by (Farra and Madariaga, 1987):

$$\Delta \mathbf{y}(\tau) = \mathcal{P}_0(\tau, \tau_0)\Delta \mathbf{y}(\tau_0) + \int_{\tau_0}^{\tau} \mathcal{P}_0(\tau, \tau')\Delta B(\tau')\, d\tau', \tag{13}$$

where $\mathcal{P}_0$ is the propagator of the paraxial rays in the unperturbed medium, $\Delta \mathbf{y}(\tau_0)$ is the initial perturbation and

$$\Delta B = \begin{pmatrix} 0 \\ \dfrac{1}{2}\,\nabla \Delta u^2 \end{pmatrix}. \tag{14}$$

Using the expression of the unperturbed propagator (12), we can write:

$$\begin{aligned} \Delta \mathbf{x}(\tau) &= \Delta \mathbf{x}(\tau_0) + (\tau - \tau_0)\Delta \mathbf{p}(\tau_0) \\ &\quad + \frac{1}{2}\int_{\tau_0}^{\tau} (\tau - \tau')\nabla \Delta u^2\, d\tau', \\ \Delta \mathbf{p}(\tau) &= \Delta \mathbf{p}(\tau_0) + \frac{1}{2}\int_{\tau_0}^{\tau} \nabla \Delta u^2\, d\tau', \end{aligned} \tag{15}$$

where $\Delta \mathbf{x}(\tau_0)$ is the initial position perturbation and $\Delta \mathbf{p}(\tau_0)$ is the initial slowness perturbation. The perturbation of the traveltime, defined as $\Delta \underline{\mathbf{T}}(\tau) = \underline{\mathbf{T}}(\tau) - \underline{\mathbf{T}}_0(\tau)$, is given by the expression

$$\Delta \underline{\mathbf{T}}(\tau) = \Delta \underline{\mathbf{T}}(\tau_0) + \int_{\tau_0}^{\tau} \boldsymbol{\gamma} \cdot \Delta \mathbf{x}(\tau')d\tau' + \int_{\tau_0}^{\tau} \Delta u^2(\tau')\, d\tau'. \tag{16}$$

The perturbed propagator is the sum of two terms (Farra and Madariaga, 1987)

$$\mathcal{P}_0(\tau, \tau_0) \quad \text{and} \quad \begin{bmatrix} \Delta\mathcal{P}_{11} & \Delta\mathcal{P}_{12} \\ \Delta\mathcal{P}_{21} & \Delta\mathcal{P}_{22} \end{bmatrix}$$

with the unperturbed propagator $\mathcal{P}_0$ given by the relation (12), while the submatrices $\Delta\mathcal{P}_{ij}$ are functions of the second order derivatives of Δu^2 (Farra et al, 1989). All these expressions can be computed analytically because Δu^2 and its partial derivatives can be expanded into polynomials in the τ parameter [see Farra (1990) or Virieux (1991)].

Although the expressions to be evaluated are more complicated, the perturbation technique is still a fast method and eliminates the accuracy problem encountered in the semianalytical approach. Moreover, we simplify the cell decomposition and we get rid of the sampling parameter of the numerical solvers. Implementing it into the three-dimensional ray tracing requires the distinction between cells with a potential interface and cells without any interface. In the second case, the procedure will compute the solution at the boundary of the cell, while the first case requires searches along the ray for the intersection. This feature is added to the box strategy presented below.

REPRESENTATION OF INTERFACES AND INTERSECTIONS

When performing ray tracing, searching for an intersection is a critical task which must be achieved efficiently. Which interface must we look for? A common approach is an ordered description, called signature (Cerveny et al., 1977),

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

of the different media and interfaces that must be crossed by the ray. At a current position on the ray, the next potentially crossed interface is given by the signature. This method is fast but the signature of the ray has to be given explicitly. Another useful approach describes the ray by the type of conversion (reflection or transmission) expected at each intersected interface. This procedure might be more time-consuming because rays can be trapped in an unwanted area. A maximum number of crossed interfaces prevents an infinite loop. Both strategies are useful and depend on the problem considered.

The precise strategy to compute the intersection depends dramatically on the interface representation. We are going to investigate both the mostly used explicit description and the more difficult implicit description which allows greater flexibility to generate complex interfaces.

Explicit parameterization

The simplest description of interfaces for computing intersections with rays is an explicit representation of the interface where one cartesian coordinate is given with respect to the two others. Without loss of generality, we might say that the z coordinate is a function of x and y coordinates. Any two-dimensional function $z = f(x, y)$ is suitable for our purpose, as long as the first and second order partial derivatives, which guarantee local definitions of a tangent plane and a curvature matrix, are available.

We have found cardinal B-splines of de Boor (1978) very fast to compute because the interpolating kernels, also called blending functions, are predefined whatever values are given by the user. In order to assure the continuity of the second order derivatives, we must choose cubic interpolation which is associated to the order four of the B-spline. This representation has two weaknesses. One must specify a regular grid in x and y coordinates and the surface does not go through the given points, which are not data points but control points that attract the surface to them with a specific tension (de Boor, 1978). Computing the control points from the data points can be achieved by an inversion procedure, as suggested by Inoue (1986) for example. We assume that the control points are already specified for the ray tracing program.

Other interpolations based on more complex distributions of points are sometimes desirable. For example, an abrupt depth variation of the interface will require a very fine sampling which will over-sample any flat area of the interface. Other spline functions can be used (de Boor, 1978), where one must compute initially the interpolation kernels associated with the more complex spacing between points and with the values at these points [see Farra and Madariaga (1988) for 2-D applications]. We have not investigated further these more complex parameterizations of interfaces in 3-D geometry, because over-sampling still exists from tensorial products.

Whatever the explicit representation is, it is not difficult to compute the intersection between an interface and a ray described by its Cartesian position. Let us assume that the ray is above the interface (we are still considering the z coordinate of the interface as a function of x and y coordinates). As the ray advances, it is easy to check the relative vertical position of the ray with respect to the interface: from the x and y coordinates of the ray, we estimate the z position of the interface and compare it to the z coordinate of the ray. If the interface is still deeper, we continue to integrate the ray trajectory. As soon as the z coordinate of the ray is larger than the z position of the interface, we have to compute the intersection of the ray and the interface knowing two points one above and one below the interface: any numerical root finder will perform well, but we prefer fast Newton methods. Of course, the required position during this search is obtained by integrating ray equations with smaller and smaller integration steps. Checking where the ray is with respect to the interface is fast in case of explicit parameterization. Taking into account that we have to do it very often, one will try to preserve the explicit representation even for complex surfaces. Of course, everything we have said for the z coordinate also applies to the two other coordinates or any direction for which the explicit representation is valid [as proposed by Pereyra (1988)].

Unfortunately, different geometries required by geophysical structures, such as salt domes or faults, are not easily described by an explicit parameterization. One must consider, as Pereyra (1988) did, different patches where a local explicit representation is available. Relations between patches and between patches and rays are difficult to define: Pereyra (1988) only guarantees first-order continuity between patches which induces instability in paraxial ray tracing. Another alternative is the implicit representation which is known in interactive computer graphics (Foley and Van Dam, 1982) to be far more flexible to describe complex surfaces than explicit representation.

Implicit parameterization

For computer synthetic images, the implicit representation has met with great success for the last twenty years with the pioneer work of Bézier (1972). It provides a description of any surface that is so simple and fast it is surprising that it is not intensively used in ray-tracing programs. The reason is the difficulty in solving the intersection problem between rays and surfaces in a rather efficient way for 3-D models. The problem is much simpler for a two-dimensional medium, as we shall show.

Because the terminology is not widespread in the geophysical community, we shall define a few notations in order to avoid any misinterpretation. The often made distinction between Bézier and B-spline interpolations is rather artificial, because both of them come from Bernstein interpolating polynomials (Foley and Van Dam, 1982). For simplicity, we will also discuss this distinction, although the reader might have heard about a more general definition of Bézier interpolations which includes the two definitions in this paper. An implicit parameterization of an interface,

$$x = f_x(u, v),\ y = f_y(u, v) \quad \text{and} \quad z = f_z(u, v), \tag{17}$$

introduces two new parameters u and v. We shall consider cubic interpolation in u and v in order to compute second-order derivatives and we select the associated B-spline of order 4. For example, the x coordinate is given by the relation

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

$$x = \sum_{l=1}^{4}\sum_{k=1}^{4} u^{l-1}v^{k-1} \sum_{i=1}^{4}\sum_{j=1}^{4} P_x(i,j)M(i,l)M(j,k), \tag{18}$$

where $P_x(i, j)$ is the x coordinate of the control point $\mathbf{P}(i, j)$ and the parameters u, v belong to the square $[0, 1] \times [0, 1]$. The whole interface is described by changing the sixteen control points. The matrix

$$\underset{\sim}{\mathbf{M}} = 1/6\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \tag{19}$$

is the matrix, denoted also by $\underset{\sim}{\mathbf{M}}_s$, of weighting factors specific to the B-spline interpolation. An equivalent matrix is available for Bézier interpolation which is denoted by $\underset{\sim}{\mathbf{M}}_\mathbf{b}$. The total surface is composed by patches of sixteen control points with second order continuity between patches. However, specifying twice one control point will induce a jump in the curvature, while repeating three times the same point will restrict the definition of the tangent plane. This procedure can be used to define complicated surfaces such as faults, pinch outs, and salt domes.

The B-spline interpolation has many advantages in common with Bézier interpolation. Additional advantages of B-spline interpolation is the C^2 continuity needed in our ray tracing. This argument in itself is sufficient to justify our selection. It is still interesting to give the other slight differences between these two interpolations. An advantage is the local support of the blending functions of B-splines, but the order four of B-spline interpolation used in this paper is too low to exhibit this advantage—when we move one of the sixteen points, the local shape of the surface described by the sixteen control points is entirely modified. Finally, the Bézier interpolation forces the surface to go through the four corner points of the local patch. These points are preferential points in the description of the surface. On the contrary, the B-spline interpolation will give an identical importance to any control points. In any case, one might switch from one representation to the other one by simple matrix multiplications (see appendix).

Two important properties of Bézier and B-spline representations will play a key role in efficient ray tracing. The first property is the convex hull property which says that the surface lies inside the convex volume defined by the control points. Figure 1 shows a B-spline curve inserted in the convex polygon of the four control points. The same property holds in three dimensions: it is only more difficult to plot. The second property is the subdivision or splitting widely used in computer graphic design to draw a curve or a surface with a given accuracy (Lane and Carpenter, 1979; Cohen et al., 1980; Lyche and Morken, 1986). In the appendix, we give the transformations as well as the geometrical interpretation of the subdivision technique. Figure 2 shows how we deduce more and more control points for a given curve. From four initial control points which are outside the plotting area, we deduce five control points given by circles. Repeating the procedure twice, we deduce more and more control points for the same curve. These control

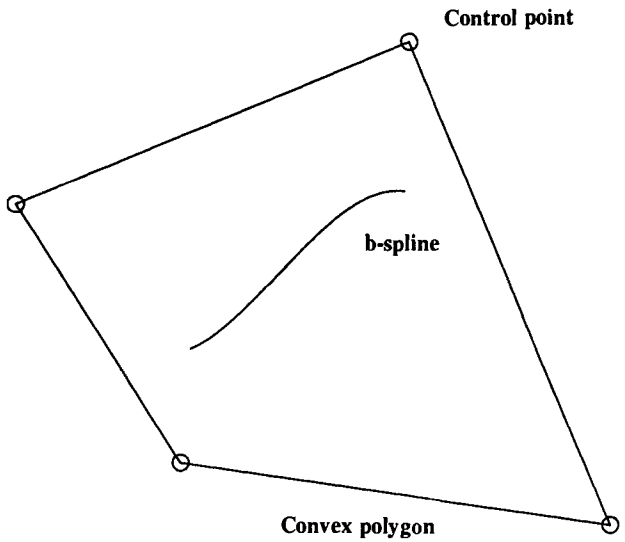

FIG. 1. An illustration of the convex hull property for B-spline interpolation. The four control points are circles delimiting a polygon inside which is the interpolated curve. For surfaces, the property is also true.

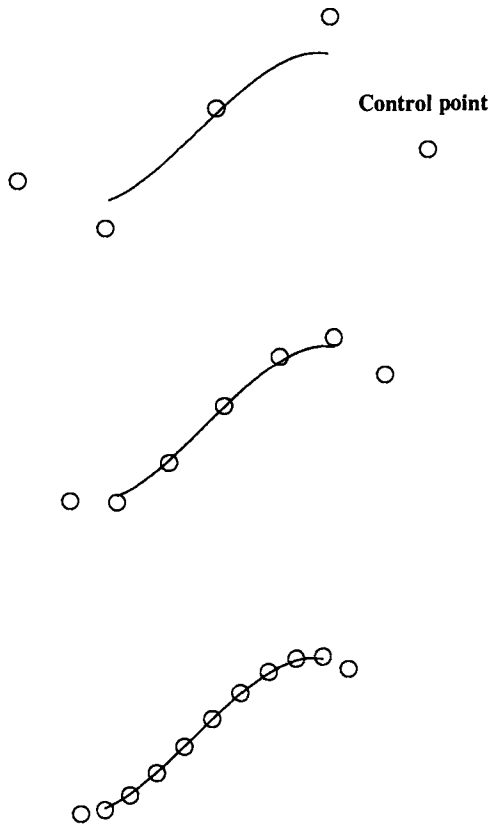

FIG. 2. An example of subdivision technique. From four initial control points outside the figure, five control points are deduced for the same interpolated curve. The next subdivisions will give 7 and 11 control points which are more and more located near the interpolated curve.

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

points are located closer and closer to the curve, which means that the convex surface including the curve is smaller and smaller: a key advantage for solving the intersection with rays. Of course, the procedure is identical for surfaces.

Computing the intersection between a ray and an interface is very complicated, because any evaluation with respect to an interface requires the knowledge of the two implicit parameters (u, v) which are not given directly by the ray tracing. Moreover, a dramatic difference exists between the 2-D and 3-D cases. Let us assume that we want to test the intersection of a straight segment between points A and B of a ray and an interface (Figure 3 or Figure 4). In a two-dimensional medium, we rotate the coordinates in such a way that the segment AB is vertical. Any point of the segment has now the same horizontal coordinate X. Because the representation (17) is independent of the coordinate system, we are able to solve the cubic equation $X = f_x(u)$. From the three possible solutions of the parameter u, we select the solution which corresponds to the nearest point to the starting point A. Unfortunately, this procedure does not work in a three-dimensional medium, because, after the rotation, we have to solve two coupled polynomial equations, $X = f_x(u, v)$ and $Y = f_y(u, v)$: we need to obtain all available solutions in order to select the nearest intersection point to the starting point.

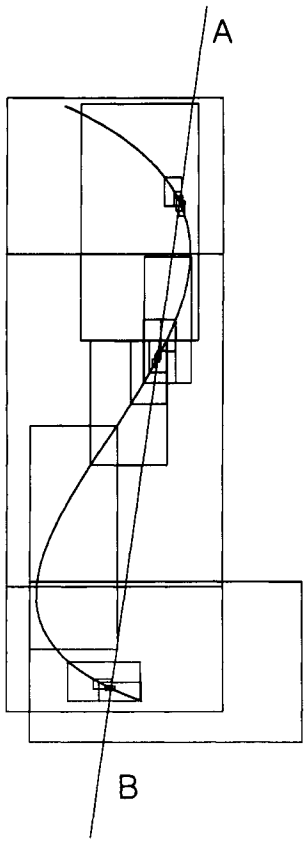

FIG. 3. The subdivision technique is applied to the intersection of a B-spline curve and a straight segment (AB). Rectangles including the polygon of four successive control points are drawn during subdivisions. The three intersections are detected and accurately obtained.

Using the convex hull property and the subdivision technique, we succeed in solving the intersection problem. Let us first illustrate this in the two-dimensional case. We use the rectangle containing the convex surface defined by the control points. It is a simple matter to check whether the segment AB goes through the rectangle delimited by the current four control points. If it does, we perform the subdivision and check the intersection between the ray and the deduced smaller rectangles. We might iterate until the selected rectangle is small enough that we might consider it as a point. Figure 3 shows this cascade of rectangles down to three intersections between the segment AB and the B-spline curve: three points are obtained and we are sure that we have obtained all of them. The procedure is slower than any minimization procedure, but it will find all the solutions. Figure 4 shows the three intersections of the segment AB with a B-spline surface. Only the base rectangle of parallelipeds is shown. The implicit parameters (u, v) are also evaluated during the subdivision procedure at its associated level of precision.

Locally, it is wise to switch to a method which has a faster convergence to the intersection point. Denoting positions of point A by (x_0, y_0, z_0) and point B by (x_1, y_1, z_1), one must solve the problem,

$$\begin{aligned} x_0 + t(x_1 - x_0) &= f_x(u, v) \\ y_0 + t(y_1 - y_0) &= f_y(u, v) \\ z_0 + t(z_1 - z_0) &= f_z(u, v), \end{aligned} \tag{20}$$

with a given initial solution (u, v, t). This procedure has to be used when a unique solution is expected starting from the current solution of the subdivision procedure.

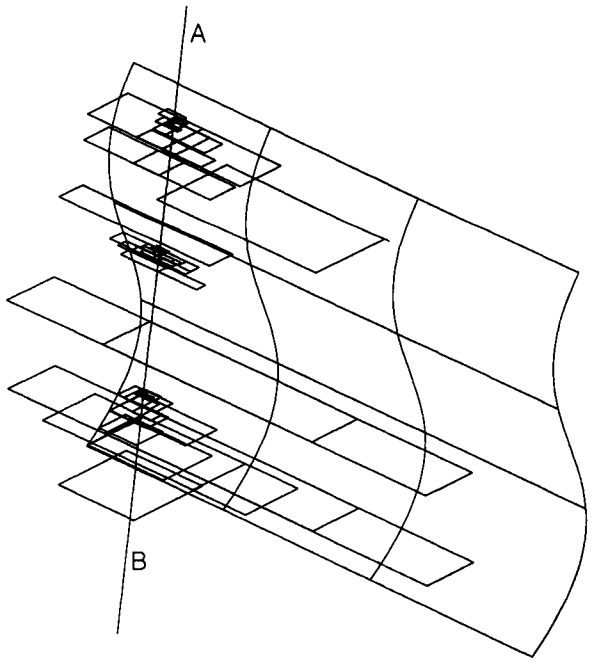

FIG. 4. Same as Figure 3 for a curved surface in a 3-D medium plotted using a perspective view. Only the base rectangle of each paralleliped is drawn during subdivisions. Three intersections are also detected and obtained.

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

Rotations, which are used in the 2-D approach, are no longer required. But we can apply them and reduce the final problem to the finding of the two implicit parameters (u, v). Performing rotations also has the advantage of giving a very simple criterion for possible multivalued solutions, because the ray is vertical. When slopes of the 24 segments connecting the 16 control points have the same sign, we switch to the search of the a priori single local solution.

We have found in our numerical investigation that the following strategy presents a good compromise between stable, but inefficient, subdivision methods and an unstable, but fast, local search for a solution. For the first four levels of subdivision, we perform the subdivision procedure in the general Cartesian coordinate system. During the integration process along the ray, most of the time we found that the ray is too far away from the interface so we stop the subdivision procedure. But, if the intersection between the ray and the interface is still a plausible hypothesis because boxes are crossed by the ray, we perform rotations and we continue the subdivision with a check for single-valued functions. Once we are sure that only one solution is possible in the local subdivision (we assume it at the tenth level of subdivision in any case), we perform the local search for this single solution. This strategy gives a good compromise between a quick rejection when the ray does not really cross the interface and an accurate determination of every intersection.

We have solved the local problem of intersecting a straight line and a surface defined with an explicit or an implicit representation based on B-spline interpolation. All intersections are obtained and we select the nearest one to be the correct impact of the ray on the interface. This very powerful procedure is too slow to be done at every integration step along the ray so we must insert it in a global strategy we shall now discuss.

Strategy based on a box hierarchy

Instead of testing at every integration step the intersection of the ray with the interface, we might test the intersection with more simple surfaces which define a volume containing the interface. By iteration, we could replace the interface by a hierarchy of boxes which includes parts of it. The ideal number of boxes depends on the problem at hand, but we find that three levels cover most of the applications. The interface must be included in a general parallelepipedic box, while the elementary parallelepiped deduced from the sixteen control points is the final stage before the precise search for the intersection. An intermediate series of boxes takes into account the local complexity of the interface, as shown in Figure 5.

Testing the crossing of a straight line (a portion of the ray) with a parallelepiped box is very fast and is performed at every integration step along the ray. Once we have located an elementary paralleliped crossed by the straight line, we must look for the intersection with the B-spline surface, following the mixed strategy previously described.

Paraxial transformation at the interface

Paraxial transformations $\underline{\underline{\Pi}}$ and $\underline{\underline{T}}$ across interfaces require the computation of the normal vector and the curvature matrix $\underline{\underline{C}}$ at the intersection point (see Farra et al., 1989). For the explicit representation which can be written under the standard form $S(x, y, z) = 0$, the corresponding expressions have been given by Cerveny (1985) and are not repeated here. For the implicit representation, we define the normal vector as the cross product of the two independent tangent vectors $(\partial x/\partial u, \partial y/\partial u, \partial z/\partial u)$ and $(\partial x/\partial v, \partial y/\partial v, \partial z/\partial v)$. Moreover, we define an additional parameter w which samples the normal direction. The unitary vector $(\partial x/\partial w, \partial y/\partial w, \partial z/\partial w)$ is parallel to the vector $(\partial S/\partial x, \partial S/\partial y, \partial S/\partial z)$. Moreover, we must solve the linear system,

$$\frac{\partial^2 S}{\partial u^2} = 0, \quad \frac{\partial^2 S}{\partial v^2} = 0, \quad \frac{\partial^2 S}{\partial w^2} = 0,$$
$$\frac{\partial^2 S}{\partial u \partial v} = 0, \quad \frac{\partial^2 S}{\partial u \partial w} = 0, \quad \frac{\partial S}{\partial v \partial w} = 0, \tag{21}$$

for the six unknowns $(\partial^2 S/\partial x^2, \partial^2 S/\partial y^2, \partial^2 S/\partial z^2, \partial^2 S/\partial x \partial y, \partial^2 S/\partial x \partial z, \partial^2 S/\partial y \partial z)$ of the curvature matrix. The first equation of linear system (21) will give explicitly:

$$\left(\frac{\partial x}{\partial u}\ \frac{\partial y}{\partial u}\ \frac{\partial z}{\partial u}\right) \underline{\underline{C}} \begin{pmatrix} \frac{\partial x}{\partial u} \\ \frac{\partial y}{\partial u} \\ \frac{\partial z}{\partial u} \end{pmatrix} = -\frac{\partial S}{\partial x}\frac{\partial^2 x}{\partial u^2} - \frac{\partial S}{\partial y}\frac{\partial^2 y}{\partial u^2} - \frac{\partial S}{\partial z}\frac{\partial^2 z}{\partial u^2} \tag{22}$$

where $\underline{\underline{C}}$ is the curvature matrix containing the second order partial derivatives of the function S. We solve numerically the system (21) at the hitting point in order to apply transformations $\underline{\underline{\Pi}}$ and $\underline{\underline{T}}$ to the paraxial canonical vectors.

SAMPLING STRATEGY AT THE FREE SURFACE

Tracing rays with known initial conditions is much faster and easier than with boundary conditions, although the second problem is the one we must face in practice. In order to find rays connecting the source and the stations, we proceed in three separate steps: shooting, branch investigation, and two point ray tracing. This strategy is performed for a specified type of rays defined by its signature.

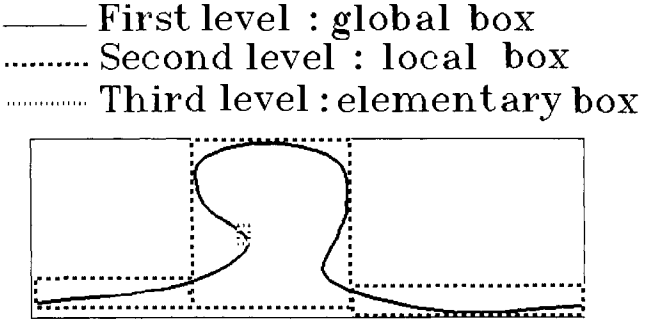

FIG. 5. The three levels of boxes surrounding the interface which seems necessary to speed up the 3-D ray tracing.

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

We must get information about the medium, and the best way to obtain it is to trace rays inside the medium. Different procedures can be investigated. We may require a uniform sampling at the free surface for emerging points of the shooting rays (a very difficult task), or a uniform sampling of initial shooting angles. The first procedure will simplify the two-point ray tracing, while the second approach is a quite natural procedure. We choose an intermediate strategy in which one of the shooting angles is kept constant while we investigate branches by varying the second one. This investigation is easy because it depends on one parameter. Keeping the azimuthal angle or the take-off angle constant will depend on the problem at hand. One technique might succeed better than the other one depending mainly on the geometry of interfaces. For plane interfaces, the selected take-off angle sampling is better adapted than an azimuthal sampling to obtain a good spatial repartition of rays (Figure 6). In the text, we shall mention only the azimuthal sampling, but what we are writing will also apply to the take-off angle sampling.

Along the line traced on the free surface for a given initial azimuthal angle, branch extremities are located at caustics, shadow zones, or boundaries of the medium. Caustics are deduced from the paraxial information, while shadow zones are obtained when the jump in distance between two consecutive rays is too important for a small increase in the take-off angle. Figure 7 presents a perspective view of rays as well as the free surface trace for a given initial azimuthal angle. Although the interface has a simple valley shape, the free surface trace presents already a complicated pattern. The paraxial information allows an estimate of the next angular step in order to guarantee a roughly homogeneous spacing between emerging points of rays at the free surface.

Once we have finished this a priori sampling of the medium, broken lines, connecting the end points of the rays that leave the source in the same azimuthal plane, are drawn at the free surface. We assume the sampling to be dense enough to have coherence between two nearby lines with branch ends located nearly in the same area (Figure 8). If a station lies inside the quadrilateral defined by four points of two neighboring azimuthal sections of a given branch, we deduce that a ray of this branch arrives at the station. At the end of this second step, we have collected potential rays of different branches arriving at the different stations.

In a third and final step, we perform shooting and obtain the different rays arriving at the stations. As already mentioned, the two-point ray tracing is performed using paraxial rays. Three methods are usually considered for this difficult problem: the shooting method, the bending method (Julian and Gubbins, 1977; Pereyra et al., 1980) and the continuation method (Keller and Perozzi, 1983). As stressed by Virieux et al. (1988), the paraxial procedure might be considered equivalently from these three points of view. The final result is a paraxial ray connecting the source and the station. This paraxial ray can be considered as a "true" ray with an accuracy related to its vicinity with the reference ray.

We must underline that the whole procedure depends on the a priori azimuthal sampling we have chosen: too small a sampling will be very time-consuming while rough sampling can result in branch incoherence. For very complex struc-

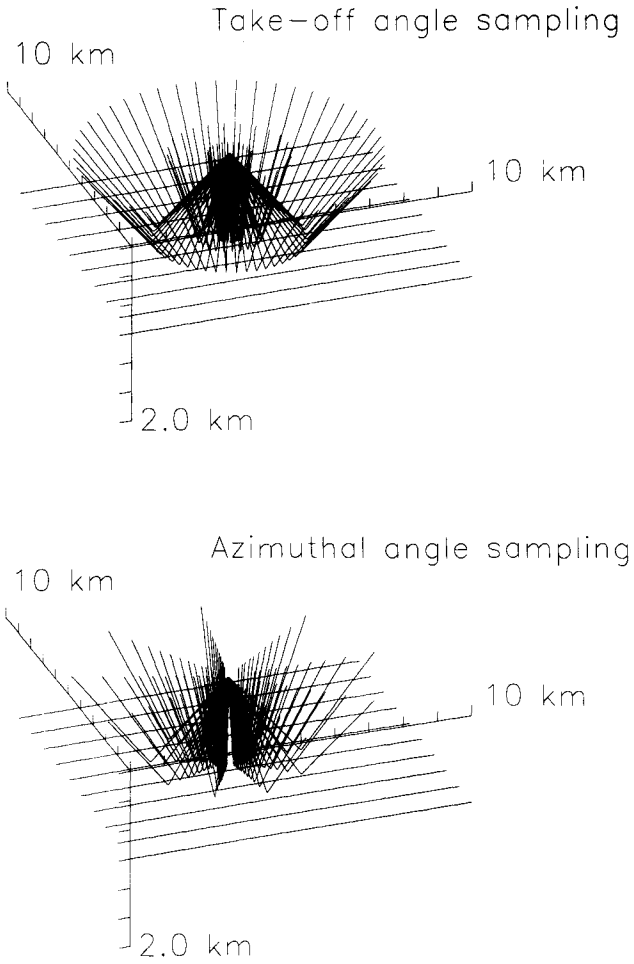

FIG. 6. Examples of the two options for sampling the 3-D medium: the take-off angle sampling on the top panel and the azimuthal sampling on the bottom panel. A horizontal interface is used for illustration.

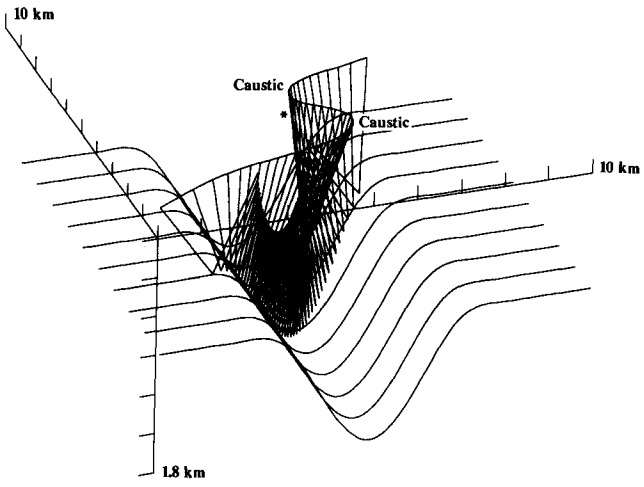

FIG. 7. Trace of the hitting points of rays at the free surface for a given azimuth when an interface has a valley shape. Note the already complex pattern of this trace with the presence of two caustics. For points of the rays in the vicinity of the source (shown by a star), the plotting has been inhibited in order to see the trace continuously.

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

tures, interaction with a hierarchical nested control screen as proposed by Pereyra (1988) would be very helpful for specific targets inside the medium.

If one station is not illuminated by a given branch while a neighboring station is, we try to reach the station by perturbing the initial conditions of the ray illuminating the reached station. The procedure often fails because our initial sampling of the medium by rays has been precise enough to guarantee that the boundary of the current branch goes between the two stations. When it succeeds, it will recover from minor local under-sampling of branches.

Because locating branches is rather time-consuming, we prefer to use any available initial angles. For example, we can use angles computed in a previous step of an iterative inversion. When moving from one source to an adjacent one, we can exploit the knowledge already obtained and use the initial angles obtained at the first source as starting values for shooting from the new one. Of course, we must locate branches from time to time to check the appearance of new rays arriving at the stations.

SYNTHETIC EXAMPLES

Two simple synthetic examples will illustrate the different techniques we have implemented for tracing rays in a 3-D medium with complex interfaces. We shall restrict ourself to one interface for visualization. More interfaces will not change the philosophy of the ray tracing.

The valley example

We consider a valley which has a translation invariance in the y direction. We assume a constant velocity in the upper medium, although we trace inside it using a Runge-Kutta solver. A first investigation will give the Figure 9, where one can see the triplication created by the valley. Using branch coherence at the free surface, we shoot at the different stations and we obtain expected multiple rays when the station is in the triplication area (Figure 10). For this example, the take-off angle sampling gives better results than the azimuthal angle sampling where some of the stations inside the triplication are not connected to the source by three rays as they should be. Local iterations from neighboring stations will recover in any case from these sampling errors.

The salt dome example

A salt dome interface can be represented only with implicit interpolations (Figure 11). We still assume constant velocity in the upper medium. Figure 11 shows the rays leaving the source in a same azimuthal plane. When the source illuminates one wall of the salt dome, one can see the important spatial variation of the rays in the small pencil corresponding to the vertical wall. The coherence of branches is difficult to obtain in this narrow pencil and seems to require locally dense sampling strategies. Figure 12 gives an example of rays illuminating the overhanging part of the salt dome and arriving at a short range network of stations.

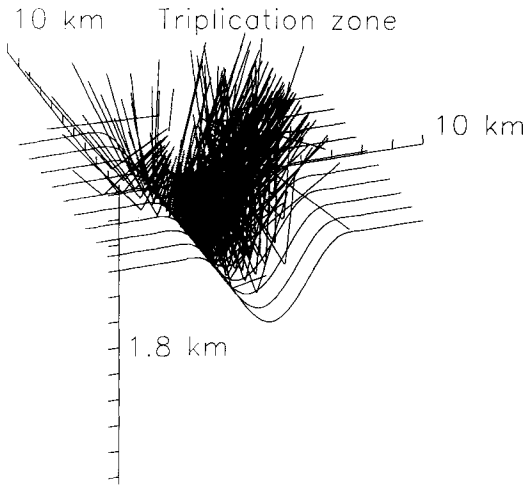

FIG. 9. The initial sampling of the medium for an interface with a valley shape. The take-off angle option has been used and the triplication is apparent from this systematic shooting.

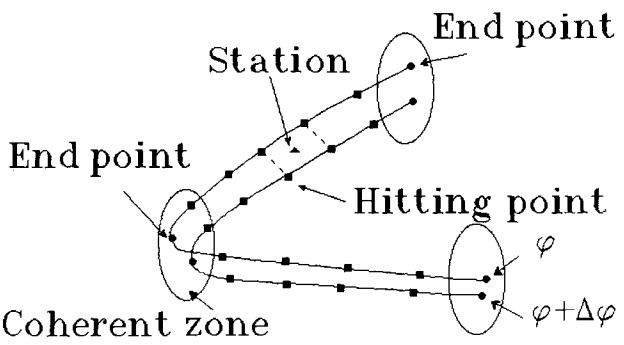

FIG. 8. Schematic strategy for describing rays arriving at the free surface: branch ends are detected by the variation of index KMAH and a coherence of branch ends (drawn as ellipses) is assumed for two nearby azimuthal samplings. If the station lies inside the quadrilateral defined by the end points of four nearby rays, a ray for this branch must arrive to the station.

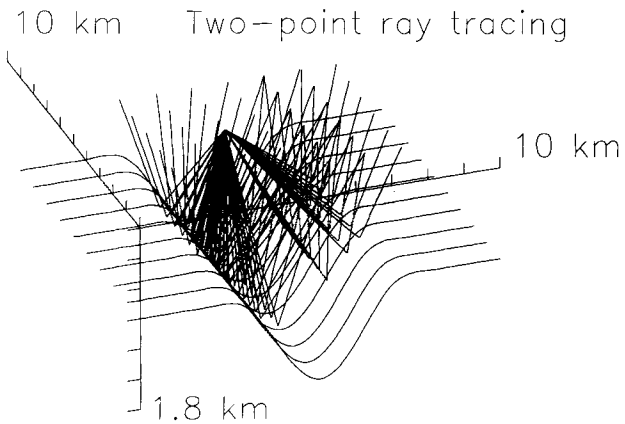

FIG. 10. Shooting at a network of stations. The initial rays are given in Figure 9. A line of stations is located inside the triplication and three rays arrive at these stations. Note the different pencils leaving the source.

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

This example illustrates the flexibility of implicit representations of interfaces.

CONCLUSIONS

We have analyzed the different difficulties encountered when tracing rays in complex three-dimensional media with interfaces. For rays and paraxial rays, we have implemented a second-order Runge-Kutta solver based on a cardinal B-spline interpolation of order four of the square of slowness. However, recent perturbation analytical strategies will be an efficient and accurate alternative in the future. For the same grid, the ratio of CPU times between numerical and perturbed analytical strategies is between two and ten, depending on the order of the polynomial expansion of the perturbation term. Interfaces are defined with an explicit or an implicit representation based on the B-spline interpolation of order four. We use techniques of computer graphics to overcome the problem of intersecting the ray with an implicit interface. Without using a box strategy for the interfaces, the intersection computation is ten times faster for an explicit representation compared to an implicit representation of the same surface. The box strategy will reduce this ratio by a factor which depends on the surface complexity and the number of rays to trace. Finally, we suggest a strategy for finding the different rays arriving at a station. A preliminary sampling of the medium is necessary for this and a careful selection of parameters controlling this sampling is still required for the success of this search. The different rays arriving at a station can be used to synthesize seismograms or to make a traveltime tomographic image.

ACKNOWLEDGMENTS

The authors would like to thank Pr Mechler who has underlined the importance of subdivision techniques for Bézier curves. Many thanks to J. Brac, F. Delprat and V. Vieugué for interesting discussions. During the visit of J.V.

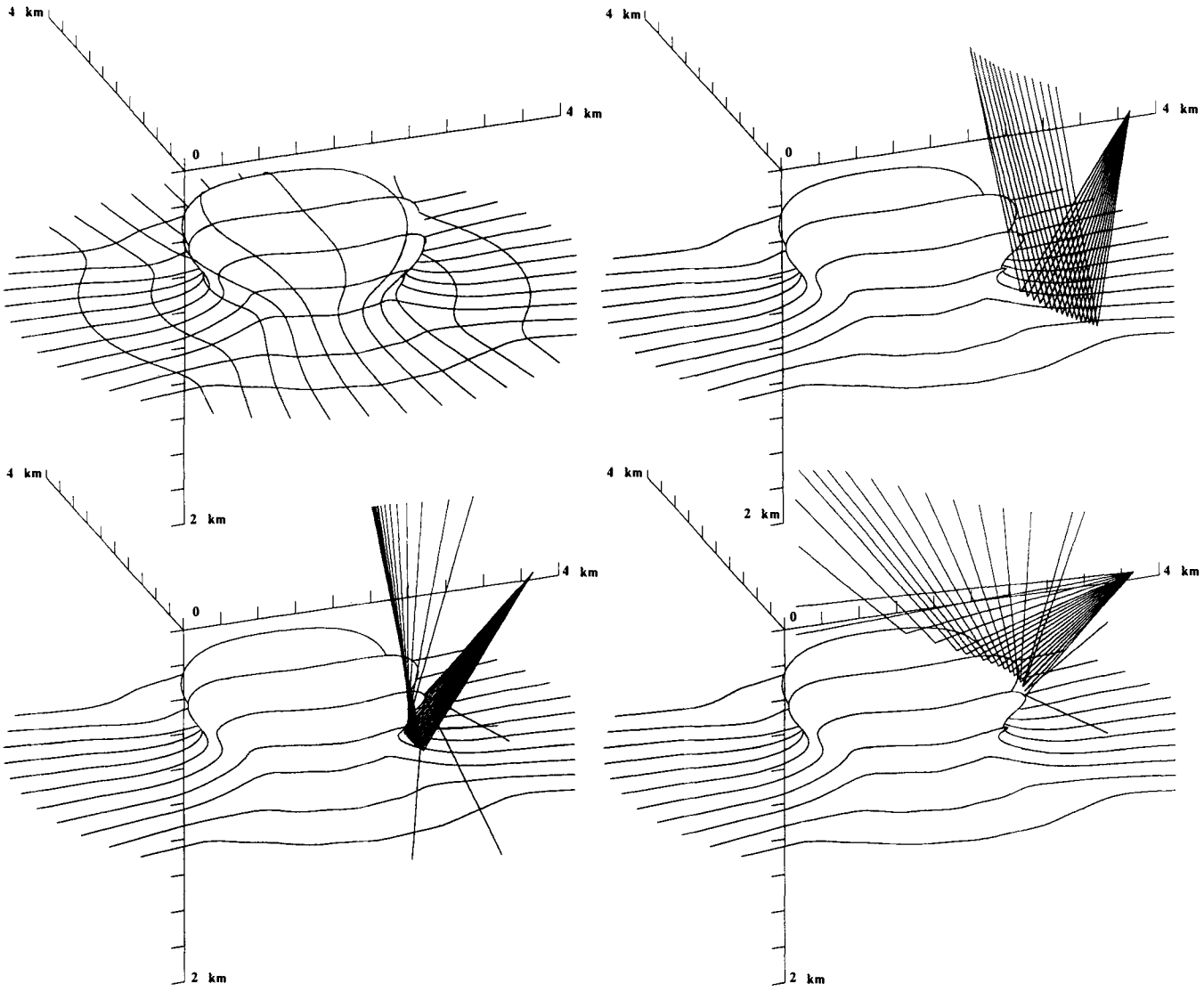

FIG. 11. Salt dome interface using implicit representation. For a given azimuth, three pencils are drawn corresponding to the three different patterns we expect from this geometry: two relatively simply flat areas and a complex situation associated with the overhanging wall.

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

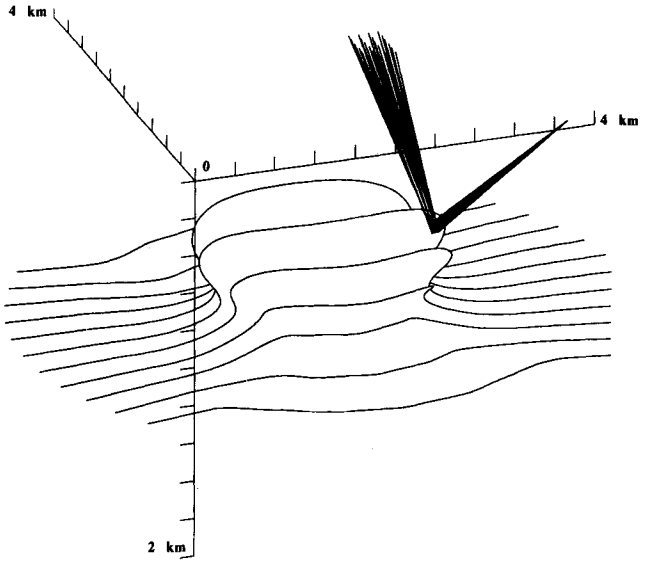

FIG. 12. An example of rays arriving at a narrow network of stations. The overhanging part of the dome is illuminated in this particular case.

at Seismological Observatory, University of Bergen and Norsk Hydro A/S, attention has been drawn to the Oslo algorithm by A. Hanyga, H. B. Helle and J. Pajchel. Many thanks to them. This research was carried out as part of the Pre-stack Structural Interpretation consortium project (PSI). The authors acknowledge the support provided by the sponsors of this project. Contribution IPG N°1164.

REFERENCES

Aki, K., and Richards, P., 1980, Quantitative seismology: Theory and methods: W. H. Freeman & Co.

Bézier, P., 1972, Numerical control—Mathematics and applications: John Wiley & Sons.

Burridge, R., 1976, Some mathematical topics in seismology: Courant Institut of Mathematical Sciences, New York University.

Cerveny, V., 1985, The application of ray tracing to the numerical modelling of seismic wave fields in complex structures, *in* Handbook of geophysical exploration, Section 1, Seismic exploration, Vol 15A: Geophysical Press, 1–119.

——— 1987, Ray tracing algorithms in three dimensional laterally varying layered structures, *in* Nolet, G., Ed., Tomography in seismology and exploration seismics: D. Reidel Publ. Co., 99–133.

Cerveny, V., Langer, J., and Psencik, I., 1974, Computation of geometric spreading of seismic body waves in laterally inhomogeneous media with curved interfaces: Geophys. J. Roy. Astr. Soc., **38**, 9–19.

Cerveny, V., Molotkov, I. A., and Psencik, I., 1977, Ray method in seismology: Univ. of Karlova.

Cerveny, V., and Psencik, I., 1983, Gaussian beams and paraxial ray approximation in three-dimensional elastic inhomogeneous media, J. Geophys., **53**, 1–15.

Chapman, C. H., 1985, Ray theory and its extensions: WKBJ and Maslov seismograms: J. Geophys., **58**, 27–43.

Chiu, S. K., Kanasewich, E. R., and Phadke, S., 1986, Three-dimensional determination of structure and velocity by seismic tomography, Geophysics, **51**, 1559–1571.

Cohen, E., Lyche, T., and Riesenfeld, R., 1980, Discrete *B*-splines and subdivision techniques: Comp.-Aided Geom. Design Comp. Graph., **14**, 87–111.

Conte, S. D., and de Boor, C., 1983, Elementary numerical analysis: An algorithmic approach: McGraw-Hill Book Co.

de Boor, K., 1978, A practical guide to splines: Springer-Verlag.

Deschamps, G. A., 1972, Ray techniques in electromagnetics: Proc. IEEE, **60**, 1022–1035.

Farra, V., 1987, Approche Hamiltonienne de la théorie des rais: Application à l'étude de perturbations, inversion des temps de parcours en sismique réflexion: Thèse, Université Paris.

——— 1990, Amplitude computation in heterogeneous media by ray perturbation theory: A finite element approach, Geophys. J. Internat., **103**, 341–354.

Farra, V., and Madariaga, R., 1987, Seismic waveform modeling in heterogeneous media by ray perturbation theory: J. Geophys. Res., **92**, 2697–2712.

——— 1988, Non linear reflection tomography: Geophys. J., **95**, 135–147.

Farra, V., Virieux, J., and Madariaga, R., 1989, Ray perturbation theory for interfaces, Geophys. J. Internat., **99**, 377–390.

Foley, J. D., and Van Dam, A., 1982, Fundamentals of interactive computer graphics: Addison Wesley Publ. Co.

Gjoystdal, H., Reinhardsen, J. E., and Ursin, B., 1984, Traveltime and wavefront curvature calculations in three-dimensional inhomogeneous layered media with curved interfaces: Geophysics, **49**, 1466–1494.

Haas, A., Dezard, Y., Guizou, J. L., and Duval, J., 1987, Seismic tomography in heterogeneous media: the stochastic inversion by ray continuation method: 57th Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 836–838.

Hanyga, A., 1988, Numerical methods for tracing rays and wavefronts, *in* Doornbos, D. J., Ed., Seismological algorithms: Computational methods and computer programs: Academic Press.

Inoue, H., 1986, A least-squares smooth fitting for irregularly spaced data: Finite element approach using the cubic *B*-splines basis, Geophysics, **51**, 2051–2066.

Julian, B. R., and Gubbins, D., 1977, Three dimensional seismic ray tracing: J. Geophys., **43**, 95–114.

Keller, H. B., and Perozzi, D. S., 1983, Fast seismic ray tracing: SIAM J. Appl. Math., **43**, 981–992.

Lane, J. M., and Carpenter, L. C., 1979, A generalized scan line algorithm for the computer display of parametrically defined surfaces, Comp. Graph. Image Process., **11**, 290–297.

Luneberg, R. K., 1944, Mathematical theory of optics, Brown University.

Lyche, T., and Morken, K., 1986, Making the Oslo algorithm more efficient, SIAM, **23**, 663–675.

Müller, G., 1984, Efficient calculation of Gaussian-beam seismograms for two-dimensional inhomogeneous media, Geophys. J. Roy. Astr. Soc., **79**, 153–166.

Pereyra, V., 1988, Two-point ray tracing in complex 3-D media: 58th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1056–1060.

Pereyra, V., Lee, W. H. K., and Keller, H. B., 1980, Solving two-point seismic ray tracing problem in a heterogeneous medium, Part I: A general adaptive finite difference method: Bull. Seis. Soc. Am., **70**, 79–99.

Sambridge, M. S., and Kennett, B. L. N., 1990, Boundary value ray tracing in heterogeneous medium: A simple and versatile algorithm: Geophys. J. Internat., **101**, 157–168.

Stone, B. D., and Forbes, G. W., 1990, Optimal interpolants for Runge-Kutta ray tracing in inhomogeneous media, J. Opt. Soc. Am., **7**, 248–254.

Virieux, J., 1991, Fast and accurate ray tracing by Hamiltonian perturbation: J. Geophys. Res., **96**, 579–594.

Virieux, J., Farra, V., and Madariaga, R., 1988, Ray tracing in laterally heterogeneous media for earthquake location: J. Geophys. Res., **93**, 6585–6599.

Ziolkowski, R. W., and Deschamps, G. A., 1980, The Maslov method and the asymptotic Fourier transform: Caustic analysis: Electromag. Lab. Sci. Rep. 80-9, Univ. Ill.

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

APPENDIX

SUBDIVISION OF BÉZIER AND *B*-SPLINE SURFACES

Let us consider one-dimensional interpolating functions in order to illustrate the geometrical properties of *B*-splines. The coordinate x can be interpolated from the x-coordinates of four control points using the relation:

$$x(u) = \underset{\sim}{\mathbf{U}}\underset{\sim}{\mathbf{M}}_s\underset{\sim}{\mathbf{P}}_x = [1\ u\ u^2\ u^3]\underset{\sim}{\mathbf{M}}_s\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \qquad \text{(A-1)}$$

where the matrix $\underset{\sim}{\mathbf{U}}$ are powers of the implicit parameter u ranging from 0 to 1. The expression holds also for Bézier interpolation where the matrix $\underset{\sim}{\mathbf{M}}_s$ (equation 19) has to be replaced by $\underset{\sim}{\mathbf{M}}_b$. The inversion of the relation $\underset{\sim}{\mathbf{M}}_s\underset{\sim}{\mathbf{P}}_x = \underset{\sim}{\mathbf{M}}_b\underset{\sim}{\mathbf{Q}}_x$ in order to obtain the control points $\underset{\sim}{\mathbf{Q}}$ for Bézier interpolation shows the equivalence between *B*-splines and Bézier interpolations.

The subdivision for Bézier interpolation is related to the geometrical construction of Figure A-1 and gives seven new control points from the four initial ones. The matrix associated to the geometrical construction is simply given by the relation:

$$\underset{\sim}{\mathbf{Q}}'_x = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/4 & 1/2 & 1/4 & 0 \\ 1/8 & 3/8 & 3/8 & 1/8 \\ 0 & 1/4 & 1/2 & 1/4 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \qquad \text{(A-2)}$$

where $\underset{\sim}{\mathbf{Q}}$ are the new seven deduced control points corresponding to two new patches with one common point. From the equivalence between Bézier interpolations and *B*-splines, we deduce explicitly the subdivision matrix for *B*-splines,

$$\underset{\sim}{\mathbf{P}}'_x = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/8 & 3/4 & 1/8 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 1/8 & 3/4 & 1/8 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \qquad \text{(A-3)}$$

used in our numerical implementation. Only five points are deduced for *B*-splines (Figure A-2) corresponding to two new patches with three common points; here a notable difference appears in geometric properties of Bézier and *B*-spline curves. Identical properties hold for surface description, because we use tensorial products, but the closed-form equation (A-3) of *B*-spline subdivisions given here is essential for efficiency.

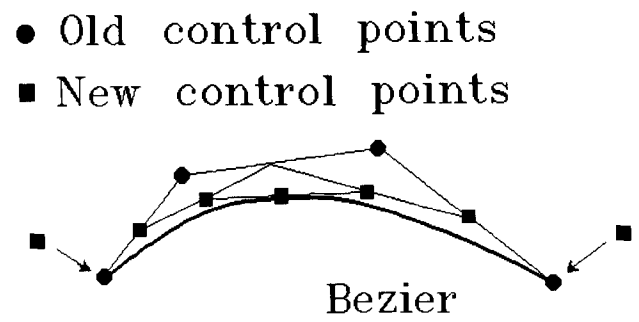

FIG. A-1. A geometric interpretation of the subdivision technique of Bézier curves.

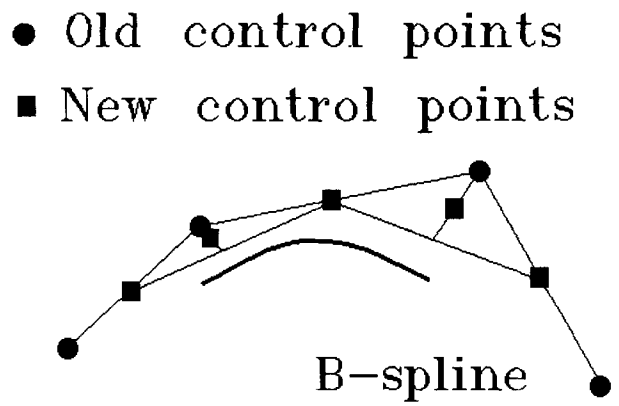

FIG. A-2. A geometric interpretation of the subdivision technique of *B*-splines.

Downloaded 07 Nov 2010 to 86.211.8.78. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/