Numerical Integration for Kirchhoff Migration

F. Assouline, P. Lailly

Numerical Integration for Kirchhoff Migration

F. Assouline[1] and P. Lailly[2]

1 Statoil Research, Arkitekt Ebbelsvei 10, Rotvoll, 7005 Trondheim - Norway
2 IFP, 1 et 4, avenue de Bois-Préau, 92500 Rueil-Malmaison Cedex - France
e-mail: frass@statoil.com - patrick.lailly@ifp.fr

Résumé — **Intégration numérique pour la migration de Kirchhoff** — Dans les acquisitions sismiques 3D, les points milieux d'une collection à déport constant sont distribués de façon irrégulière. Dans une telle situation, un simple « diffraction stack » ne peut réaliser correctement la migration de Kirchhoff car cette technique ne fournit pas une approximation consistante de l'intégrale de Kirchhoff. La solution à ce problème réside dans l'utilisation de véritables formules d'intégration numérique telles que celles basées sur des interpolations polynomiales de type Lagrange ou Hermite. Une intégration numérique basée sur l'interpolation de Lagrange peut parfaitemeent prendre en compte le caractère irrégulier de la distribution spatiale des points milieux, pour peu que ceux-ci soient suffisamment rapprochés. L'interpolation d'Hermite, bien qu'en théorie plus précise que la précédente, ne fournit qu'une approximation très relative de la qualité des images migrées : cette amélioration n'est en fait perceptible qu'au voisinage de l'événement migré. Les deux approches sont très faciles à implémenter : elles reviennent à faire une compensation adéquate (qui dépend du schéma d'interpolation retenu) de l'amplitude des différentes traces sismiques. La mise en place de ce simple prétraitement permet de transformer tout code de « diffraction stack » en une approximation de l'intégrale de Kirchhoff, ceci au prix d'une augmentation négligeable du temps de calcul. Enfin ce prétraitement permet à ces techniques d'intégration numérique d'être mises en œuvre conjointement à l'utilisation de filtres conçus pour éviter « l'aliasing de l'opérateur de migration ».

Abstract — *Numerical integration for Kirchhoff migration* — *In 3D seismic surveys, common offset data often involve an irregular distribution of midpoints. In such a situation, common offset Kirchhoff migration cannot be correctly performed by means of a mere discrete diffraction stack algorithm. Such an algorithm indeed corresponds to an inconsistent numerical integration formula. To overcome this difficulty, genuine numerical integration formulas (yielding a consistent approximation of the continuous diffraction stack) have to be used, e.g. numerical integration formulas based on polynomials leading to the so-called Lagrange or Hermite interpolations. A numerical integration formula based on Lagrange interpolation can cope with irregularly sampled midpoints provided that the density of midpoints involved in the common offset gather is sufficient. Besides, Hermite interpolation, more accurate in theory than the former, also provides relative improvement in the images at the vicinity of the migrated event,. Both techniques can be implemented by means of a simple preprocessing (adequate scaling) of the data. Thus they are quite easy to implement in any existing diffraction stack code. In addition, they can be used in combination with filters to prevent aliasing of the migration operator. The additional computation cost is negligible compared with the cost of running the diffraction stack itself.*

INTRODUCTION

Seismic imaging of complex geologic structures is still a challenging problem today. These structures are characterized by complicated interface geometries giving rise to strong lateral velocity variations (like those caused by salt bodies in the North Sea and in the Gulf of Mexico). In such cases, classical post-stack imaging techniques do not work. 3D prestack depth migration is recognized as the ideal technique provided an accurate velocity model has been determined. Migration velocity analysis (MVA) has accordingly aroused considerable interest for this purpose. This technique relies on a sound interpretation of 3D common offset migrated data.

In 3D seismic surveys, offset is a vector (characterized by a norm and an azimuth) and irregularities in acquisition (due for instance to streamer feathering in marine acquisition) are frequent. These irregularities give rise to offset variations (in norm and/or in azimuth) and to irregularly sampled data (*i.e.* nonuniform distribution of midpoints [1]). The influence of offset variations on migrated images has been investigated by [1]. This paper examines the influence of the second kind of irregularities, *i.e.* the influence of a nonuniform distribution of midpoints, on the quality of the migrated image and solutions are proposed to overcome the difficulties met when imaging such data.

For the study we use a simple kinematic Kirchhoff migration algorithm (a mere diffraction stack) without taking

account of dynamic effects, *i.e.* without attempting to preserve seismic reflection amplitudes. The conclusions we draw and the techniques proposed are nonetheless valid and suitable for a true-amplitude migration.

Using a test example, we first illustrate the difficulties encountered when imaging irregularly sampled data with a mere diffraction stack. We then analyze the underlying difficulty and show at this stage that a consistent numerical integration formula is required to obtain a correct image. For the sake of simplicity, we first examine the problem of numerical integration in 1D (for application to 2D migration) and then go on to 2D (for application to 3D migration), which is precisely the situation that interests us.

1 3D KIRCHHOFF MIGRATION: A TEST EXAMPLE

Irregularities in acquisition cause offset variations (in norm and/or azimuth) and an irregular distribution of midpoints. It has been shown [1] that mixing offsets has no drastic influence on the quality of migrated images if the offsets vary slowly with the midpoint coordinate. However, in the case of a non uniform distribution of midpoints, severe artefacts appear, as in Figures 2 and 3, which show common offset migrated synthetic data [2]. For this experiment, the 3D model shown in Figure 1 was used: two layers are separated by an anticline reflector exhibiting a relatively high curvature

---

(1) In this paper, midpoint means the true midpoint, not to be confused with the center of the associated bin.

(2) The offset variations involved in these data being small (the data come from a marine acquisition) we call a "common offset migration", a migration technique that actually does account for the actual locations of sources and receivers (and hence the offset variations).
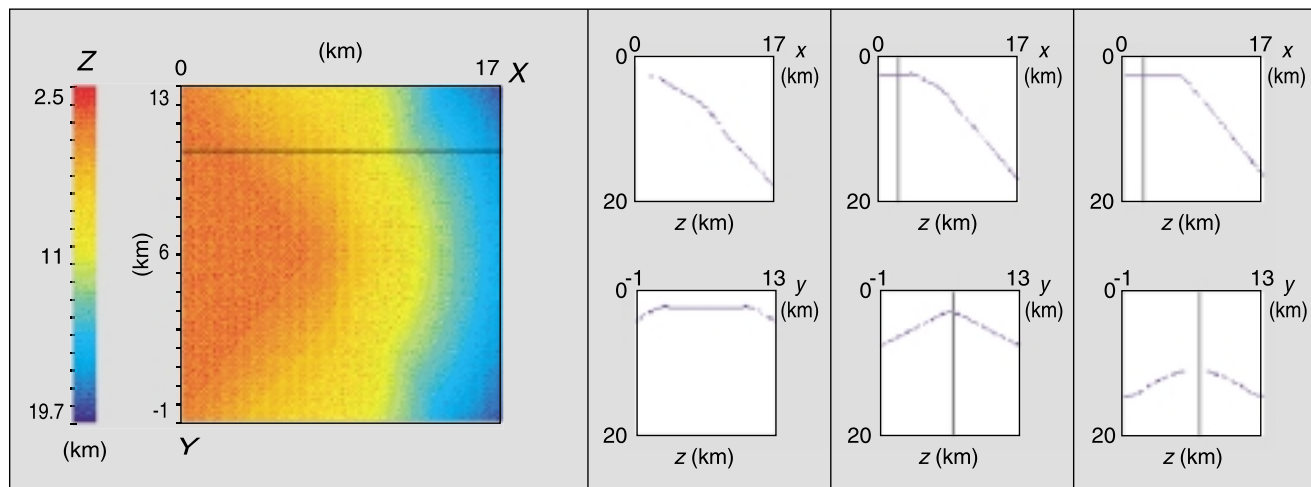

Figure 1

Synthetic 3D model. Map of interface (left) and different sections in the model:
– top row: constant *Y* sections, left to right *Y* = 2 km, *Y* = 4 km, *Y* = 6 km;
– bottom row: constant *X* sections, left to right *X* = 1 km, *X* = 7 km, *X* = 17 km.
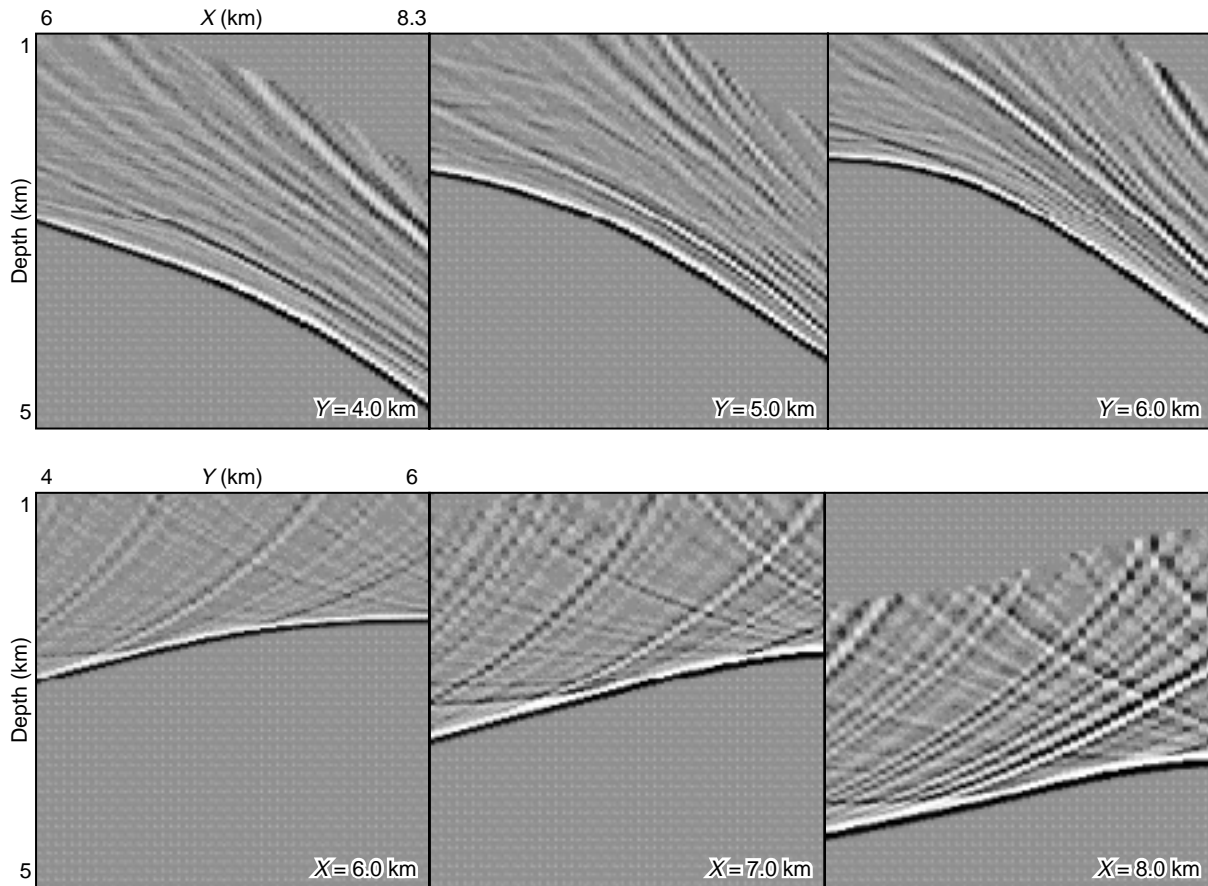The velocity above the reflector is constant (v = 3.0 km/s)

Figure 2

Small offset (0.2 km) migrated synthetic data associated with an actual marine acquisition geometry:
– constant Y sections (top);
– constant X sections (bottom).
In the depth migrated data volume obtained with the exact migration velocity ($v = 3.0$ km/s). The amplitude of the imaging artefacts is ten times lower than that of the migrated event corresponding to the reflector of the model.

and dips up to 45°. Above the interface the velocity is constant $v = 3.0$ km/s. The acquisition survey (source-receiver positions) used to compute the synthetic data comes from an actual marine acquisition in the North Sea. The synthetic data are generated by convolving travel-times from ray tracing [2] with a Ricker wavelet centered on 25 Hz. Our imaging technique is a mere diffraction stack. In the sequel we consider two subsets of data: one associated with small offsets (around 0.2 km) and the other with large offsets (around 3 km).

Figures 2 and 3 show, for the two considered offsets, some in-line and cross-line sections extracted from the depth migrated data volumes obtained with the exact velocity model: as expected, they show the migrated event that peaks on the reflector in the model, but they also show severe migration artefacts that strongly degrade the images (these artefacts are not visible at depths less than 2 km above the

reflector because, in order to save computing time, the image was not calculated above this limit; this will be the case throughout this paper). Note that the same difficulty appears for small and large offsets. These artefacts are strong enough to harm a migration velocity analysis process.

This paper proposes a way to remove these artefacts. To understand the difficulties met in imaging, we first present a review of some theoretical aspects involved in Kirchhoff migration.

2 THE UNDERLYING DIFFICULTY: AN ANALYSIS

2.1 Kirchhoff Migration:
the Continuous Diffraction Stack

For the sake of simplicity, we consider here an offset that does not vary with the midpoint coordinate. The common offset
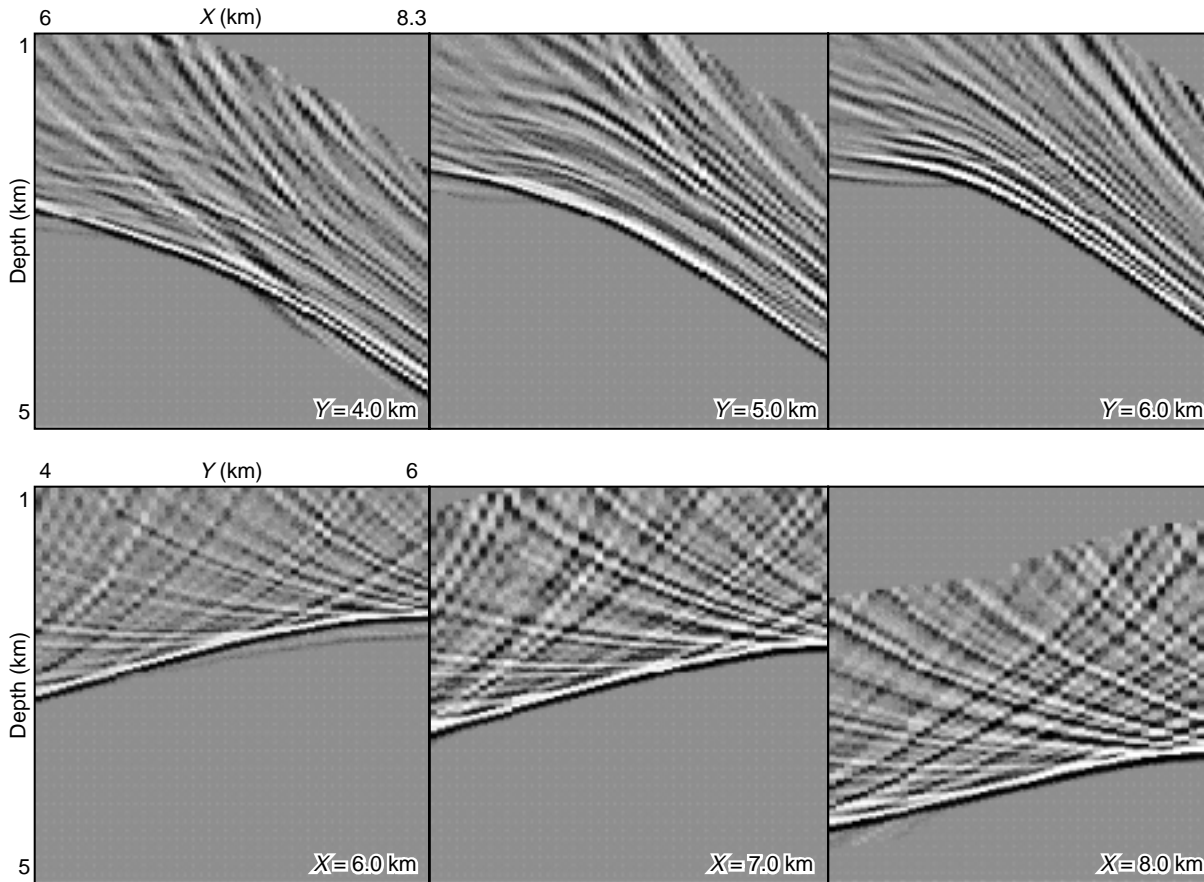
Figure 3

Large offset (3 km) migrated synthetic data associated with an actual marine acquisition geometry:
– constant *Y* sections (top);
– constant *X* sections (bottom).
In the depth migrated data volume obtained with the exact migration velocity ($v = 3.0$ km/s). The amplitude of the imaging artefacts is ten times lower than that of the migrated event corresponding to the reflector of the model.

migrated image $m_{\vec{h}}^{v}$ at a subsurface point **M** for an offset $\vec{h} = \vec{r} - \vec{s}$ (where $\vec{s}$ and $\vec{r}$ characterize the locations of the source *S* and the receiver *R*, respectively) and a velocity field *v*, is defined as the superposition of elementary migrated images m_{el}^{v}, each of those elementary images being characterized by a specific midpoint with location defined by vector $\vec{q} = \dfrac{\vec{s} + \vec{r}}{2}$:

$$m_{\vec{h}}^{v}(\mathbf{M}) = \int\limits_{q_x} \int\limits_{q_y} m_{el}^{v}(\mathbf{M}; \vec{q}, \vec{h})\, dq_x\, dq_y \qquad (1)$$

The value of the elementary image m_{el}^{v} at point **M** is obtained as:

$$m_{el}^{v}(\mathbf{M}; \vec{q}, \vec{h}) = \mathrm{d}\big(\vec{q}, \vec{h}, t_v(\mathbf{M}; \vec{q}, \vec{h})\big) \qquad (2)$$

where $\mathrm{d}(\vec{q}, \vec{h}, t)$ is the seismic trace associated with midpoint $\vec{q}$ and offset $\vec{h}$. $t_v(\mathbf{M}; \vec{q}, \vec{h})$ is the travel-time from source *S* to subsurface point **M** to receiver *R*, for the migration velocity field considered *v*.

Considering seismic data with a single event localized in the vicinity of the arrival time function $t_e(\vec{q}, \vec{h})$, the use of the stationary phase method ([3], for instance) yields the location of the support of the common offset migrated event.

It is the set of points **M** solution of the system:

$$\begin{cases} t_v(\mathbf{M}; \vec{q}, \vec{h}) - t_e(\vec{q}, \vec{h}) = 0 \\[2mm] \dfrac{\partial}{\partial q_i}\big(t_v(\mathbf{M}; \vec{q}, \vec{h}) - t_e(\vec{q}, \vec{h})\big) = 0 \qquad i = x, y \end{cases} \qquad (3)$$

Note that this result relies on different assumptions: high frequency asymptotics, smooth kinematics in the data and a smooth migration velocity field.

The first Equation of system (3), namely the phase condition, gives the geometry of an elementary migrated event: it is localized in the vicinity of the isochron surface, which is the set of possible reflection points **M** in the subsurface. The second Equation of system (3), namely the stationarity

condition, gives the geometry of the migrated event: it is the envelope of the family of isochron surfaces, parameterized by $\vec{q}$. Thus, common offset migrated data show reflections imaged as migrated events with the geometry described above regardless of the migration velocity field. This means that even if data are migrated with an erroneous migration velocity field, we can expect the migrated data to show organized events rather than a fuzzy image. These organized events are called migrated events: they show a deformed geometry of the associated reflectors, the image of the geologic interface being all the less deformed as the migration velocity is right.

2.2 Numerical Diffraction Stack

A widely used discrete diffraction stack involves a straightforward summation of the elementary migrated images (2):

$$m_{\overline{h}}^{v}(\mathbf{M}) = \sum_i m_{el}^{v}(\mathbf{M}; \vec{q}_i, \vec{h}) \qquad (4)$$

We now examine whether the migrated data obtained with this discrete formula can approximate the data that would be obtained by using the continuous formula (1). For the sake of simplicity, we consider a 2D migration, thus involving, for given $\mathbf{M}$ and h, a 1D integration in variable q. Let us consider the function to be integrated, $m_{el}^{v}(q)$, having the shape described in Figure 4. If the midpoint sampling is both regular and fine (small and constant sampling interval Δq), discrete formula (4) is (up to the constant factor Δq) a consistent approximation of the continuous diffraction stack: it is the classical rectangle formula for numerical integration of continuous functions. However, if the midpoint sampling

is irregular, *i.e.* if the midpoint distribution is not uniform, the discrete sum (4) is unlikely to approximate the integral of the continuous function. From the migration standpoint, this means that using the discrete sum (4) yields a mere superposition of elementary migrated events that can hardly be compared with the continuous diffraction stack: the constructive interferences between elementary migrated events that are expected to build the migrated event (the event that peaks on the envelope) only take place to a partial extent so that the migrated event shows up among a multitude of migration smiles (the elementary migrated events). This is observable in Figures 2 and 3. The important point at this stage is to realize that the difficulty lies in the use of the mere diffraction stack (4) in conjunction with a non uniform distribution of midpoints, and this difficulty persists even if the distance is very short between neighboring midpoints. This is illustrated by the following experiment.

Let us again consider the same acquisition survey as before, but with midpoint sampling refined by adding synthetic data to the original acquisition. The technique used for this refinement is explained in detail in the Section 5 *(Fig. 22)* but at this stage we only need to know that the midpoint distribution is still not uniform and that the mean distance between neighboring midpoints is two times shorter than before. Figure 5 shows the corresponding migrated data obtained for the large offset (3 km) and the exact velocity model: it still shows a multitude of migration smiles and the image quality (in terms of interpretability) is not improved at all as compared with the image shown in Figure 3, which confirms our hypothesis.

This paper proposes an imaging technique that can cope with non uniformly distributed midpoints. To do this, we
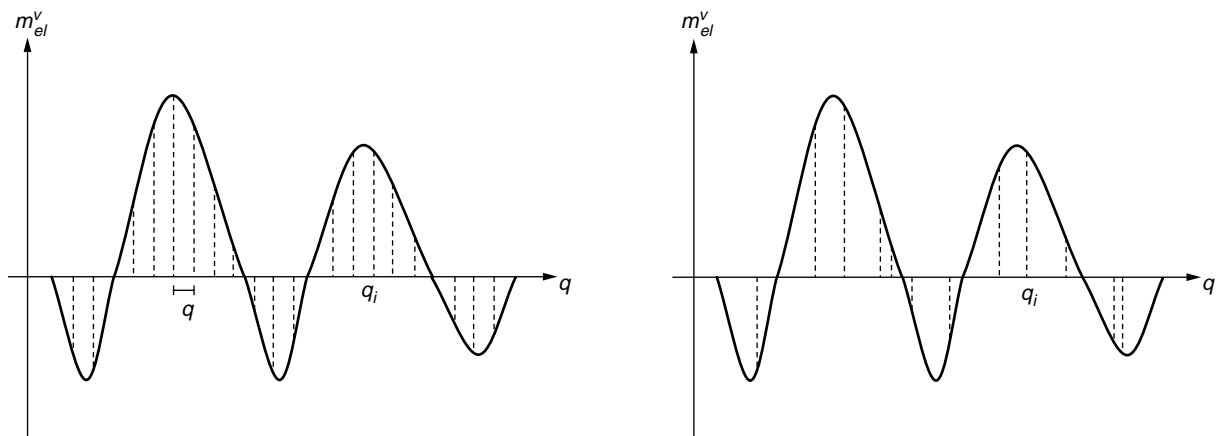

Figure 4

Sketch of the function m_{el}^{v} to be integrated. Left: the function is sampled with a constant spacing between discretization points q_i; in this situation, the sum of the samples approximates (up to the factor Δq) the integral of the continuous function (rectangle formula). Right: the function is sampled with a nonconstant spacing between discretization points q_i; in this situation, the sum of the samples can hardly be compared with the integral of the continuous function.
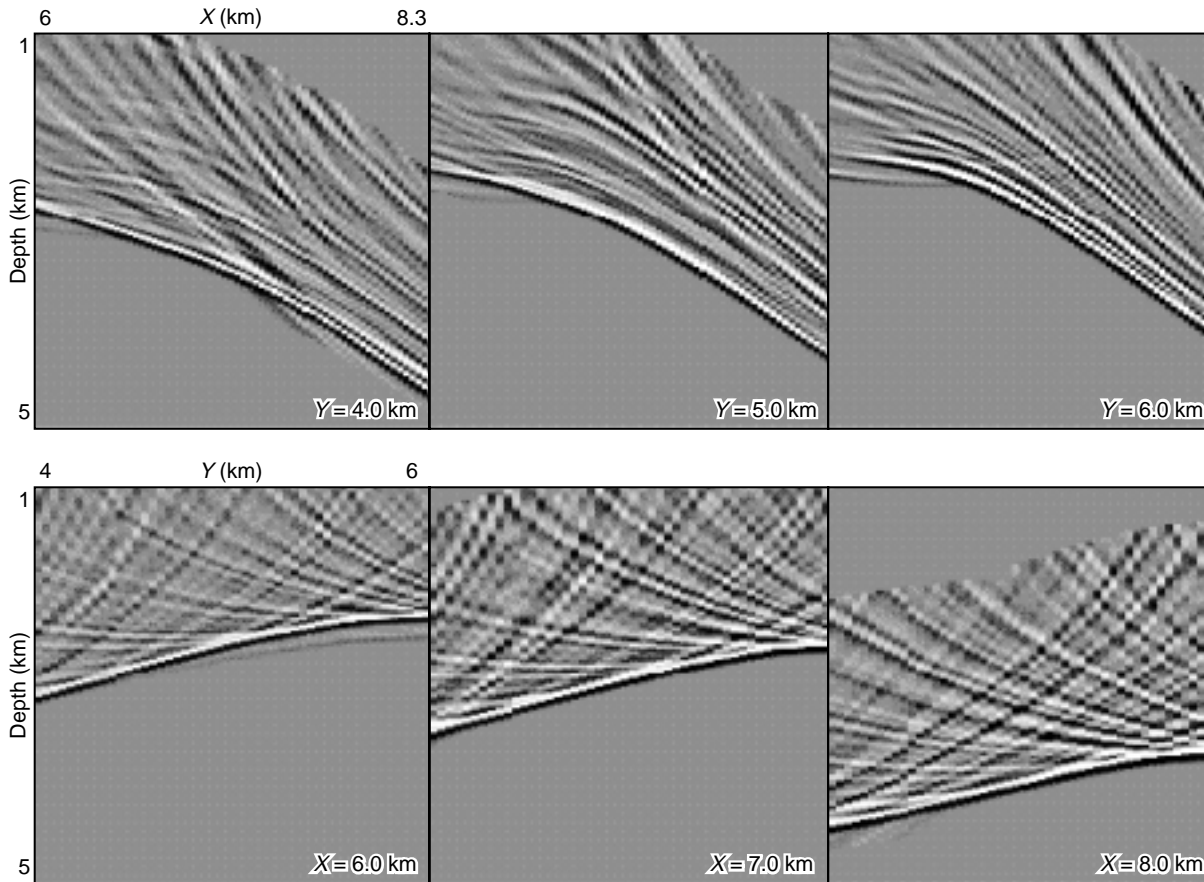
Figure 5

Large offset (3 km) migrated synthetic data associated with the refined actual marine acquisition geometry: constant *Y* sections (top) and constant *X* sections (bottom) in the depth migrated data volume obtained with the exact migration velocity ($v = 3.0$ km/s). The amplitude of the imaging artefacts is ten times lower than that of the migrated event corresponding to the reflector of the model.

have to make use of a consistent (meaning that if the distances between neighboring midpoints go to zero, the result provided by the numerical integration formula converges to the integral of the continuous function) and, if possible, accurate numerical integration formula. The derivation and testing of these formulas are the goals of the subsequent sections. However, before opening this subject, we want to state what can and cannot be expected from a consistent numerical integration formula. In the case of an exact migration velocity [3]:

– We expect that, when subsurface point **M** is below the reflector, the numerical integration formula yields a zero value for the migrated image. This condition is, in fact, almost always fulfilled as elementary migrated images are in general zero below the reflector (however, some

elementary migrated events could cross a strongly tilted envelope).
– We expect that, when subsurface point **M** is on the reflector, even though we do not want to preserve amplitudes (but to approximate the continuous diffraction stack), the numerical integration formula gives an indication of the reflectivity for the considered offset. Looking at the values taken by the different elementary migrated images for such a point **M** *(Fig. 6)*, we find that the function to be integrated is smooth and that our sampled data give detailed information on this function (note, however, that if the midpoint sampling were too coarse (holes in the acquisition), the samples would not correctly represent the continuous function. This is a hopeless situation: no numerical integration formula can create missing information). All we can expect from a numerical integration formula is that it gives a good approximation of the integral of the continuous function. From the stationary phase approximation, we know that the only samples that con-

_____

(3) These expectations also hold in the case of an erroneous migration velocity provided that we replace "reflector" by "migrated event".

Figure 6

Values of function m^v_{el} at regularly spaced midpoints $\vec{q}$ with a 50 m spacing both in the x and y directions, for a subsurface point located on the interface. Left: 3D view of the function. Right: section obtained for $q_x = 2.6$ km.
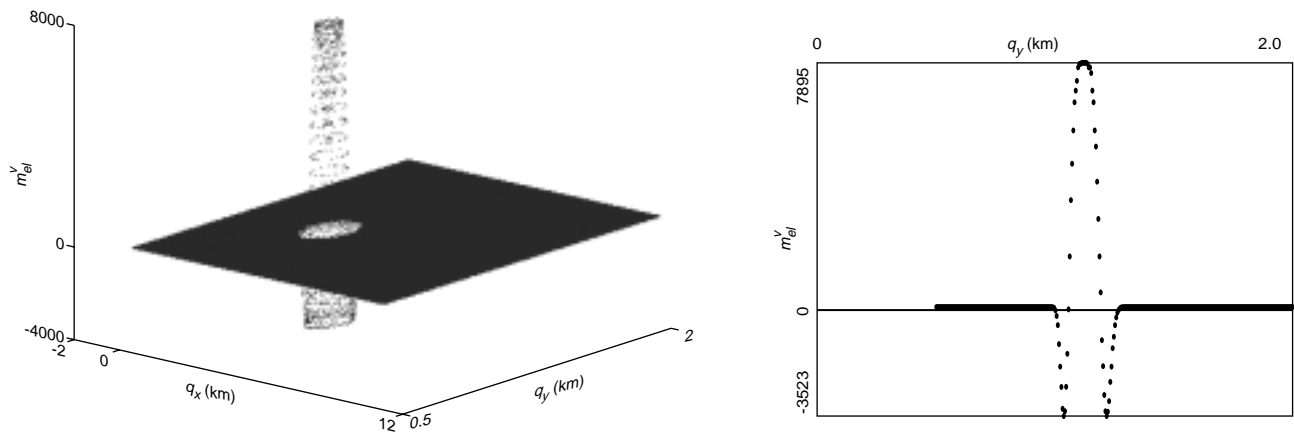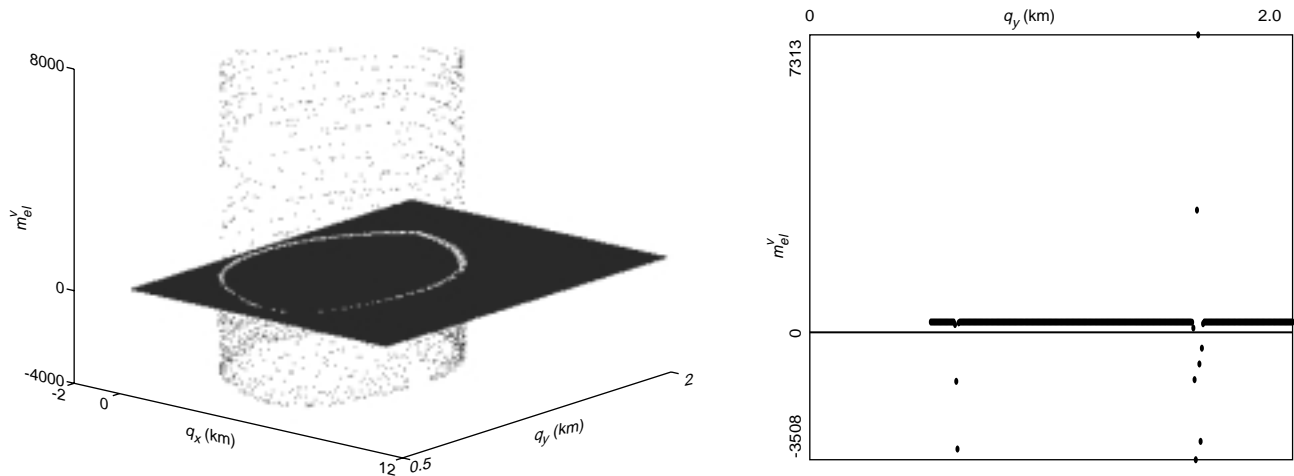

Figure 7

Values of by function m^v_{el} at regularly spaced midpoints $\vec{q}$ with a 50 m spacing both in the x and y directions, for a subsurface point located 2.5 km above the interface. Left: 3D view of the function. Right: section obtained for $q_x = 4.0$ km.

tribute to the reflectivity evaluation are those in the neighborhood of the point where the phase is stationary (the point with highest positive amplitude) and that the contribution of all the other samples must cancel each other out.

– However, looking at the values of the different elementary migrated images for a point **M** located 2.5 km above the reflector *(Fig. 7)*, we cannot expect an accurate result from the best numerical integration formula: the function to be integrated is not sampled finely enough regarding its very rapid variations. In such a situation, the migration operator turns out to be aliased and, the only alternative to another survey, is to apply filters, as described *e.g.* by [4] or [5], which in fact make the function to be integrated vary more slowly.

2.3 The Underlying Difficulty: Conclusion

This analysis shows that if midpoints are irregularly sampled in an acquisition, Kirchhoff migration cannot be correctly performed by using a mere discrete diffraction stack. A consistent numerical integration is needed to approximate the continuous diffraction stack. Besides, at points **M** far above the reflector, the function to be integrated varies very rapidly with midpoint coordinate. In this situation, midpoint sampling is usually not fine enough to correctly represent the function to be integrated, thereby entailing the use of appropriate filters to prevent aliasing in the migration operator. The use of these filters in combination with a consistent and, if possible, accurate numerical integration formula is the key

for obtaining good migrated images. Designing such filters has been the object of different studies (see for instance [4] or [5]). We ignore this problem in the sequel to focus rather on the design of consistent and accurate numerical integration formulas for use in combination with such filters. We describe the principle of the derivation of such formulas in a 1D situation (*i.e.* for applications to 2D migration) for information, and then switch to the materially important 2D situation (for application to 3D migration).

3 NUMERICAL INTEGRATION IN 1D

3.1 Principle

Numerical integration consists in calculating an approximation of the integral, over some interval [a, b], of a function $f(x)$ from its values f at different points x_i (called in the sequel *interpolation nodes* in the sequel) in [a, b]. Let these interpolation nodes be numbered from 1 to I, assuming that $x_1 = a$ and $x_I = b$. A straightforward approach to numerical integration is based on interpolation procedures: we calculate a function $f_{int}(x)$ (*int* stands for interpolating) that interpolates the values $f(x_i)$ (Lagrange interpolation) and possibly the first order derivatives $f'(x_i)$ or even higher order derivatives (Hermite interpolation) and then approximate $\int_a^b f(x)\, dx$ by $\int_a^b f_{int}(x)\, dx$. This procedure assumes that function $f_{int}(x)$ approximates function $f(x)$ which, obviously, requires that the distance between successive points x_i be small enough regarding the roughness of function f.

This section introduces the basic elements involved in numerical integration in a simple 1D situation, enabling us to be concise in the more complex 2D situation.

3.2 Numerical Integration by Means of Lagrange Interpolation

3.2.1 Lagrange Interpolation

We begin with the simplest interpolation procedure, namely the Lagrange interpolation: this interpolation only uses the values $f(x_i)$. We also deal with its simplest implementation (known as P^1 finite element), in which the interpolation function is piecewise linear *(Fig. 8)*.

The set of these interpolation functions is the vector space generated by *basis functions* as described in Figure 9, each of them associated with one interpolation node. These basis functions are the piecewise first order polynomials defined by:

$$e_i(x_j) = \delta_{ij} \quad \forall i, j = 1, \ldots, I \tag{5}$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if not.

Thus the interpolation function $f_{int}(x)$ can be expressed as:

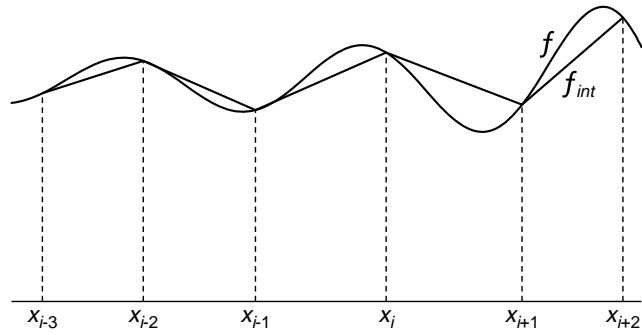$$f_{int}(x) = \sum_{i=1}^{I} f(x_i)\, e_i(x) \tag{6}$$

Figure 8

In the simplest 1D Lagrange interpolation procedure we interpolate between points x_i by a linear function. The interpolation function has C^0 regularity.
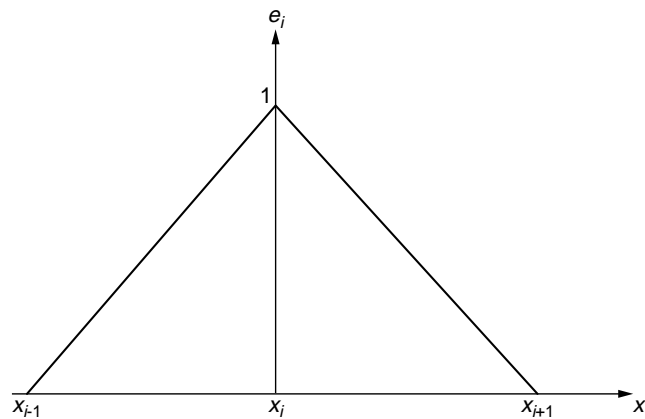

Figure 9

Lagrange basis function $e_i(x)$: this function is associated with interpolation node with abscissa x_i.

3.2.2 Numerical Integration Formula

The numerical integration formula is obtained by integrating function $f_{int}(x)$ which, from (6), gives:

$$\int_a^b f_{int}(x)\, dx = \sum_{i=1}^{I} f(x_i)\, w_i \tag{7}$$

with:

$$w_i = \int_a^b e_i(x)\, dx = \frac{1}{2}(x_{i+1} - x_{i-1}) \tag{8}$$

From Equation (7) we observe that numerical integration appears as a weighted summation of the values $f(x_i)$, the *weights* being the numbers w_i defined in (8). From Figure 8 we also see that this numerical integration is an application of the very classical trapezoidal formula.

3.2.3 Error Estimate

To determine the accuracy of the numerical integration formula, it is important to calculate an error estimate denoted by E_L: this estimate gives a bound of the error (*i.e.* the absolute value of the difference between $\int_a^b f(x)\,dx$ and $\int_a^b f_{int}(x)\,dx$). More precisely, it allows us to highlight the dependency of the error on the *typical distance*[4] h between interpolation nodes (this distance can, for instance, be the minimum distance between successive interpolation nodes) and on the smoothness of function. This smoothness is quantified by a norm involving the derivatives of function $f(x)$. The error estimate, whose derivation is recalled in Appendix A, is of the form:

$$E_L = C_f^L (b-a) h^2 \qquad (9)$$

where C_f^L is a constant that depends on the norm of $f(x)$ in $C^2([a, b])$. In E_L and C_f^L the subscript and superscript L refer to the technique based on Lagrange interpolation.

3.3 Numerical Integration by Means of Hermite Interpolation

3.3.1 Hermite Interpolation

To obtain a more accurate numerical integration procedure (*i.e.* a procedure yielding an error estimate showing for instance a h^3 dependency) more sophisticated interpolation procedures must be followed including techniques based on Hermite interpolation. In Hermite interpolation, we interpolate, at interpolation nodes x_i, the values of the derivatives of function $f(x)$ as well as the values of the function itself *(Fig. 10)*. We shall restrict ourselves to the simplest case where only first order derivatives are taken into account.

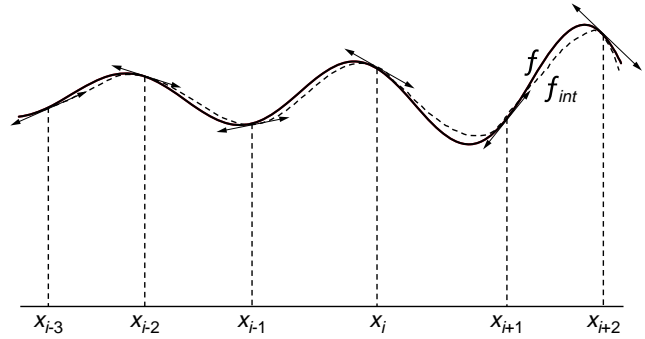
(4) Not to be confused with offset vector $\vec{h}$.

Figure 10

Hermite interpolation in 1D: the interpolation function $f_{int}(x)$ matches the values $f(x_i)$ and $f'(x_i)$.

It is straightforward to check that the interpolation function can be expressed as:

$$f_{int}(x) = \sum_{i=1}^I f(x_i)\, e_i^0(x) + \sum_{i=1}^I f'(x_i)\, e_i^1(x) \qquad (10)$$

where the *basis functions* $e_i^0(x)$ and $e_i^1(x)$ are piecewise third degree polynomials defined by:

$$e_i^0(x_j) = \delta_{ij} \quad \text{and} \quad \frac{de_i^0}{dx}(x_j) = 0 \quad \forall i,j = 1, \ldots, I \qquad (11)$$

$$e_i^1(x_j) = 0 \quad \text{and} \quad \frac{de_i^1}{dx}(x_j) = \delta_{ij} \quad \forall i,j = 1, \ldots, I \qquad (12)$$

These basis functions are shown in Figure 11. It is straightforward to obtain the analytic expression of the basis functions over some interval $[x_j, x_{j+1}]$, $j = 1, \ldots, I-1$: Equations (11) or (12) give the information for determining, in a unique way, the four coefficients defining a third
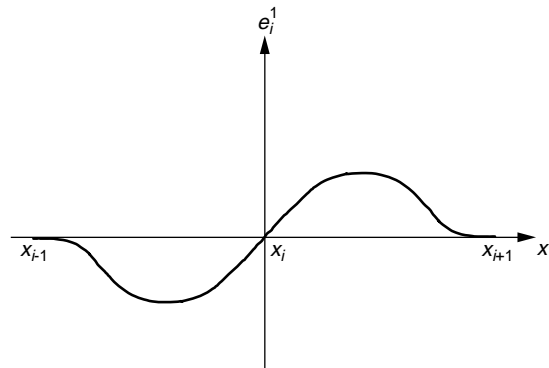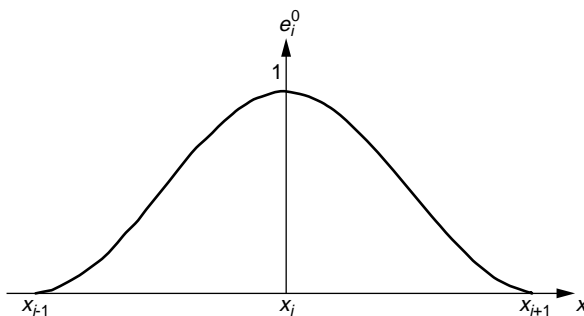
Figure 11

Basis functions $e_i^0(x)$ (left) and $e_i^1(x)$ (right) associated with Hermite interpolation.

degree polynomial. The uniqueness in this determination shows that, for any Hermite interpolation data $f(x_i)$ and $f'(x_i)$, $i = 1, \ldots, I$, there exists a unique function which is a third degree polynomial over each interval $[x_j, x_{j+1}]$, $j = 1, \ldots, I - 1$, that matches the Hermite interpolation data. It is important to note that the interpolation function f_{int} now has C^1 regularity as we have imposed a given value for its derivative at the boundaries of intervals $[x_j, x_{j+1}]$, $j = 1, \ldots, I - 1$.

3.3.2 Numerical Integration Formula

As already stated, the numerical integration formula is obtained by integrating $f_{int}(x)$. We obtain:

$$\int_a^b f_{int}(x) \, dx = \sum_{i=1}^I f(x_i) w_i^0 + \sum_{i=1}^I f'(x_i) w_i^1 \qquad (13)$$

with $w_i^0 = \int_a^b e_i^0(x) \, dx$ and $w_i^1 = \int_a^b e_i^1(x) \, dx$

Now the numerical integration formula appears not only as a linear combination of the values of $f(x_i)$ but also of the values of $f'(x_i)$. Note that coefficients w_i^0 and w_i^1 are straightforward to calculate since functions e_i^0 and e_i^1 are third degree polynomials.

3.3.3 Error Estimate

As shown in Appendix B, the error estimate is of the form

$$E_H = C_f^H (b - a) h^4 \qquad (14)$$

where C_f^H is a constant that now depends on the norm of $f(x)$ in $C^4([a, b])$. H used as subscript or superscript refers to Hermite interpolation. Thus, for a smooth function f (having not too large 4th order derivative), the numerical integration procedure based on Hermite interpolation is more accurate than that based on Lagrange interpolation (P^1 finite elements): the former yields an error estimate in $O(h^4)$ *(see Eq. (14))* whereas the latter yields an error estimate in $O(h^2)$ *(see Eq. (9))*.

3.3.4 Numerical Differentiation

In the problem at hand, the values $f'(x_i)$ are not directly available. It is hence normal to try to estimate them from the values $f(x_i)$ by a numerical differentiation procedure such as the finite difference technique. If we wish to preserve the good accuracy associated with Hermite interpolation, the numerical differentiation procedure must also be accurate enough. This section describes a numerical differentiation procedure that gives an error in $O(h^n)$ with n chosen such that the overall error involved in the resulting numerical integration formula is of a higher order than that involved in Lagrange interpolation *(Eq. (9))*.

We begin with the classical description of the numerical differentiation procedure (see for instance [6]) and the derivation of the associated error estimates. We wish to estimate the derivative of function $f(x)$ at a given interpolation node, which we place, by an appropriate translation of the origin, at $x = 0$. We want the derivative to have the form of a linear combination of the values $f(x_i)$ (the derivation operator is linear) plus an error estimate in $O(h^n)$:

$$\frac{df}{dx}(0) = \sum_k \alpha_k f(x_k) + O(h^n) \qquad (15)$$

Here, the subscript k refers to a local (*i.e.* relative to the node that has been placed at $x = 0$ which corresponds to $k = 0$) numbering *(Fig. 12)* of nodes (as opposed to subscript i associated with the overall node numbering). The summation on k only takes place at interpolation nodes located in the neighborhood of the one at $x = 0$, the latter being included. The numbers α_k (as well as the number of neighbor interpolation nodes that must be included in the summation) are determined so as to yield an error estimate in $O(h^n)$.

The derivation of the error estimate is obtained classically by means of Taylor's expansions. We write for any interpolation node x_k, including $x_k = 0$:

$$\begin{aligned} f(x_k) = f(0) &+ x_k \frac{df}{dx}(0) + \frac{1}{2!} x_k^2 \frac{d^2 f}{dx^2}(0) \\ &+ \frac{1}{3!} x_k^3 \frac{d^3 f}{dx^3}(0) + \frac{1}{4!} x_k^4 \frac{d^4 f}{dx^4}(\xi_k) \end{aligned} \qquad (16)$$

where ξ_k is between 0 and x_k. Calculating $\sum_k \alpha_k f(x_k)$ with $f(x_k)$ given by Equation (16) we find that if numbers α_k
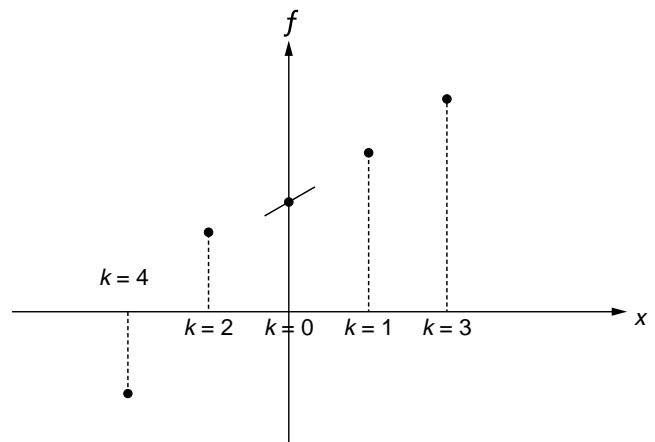

Figure 12

Numerical differentiation in 1D. In this procedure we try to estimate the value of the derivative at the node at $x = 0$ from the values of the function at neighboring nodes. Note the use of a local numbering (index k) of nodes, the one at $x = 0$ corresponding to $k = 0$.

satisfy the system:

$$\sum_k \alpha_k = 0 \tag{17}$$

$$\sum_k \alpha_k x_k = 1 \tag{18}$$

$$\sum_k \alpha_k x_k^2 = 0 \tag{19}$$

$$\sum_k \alpha_k x_k^3 = 0 \tag{20}$$

we then have:

$$\sum_k \alpha_k f(x_k) = \frac{\mathrm{d}f}{\mathrm{d}x}(0) + \frac{1}{4!} \sum_k \alpha_k x_k^4 \frac{\mathrm{d}^4 f}{\mathrm{d}x^4}(\xi_k) \tag{21}$$

As shown in Appendix C, the term $\sum_k \alpha_k x_k^4 \frac{\mathrm{d}^4 f}{\mathrm{d}x^4}(\xi_k)$ is $O(h^3)$ so that finally Equation (15) holds with approximation order $n = 3$.

This accuracy requires numbers α_k to satisfy the system of four linear equations (17)–(20): hence four interpolation nodes are taken into account (including the one at $x = 0$). If we want Equation (15) to hold only with approximation order $n = 2$ we simply need system (17)–(19) to be fulfilled, thus having to account for only three interpolation nodes. This yields the simplest implementation for a numerical estimation of derivatives that gives an overall error *(see Appendix D)* for the resulting numerical integration formula in $O(h^3)$.

4 NUMERICAL INTEGRATION IN 2D

The principle of numerical integration in 2D is the same as in 1D: we construct an interpolation function and then calculate the integral, on the considered area, of the interpolation function, which is theoretically simple since the interpolation function is a piecewise polynomial.

4.1 Interpolation in 2D and Finite Elements

4.1.1 The Interpolation Problem

In 2D interpolation consists in constructing a function $f_{int}(x, y)$ that assumes specified values at the interpolation nodes and, in the case of Hermite interpolation, whose gradient (or even higher order derivatives) also assumes specified values at these interpolation nodes. The interpolation nodes are, of course, distributed on the area over which the integral is to be calculated. Returning to the physics of the problem, the interpolation nodes are the midpoints involved in the common offset gather: the locations of these midpoints are defined by vectors $\vec{q}_i$, $i = 1, \ldots, I$. Integration takes place in the midpoint coordinates *(see Eq. (1))*. We shall use the
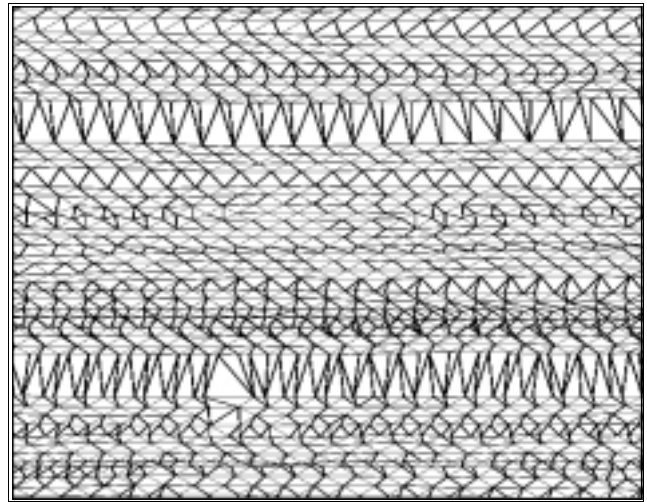

Figure 13

An example of triangulation. Note that triangles can have summits or edges in common but not pieces of edges.

classical technique of finite elements (see [7] for instance) to solve this interpolation problem. More specifically, we shall use simplectic finite elements here: we start by splitting the integration domain into non overlapping triangles whose summits are the interpolation nodes *(Fig. 13)*. This splitting is called triangulation. More precisely, we assume that in such a triangulation, no summit lies inside the edge of another triangle: triangles can have summits or edges in common but not pieces of edges. We also assume that the integration domain, denoted by Ω, is the union of the triangles included in the triangulation.

4.1.2 The Technology of Finite Elements

The technology of finite elements is based on discrete functions (*i.e.* functions depending on a finite number of parameters). These discrete functions are polynomials on each triangle of the triangulation considered, thus allowing easy computation of basic mathematical quantities: the values of the function or of some of its derivatives or its integral on some domain, etc. The nature of the polynomial function depends in particular on the problem to be solved (Lagrange *versus* Hermite interpolation in our case). The discrete functions considered belong to a (finite dimensional) vector space generated by some *basis functions:* any discrete function is a linear combination of these basis functions. The above mentioned computation of basic mathematical quantities is then straightforward once the basis functions are exhibited. In the finite elements we consider, the support of these basis functions is a set of triangles at the vicinity of a specific interpolation node *(Fig. 14)*. It is normal to define the basis functions piece by piece, *i.e.* separately in each of
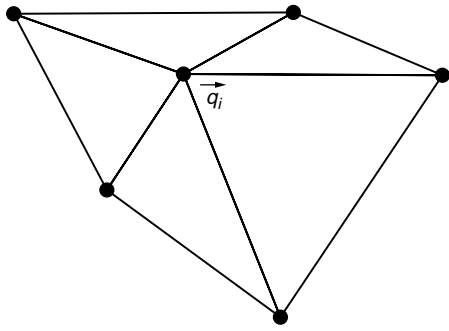
Figure 14

In the different finite elements considered in this paper, the basis functions are associated with a specific vertex of the triangulation whose location is defined by vector $\vec{q}_i$ (and, for Hermite interpolation, with the quantity to be interpolated at this vertex). The basis functions have their support localized on the triangles that surround the vertex considered.
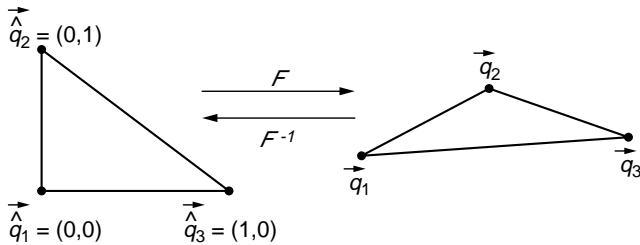

Figure 15

Piecewise construction of some basis functions associated with node $\vec{q}_i$. To construct the piece of basis function associated with triangle T on the right, we use the affine invertible transformation F^{-1} that maps it onto the reference triangle on the left. We introduce a local (*i.e.* attached to the triangle considered) node numbering: node $\vec{q}_i$ in the global numbering becomes $\vec{q}_1$ in the local numbering. Note that these nodes $\vec{q}_1$ have a specific role: the vertex shared by all the triangles in Figure 14. This affine transformation maps $\vec{q}_1$, $\vec{q}_2$, $\vec{q}_3$ into points with coordinates (0,0), (0,1) and (1,0), respectively. The basis function on the right hand triangle is straightforwardly obtained once the basis function on the reference triangle is exhibited. This task is simplified in practice by using barycentric coordinates.

these triangles, and then to assemble the contribution of the different pieces.

It is classic to construct a piece of basis function associated with a specific triangle from a piece of basis function defined on a reference triangle, and to obtain the piece of basis function on the triangle of interest using the affine transformation that maps it into the reference triangle. This construction relies on the following general result.

Theorem: Consider two triangles (Fig. 15), one of them being considered as a reference triangle. Consider the affine invertible transformation F that maps the reference triangle $\hat{T}$ onto the other triangle T. Consider a general interpolation problem on triangle $\hat{T}$: we seek a function within a specific

space (or subspace) of polynomials (for instance, the space of third order polynomials that match a given linear equation), this space being called the space of interpolation functions and denoted by P, whose values, or the values of some derivatives, at some points (called interpolation nodes) in triangle $\hat{T}$, match some specified values (called interpolation values). Assume that this interpolation problem in triangle $\hat{T}$ gives rise to a uniquely defined interpolation operator $\hat{\Pi}$: for whatever specified interpolation values, there exists a unique interpolation function in P that matches these interpolation values. Then the associated interpolation problem on triangle T (the problem in which the interpolation values are to be matched at the interpolation nodes defined as the images by F of the interpolation nodes in triangle $\hat{T}$) has a unique solution in the space of interpolation functions defined by the composition of polynomials in P with F^{-1}.

This result is obvious for simple interpolation problems such as the simplest implementation of Lagrange interpolation but, contrary to what may appear at first glance, is not that obvious for more complex problems, such as those associated with Hermite interpolation. The reader can refer to textbooks on the finite elements method such as [8], for a proof of this theorem.

From a practical standpoint, this theorem also gives the construction of a piece of basis function in any triangle once the basis function has been exhibited on the reference triangle. Barycentric coordinates play an important role here since they remain invariant in the affine transformation considered. In general, the use of barycentric coordinates greatly simplifies the expression of basis functions in triangle T. If the locations of the vertices of triangle T are defined by vectors $\vec{q}_1$, $\vec{q}_2$, $\vec{q}_3$, the barycentric coordinates λ_1, λ_2, λ_3 of a point whose location is defined by vector $\vec{q}$ are defined as the solution of:

$$\vec{q} = \sum_{j=1}^{3} \lambda_j \vec{q}_j \tag{22}$$

with:

$$\sum_{j=1}^{3} \lambda_j = 1 \tag{23}$$

It is easy to check that the barycentric coordinates of the vertex at $\vec{q}_1$ are (1, 0, 0) and that λ_1 remains constant while moving in a direction parallel to vector $\vec{q}_3 - \vec{q}_2$.

The nature of the affine transformation F also plays an important role in the derivation of error estimates, which is an important point for our goal of accurate numerical integration (see [8] for a detailed analysis). All this is rather formal, merely the guideline for solving our 2D Lagrange and Hermite interpolation problems. In addressing these problems, we will stop being formal and become very concrete.

4.2 Numerical Integration by Means of Lagrange Interpolation

4.2.1 First Order Lagrange Interpolation in 2D (P^1 Finite Elements)

In 2D Lagrange interpolation, we calculate an interpolation function that assumes specified values at the interpolation nodes. The simplest choice (first order Lagrange interpolation) consists in finding an interpolation function that is affine on each triangle in the triangulation (P^1 finite elements): this function is defined in a unique way from the values specified at the interpolation nodes. This interpolation function obviously displays C^0 regularity. It is straightforward to realize that the interpolation function can be written as:

$$f_{int}(\vec{q}) = \sum_i f(\vec{q}_i)e_i(\vec{q}) \qquad (24)$$

where basis function $e_i(\vec{q})$ is the piecewise affine function such that:

$$e_i(\vec{q}_j) = \delta_{ij} \qquad (25)$$

Its support is localized on the triangles around node $\vec{q}_i$ *(Fig. 14)*. To exhibit the piece of basis function associated with a triangle which has $\vec{q}_i$ as summit we use a reference triangle: the affine transformation F^{-1} that maps the original triangle onto the reference triangle maps $\vec{q}_i$ into the point with coordinates $(0, 0)$ *(Fig. 15)*. The basis function on the reference triangle is quite simple. In terms of barycentric coordinates its expression is λ_1 (1 is the number of the summit with coordinates $(0, 0)$). It is also the expression of the piece of basis function for the right hand triangle in Figure 15: barycentric coordinates are invariant in the affine transformation.

4.2.2 Numerical Integration

Thanks to Equation (24) integration of the interpolation function is quite simple. The integral is a weighted summation of the values $f(\vec{q}_i)$:

$$\int_\Omega f_{int}(x, y)\,\mathrm{d}x\,\mathrm{d}y = \sum_i w_i f(\vec{q}_i) \qquad (26)$$

with:

$$w_i = \int_\Omega e_i(\vec{q})\,\mathrm{d}x\,\mathrm{d}y \qquad (27)$$

In fact, since basis functions are simple first degree polynomials on each triangle, w_i has a rather simple expression: it is one third of the sum of the areas of the triangles surrounding node i.

4.2.3 Error Estimate

We wish to estimate the error E_L involved in the numerical integration formula (26), namely the error made in replacing function $f(x, y)$ by function $f_{int}(x, y)$ in the integral. The derivation of the error estimate (see [8] for instance) is fairly technical and we merely give the result:

$$E_{L \le} \|f - f_{int}\|_{L^1(\Omega)} \le C_f^L h^2 \qquad (28)$$

Here h is the characteristic dimension of the triangles in the triangulation, constant C_f^L depends on the deformations of the triangles in the triangulation (the best triangles are equilateral triangles) and on the norm of function f in the Sobolev space $W^{2,1+\varepsilon}(\Omega)$ (ε is an arbitrary strictly positive number). Such a norm is large whenever the second order derivatives of function f are large. Note that, for this error estimate, we find the same order as in the 1D situation.

4.3 Numerical Integration by Means of Hermite Interpolation

4.3.1 A Finite Element for 2D Hermite Interpolation

We now wish to find an interpolation function $f_{int}(\vec{q})$ which takes specified values at the vertices of the triangulation and whose gradient also takes specified values at these vertices. Function $f_{int}(\vec{q})$ thus has to fulfill the system of equations:

$$f_{int}(\vec{q}_i) = f(\vec{q}_i) \qquad (29)$$
$$\nabla_x f_{int}(\vec{q}_i) = \nabla_x f(\vec{q}_i) \qquad (30)$$
$$\nabla_y f_{int}(\vec{q}_i) = \nabla_y f(\vec{q}_i) \qquad (31)$$

where ∇_x and ∇_y are the x and y components of the gradient operator $\vec{\nabla}$, respectively. Quantities $f(\vec{q}_i)$, $\nabla_x f(\vec{q}_i)$, $\nabla_y f(\vec{q}_i)$ are given data for $i = 1, \ldots, I$.

Note that this interpolation function is a well-known finite element that is described for instance in the book of [9] (Chapter 10: bending of plates). We also suggest [8] (Chapter 5, example 2.2) for a complete treatment of the problem.

As usual we shall now seek an interpolation function $f_{int}(\vec{q})$ which is a superposition of basis functions:

$$f_{int}(\vec{q}) = \sum_{i=1}^{I} f(\vec{q}_i)\, e_i^0(\vec{q}) + \sum_{i=1}^{I} \nabla_x f(\vec{q}_i)\, e_i^x(\vec{q}) \\ + \sum_{i=1}^{I} \nabla_y f(\vec{q}_i)\, e_i^y(\vec{q}) \qquad (32)$$

where basis function $e_i^0(\vec{q})$ is defined by:

$$e_i^0(\vec{q}_j) = \delta_{ij} \qquad (33)$$
$$\nabla_x e_i^0(\vec{q}_j) = 0 \qquad (34)$$
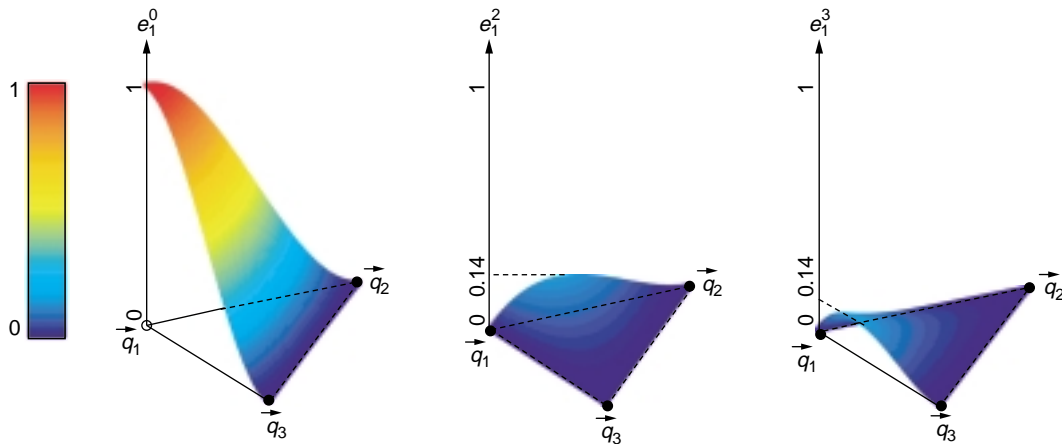$$\nabla_y e_i^0(\vec{q}_j) = 0 \qquad (35)$$

Figure 16

The three kinds of basis functions for Hermite interpolation in 2D for a reference triangle. In the first (left) the function has value one at the origin, zero at the other two vertices and, at all vertices, the derivatives along edges are zero. In the second (middle) the function has value zero at all vertices and, at all vertices, the derivatives along edges are zero except for edge $[\vec{q}_1, \vec{q}_2]$ at the origin, where this derivative has value one. In the third (right) the function has value zero at all vertices and, at all vertices, the derivatives along edges are zero except for edge $[\vec{q}_1, \vec{q}_3]$ at the origin, where this derivative has value one.

$e_i^x(\vec{q})$ by:

$$e_i^x\left(\vec{q}_j\right) = 0 \tag{36}$$

$$\nabla_x e_i^x\left(\vec{q}_j\right) = \delta_{ij} \tag{37}$$

$$\nabla_y e_i^x\left(\vec{q}_j\right) = 0 \tag{38}$$

and $e_i^y(\vec{q})$ by:

$$e_i^y\left(\vec{q}_j\right) = 0 \tag{39}$$

$$\nabla_x e_i^y\left(\vec{q}_j\right) = 0 \tag{40}$$

$$\nabla_y e_i^y\left(\vec{q}_j\right) = \delta_{ij} \tag{41}$$

The above equations must be fulfilled for all i and j with values $1, \ldots, I$. On each triangle, each basis function has to fulfill 9 conditions: on each triangle, the basis functions are to be sought within the space of third order polynomials (in fact, a subspace of space P^3 of third order polynomials since the definition of these polynomials requires the specification of 10 coefficients).

To exhibit the piece of basis function e_i^0 associated with a triangle which has $\vec{q}_i$ as summit we can equivalently impose that at all summits of the triangle, the derivatives along the edges are zero and exhibit this function by using a reference triangle *(Fig. 16, left)*. This function can be expressed in terms of barycentric coordinates (λ_k is the barycentric coordinate associated with vertex number k in the numbering associated with the triangle) in closed form:

$$e_i^0 = \lambda_1^2\left(3 - 2\lambda_1\right) + 2\lambda_1\lambda_2\lambda_3 \tag{42}$$

Instead of exhibiting basis functions e_i^x and e_i^y, we shall exhibit basis functions for which we impose, at the different

vertices, conditions on derivatives along edges instead of derivatives along the x or y directions (it is straightforward to obtain basis functions e_i^x and e_i^y once the new basis functions have been exhibited). Again making use of the reference triangle and of the node numbering attached to this triangle, the basis functions e_i^2 as described in Figure 16 (middle) can be expressed in terms of barycentric coordinates as:

$$e_i^2 = \frac{1}{2}\lambda_1\lambda_2\left(1 + \lambda_1 - \lambda_2\right) \tag{43}$$

whereas the basis functions e_i^3 as described in Figure 16 (right) can be expressed as:

$$e_i^3 = \frac{1}{2}\lambda_1\lambda_3\left(1 + \lambda_1 - \lambda_3\right) \tag{44}$$

It can be verified that the basis functions obtained by assembling the pieces described above have C^0 regularity (instead of C^1 regularity in the 1D situation).

4.3.2 Numerical Integration

Once again, numerical integration turns out to be a weighted summation of the different quantities that have been interpolated:

$$\int_\Omega f_{int}\left(x, y\right)\, \mathrm{d}x\, \mathrm{d}y = \sum_i w_i^0 f\left(\vec{q}_i\right) + \sum_i w_i^x \nabla_x f\left(\vec{q}_i\right)$$
$$+ \sum_i w_i^y \nabla_y f\left(\vec{q}_i\right) \tag{45}$$

with:

$$w_i^0 = \int_\Omega e_i^0(\vec{q})\,\mathrm{d}x\,\mathrm{d}y \tag{46}$$

$$w_i^x = \int_\Omega e_i^x(\vec{q})\,\mathrm{d}x\,\mathrm{d}y \tag{47}$$

$$w_i^y = \int_\Omega e_i^y(\vec{q})\,\mathrm{d}x\,\mathrm{d}y \tag{48}$$

4.3.3 Error Estimate

The derivation of the error estimate E_H involved in the numerical integration formula (45) is technical. In [8] (example V-2-2, page VI-23) we find the following result:

$$E_{H\le} \|f - f_{int}\|_{L^1(\Omega)} \le C_f^H h^3 \tag{49}$$

Again, h is the characteristic dimension of the triangles in the triangulation; constant C_f^H depends on the deformations of the triangles in the triangulation (the best triangles are equilateral triangles) and on the norm of function f in the Sobolev space $W^{3,1+\varepsilon}(\Omega)$ (ε is an arbitrary strictly positive number). Such a norm is large whenever the third order derivatives of function f are large: thus the Hermite interpolation quadrature formula is more accurate than the one based on Lagrange interpolation only for smooth functions f. Note that we loose one order in h as compared with the 1D situation *(Eq. (14))*: this is because the space of finite elements is no longer P^3 but just a subspace of P^3. However, this error estimate will turn out to be quite compatible with the errors that arise from a numerical evaluation of derivatives (see next section).

4.3.4 Numerical Differentiation

If we wish to use a quadrature formula based on Hermite interpolation with no direct measurements of the derivatives, we must evaluate these derivatives by some numerical procedure. Here we resume for the 2D situation what was previously done for the 1D situation, restricting ourselves to a specific order of accuracy, namely $n = 2$. The aim is again to obtain an estimation of the gradient of function $f(\vec{q})$ at a given interpolation node (which we place, by an appropriate translation of the origin, at $\vec{q} = \vec{0}$) of the form:

$$\vec{\nabla}f(\vec{q} = \vec{0}) = \sum_k \vec{\alpha}_k f(\vec{q}_k) + O(h^2) \tag{50}$$

where first, subscript k refers to a local (*i.e.* relative to the node placed at $\vec{q} = \vec{0}$ which corresponds to $k = 0$) numbering [5] of nodes (the global numbering is defined by subscript i); secondly, the summation on k takes place only

---

(5) This local node numbering must not be confused with the one attached to a specific triangle that we used previously.

---

at those interpolation nodes located at the vicinity of the one at $\vec{q} = \vec{0}$; and thirdly, the vectors $\vec{\alpha}_k$ (as well as the number of neighbor interpolation nodes that have to be involved in the summation) must be determined so as to yield an error estimate in $O(h^2)$.

We begin with the Taylor expansion at the vicinity of $\vec{q} = \vec{0}$:

$$f(\vec{q}_k) = f(\vec{0}) + \vec{\nabla}f(\vec{0}).\vec{q}_k + \frac{1}{2}\vec{q}_k.([D^2f(\vec{0})]\,\vec{q}_k) + O(h^3) \tag{51}$$

where . denotes the dot product in $\mathbb{R}^2$ and $[D^2f(\vec{0})]$ denotes the 2×2 matrix of second order derivatives of function f at $\vec{q} = \vec{0}$.

If the $\vec{\alpha}_k$'s in Equation (50) are chosen so as to satisfy:

$$\sum_k \vec{\alpha}_k = \vec{0} \tag{52}$$

$$\sum_k (\vec{q}_k.\vec{u})\,\vec{\alpha}_k = \vec{u} \quad \forall \vec{u} \in \mathbb{R}^2 \tag{53}$$

$$\sum_k (\vec{q}_k.([A]\,\vec{q}_k))\,\vec{\alpha}_k = \vec{0} \tag{54}$$

$\forall [A]$, a 2×2 symmetric matrix, then we obtain a second order accurate estimate (keep in mind the scaling based argument given in Appendix C) of $\vec{\nabla}f(\vec{q} = \vec{0})$ given by Equation (50).

The total number of conditions to be fulfilled by the $\vec{\alpha}_k$'s is 12: 2 are given by Equation (52), 4 by Equation (53), and 6 by Equation (54). Hence the $\vec{\alpha}_k$'s can be computed by solving the linear system (52)–(54) involving the values of function f at $\vec{q} = \vec{0}$ and at 5 properly chosen (so that the linear system is not singular) neighbor interpolation nodes with locations $\vec{q}_k$ (typical situations are shown in Figure 17).

This technique can be applied for each interpolation node: if $\mathbf{f}$ denotes the vector whose components are the values $f(\vec{q}_i)$, $i = 1, \ldots, I$ (subscript i refers to the global numbering of interpolation nodes), we can construct, from this vector, the vector $\vec{\nabla}_h \mathbf{f}$ (here again, subscript h refers to the characteristic dimension of the triangles) whose components are estimates of the gradient of f evaluated at the different interpolation nodes $\vec{q}_i$, $i = 1, \ldots, I$. Since the evaluation of this gradient involves a linear process *(see Eq. (50))*, the construction of this vector is nothing but the application of a linear operator (a $I \times I$ matrix) denoted $\vec{\nabla}_h$ to vector $\mathbf{f}$. The technique described above explains how to construct operator $\vec{\nabla}_h$.

It is straightforward to use this numerical differentiation technique together with the numerical integration formula based on Hermite interpolation: instead of Equation (45) we obtain:

$$\int_\Omega f_{int}(x, y)\,\mathrm{d}x\,\mathrm{d}y \simeq \mathbf{w}^0.\mathbf{f} + \vec{\mathbf{w}}.\vec{\nabla}_h \mathbf{f} \tag{55}$$

where $\mathbf{w}^0$ is the vector that gathers the weights w_i^0 defined in (46) (the first dot product involves a summation over
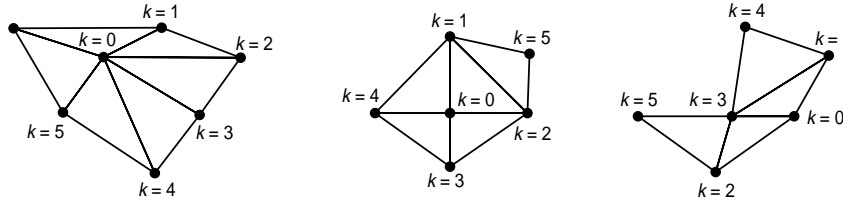
Figure 17

Typical situations for the choice of 5 neighboring interpolation nodes, with locations $\vec{q}_k$, in the vicinity of the interpolation node $\vec{q}_0$, for the numerical differentiation procedure. Left: a simple choice of 5 neighbors $\vec{q}_k$ among those associated with the 6 neighbors of $\vec{q}_0$. Middle: necessity to find one summit, $\vec{q}_5$, in a neighboring triangle. Right: as $\vec{q}_0$ is on the boundary of the integration domain, it appears necessary to find two other summits, $\vec{q}_4$ and $\vec{q}_5$, in two neighboring triangles.

---

subscript i), and $\vec{\mathbf{w}}$ is the vector (with the same dimensions as $\vec{\nabla}_h \mathbf{f}$) that gathers the weights w_i^x and w_i^y defined in (47) and (48), respectively (the second dot product involves a summation over all components of the vectors considered, namely over subscript i and over the x and y components).

As shown in Appendix D, the error made when using numerical integration formula (55) is in $O\left(h^3\right)$.

4.4 Application to Kirchhoff Migration (Diffraction Stack)

We shall now apply these interpolation procedures to our concrete problem, *i.e.* the numerical integration for Kirchhoff migration.

For given $\mathbf{M}$ and $\vec{h}$, the Lagrange interpolation function $\tilde{m}_{el}^v(\mathbf{M}; \vec{q}, \vec{h})$, that interpolates the different $m_{el}^v(\mathbf{M}; \vec{q}_i, \vec{h})$ defined by Equation (2) for $i = 1, \ldots, I$, can be written, according to Equation (24), as:

$$\tilde{m}_{el}^v(\mathbf{M}; \vec{q}, \vec{h}) = \sum_i m_{el}^v(\mathbf{M}; \vec{q}_i, \vec{h}) e_i(\vec{q}) \qquad (56)$$

Equation (26) then provides the migrated image $\tilde{m}_h^v$ at a subsurface point $\mathbf{M}$ as a weighted summation of the values of m_{el}^v at midpoints $\vec{q}_i$ of the acquisition:

$$\tilde{m}_h^v(\mathbf{M}) = \int_\Omega \tilde{m}_{el}^v(\mathbf{M}; \vec{q}, \vec{h}) \, \mathrm{d}x \, \mathrm{d}y = \sum_i w_i m_{el}^v(\mathbf{M}; \vec{q}_i, \vec{h})$$
$$(57)$$

with w_i defined by (27).

Note that weights w_i do not depend on the point $\mathbf{M}$ where the image is to be computed. Note also that, by linearity of the migration operator, $w_i m_{el}^v(\mathbf{M}; \vec{q}_i, \vec{h})$ is the elementary migrated image associated with the seismic trace $w_i \, \mathrm{d}\left(\vec{q}_i, \vec{h}, t_v(\mathbf{M}; \vec{q}_i, \vec{h})\right)$.

Thus, for a given offset gather, we can compute once and for all the weights w_i associated with each midpoint $\vec{q}_i$ and scale the corresponding seismic trace by multiplying it by w_i. The use of the simple numerical diffraction stack (4) with

scaled seismic traces yields the same result as the numerical integration formula (57): this simple preprocessing makes the mere numerical diffraction stack a consistent approximation of the continuous diffraction stack.

The same kind of procedure also applies to Hermite interpolation. The Hermite interpolation function $\tilde{m}_{el}^v$ can indeed be written, accordingly to equation (32), as:

$$\begin{aligned}
\tilde{m}_{el}^v(\mathbf{M}; \vec{q}, \vec{h}) = & \sum_i m_{el}^v(\mathbf{M}; \vec{q}_i, \vec{h}) e_i^0(\vec{q}) \\
& + \sum_i \nabla_x m_{el}^v(\mathbf{M}; \vec{q}_i, \vec{h}) e_i^x(\vec{q}) \qquad (58) \\
& + \sum_i \nabla_y m_{el}^v(\mathbf{M}; \vec{q}_i, \vec{h}) e_i^y(\vec{q})
\end{aligned}$$

where the basis functions e_i^0, e_i^x, e_i^y are defined by Equations (33) to (41).

If the derivatives are evaluated by the numerical differentiation procedure described above, the corresponding migrated image $\tilde{m}_h^v$ at a subsurface point $\mathbf{M}$ is defined, according to Equation (55), by:

$$\tilde{m}_h^v(\mathbf{M}) = \mathbf{w}^0 . \mathbf{m}_{el}^v(\mathbf{M}) + \vec{\mathbf{w}} . \vec{\nabla}_h \mathbf{m}_{el}^v(\mathbf{M}) \qquad (59)$$

where $\mathbf{w}^0$, $\vec{\mathbf{w}}$ and $\vec{\nabla}_h$ are defined in (55) and $\mathbf{m}_{el}^v(\mathbf{M})$ denotes the vector whose components are the values $m_{el}^v(\mathbf{M}; \vec{q}_i, \vec{h})$ of the elementary migrated images.

Introducing $\vec{\nabla}_h^T$, the transposed operator of $\vec{\nabla}_h$, the quadrature formula can be rewritten as:

$$\tilde{m}_h^v(\mathbf{M}) = \left(\mathbf{w}^0 + \vec{\nabla}_h^T \vec{\mathbf{w}}\right) . \mathbf{m}_{el}^v(\mathbf{M}) \qquad (60)$$

This formulation is interesting since vector $\mathbf{w}^0 + \vec{\nabla}_h^T \vec{\mathbf{w}}$ gives the weights to be applied to each elementary migrated image $m_{el}^v(\mathbf{M}; \vec{q}_i, \vec{h})$. As in the Lagrange interpolation, these weights are independent on the subsurface point $\mathbf{M}$, and a preprocessing involving a scaling of each seismic trace in the common offset gather by the corresponding weight makes the mere numerical diffraction stack a consistent and very
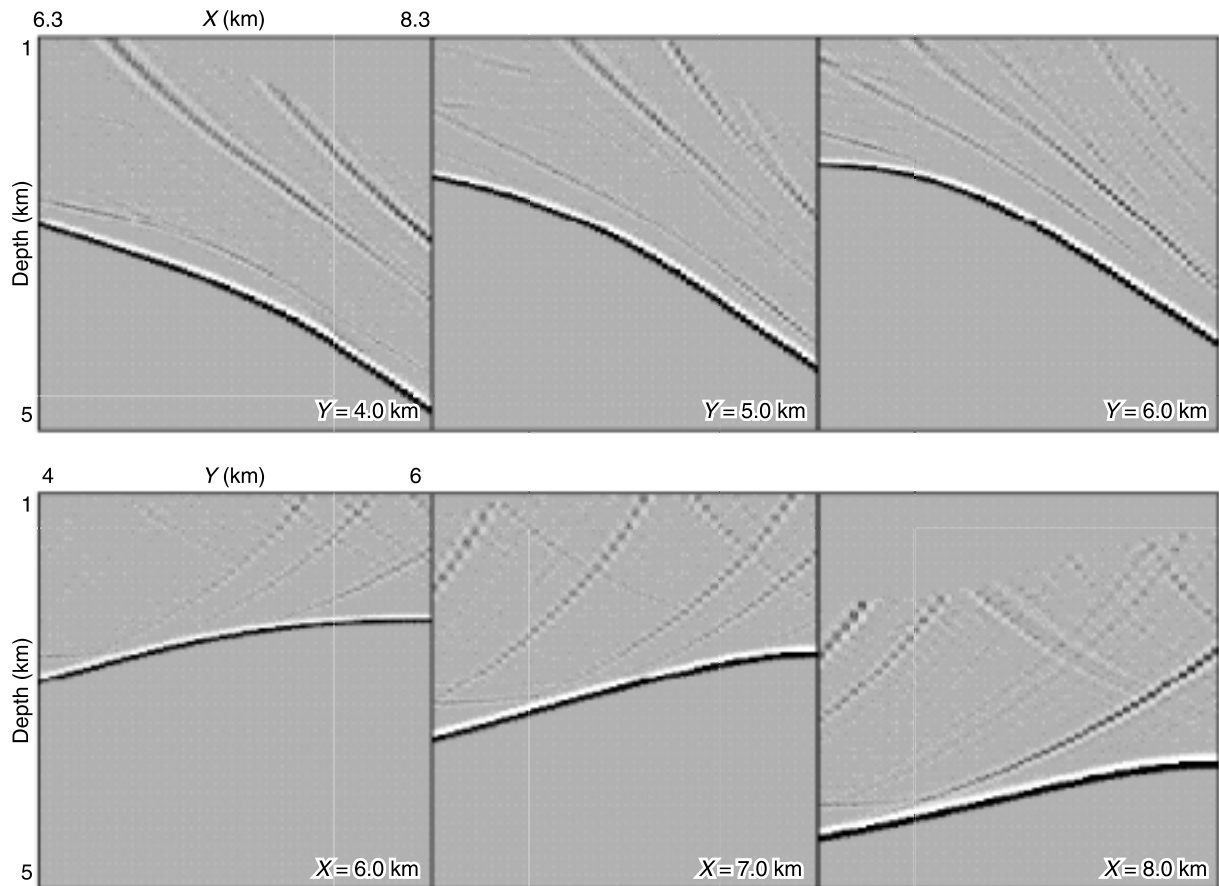
Figure 18

Migration using a Lagrange numerical integration formula, of small offset (0.2 km) data associated with an actual marine acquisition geometry *(Fig. 2)*. Constant *Y* sections (top) and constant *X* sections (bottom) in the depth migrated data volume obtained with the correct migration velocity (*v* = 3.0 km/s).

accurate (error in $O(h^3)$) approximation of the continuous diffraction stack.

In conclusion, our numerical integration formulas based on Lagrange and Hermite interpolation can be very easily implemented in any Kirchhoff migration program involving a superposition of elementary migrated images. The additional cost is a scaling of the seismic traces, which is quite negligible compared with the cost of running the migration itself.

5 NUMERICAL RESULTS

5.1 Application to the Test Example

We shall now test the diffraction stack based on the Lagrange and Hermite interpolation schemes on the previously studied synthetic data built using a real marine acquisition geometry.

Using the Lagrange numerical integration formula, the corresponding migrated data associated with a small offset

(0.2 km) are shown in Figure 18. A substantial improvement is observed in the quality[6] of the migrated images as compared with those obtained with the mere diffraction stack formula (4) *(Fig. 2)*: a considerable number of migration artefacts has been removed. The same kind of result is obtained for the large offset (3 km) (Figure 20, to be compared with Figure 3). For the large offset, however the number of remaining artefacts is much larger than for the smaller offset. The remaining artefacts are caused by local sparsities in the midpoint distribution. In such a situation, as explained in the Section 2, we do not have enough samples (values taken by the elementary migrated images at the image point considered) to correctly represent the function to be integrated. This situation is

---

(6) Note that image quality is essentially evaluated here by a visual criterion. It thus depends on the graphical representation mode selected. In all the experiments, the scaling factor used for the displays is the same so as to allow comparison between results.
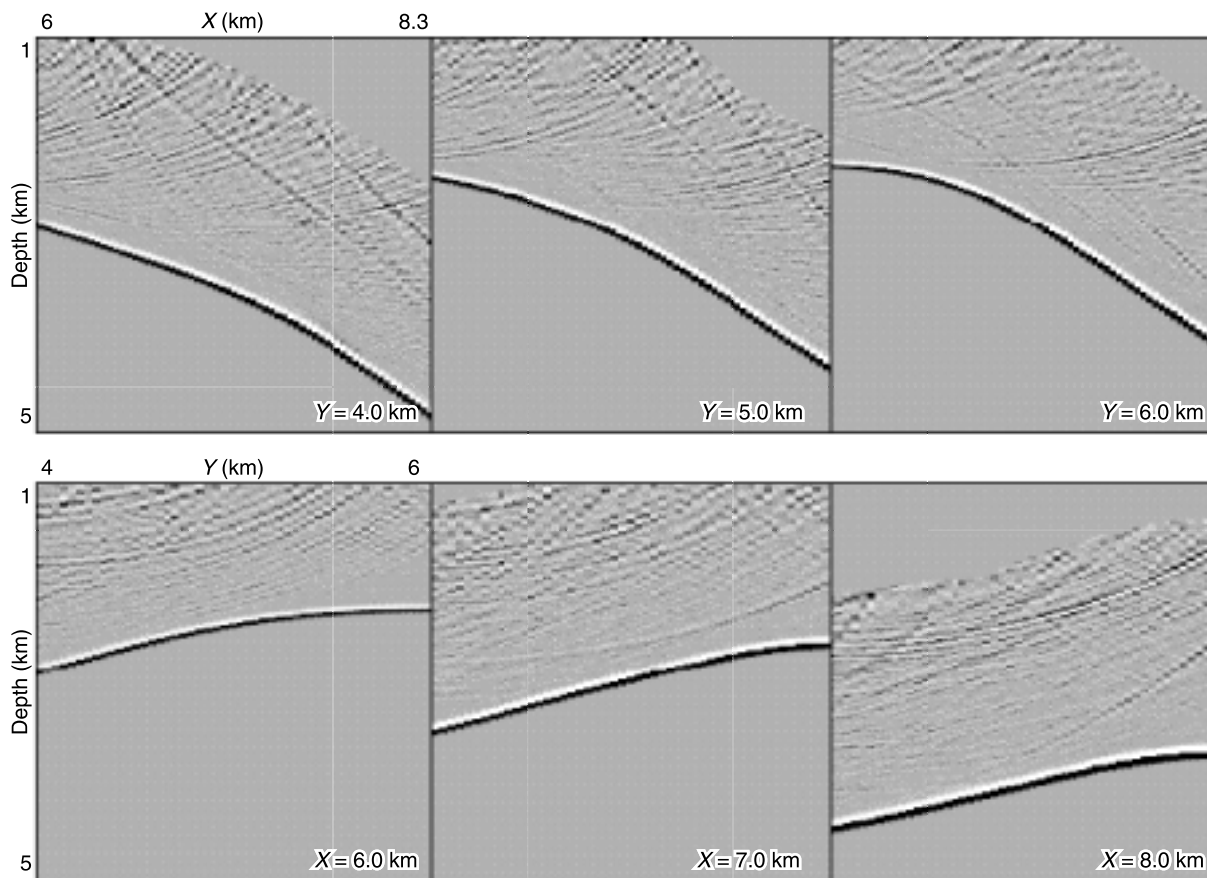
Figure 19

Migration using a Hermite numerical integration formula, of small offset (0.2 km) data associated with an actual marine acquisition geometry *(Fig. 2)*. Constant *Y* sections (top) and constant *X* sections (bottom) in the depth migrated data volume obtained with the correct migration velocity ($v = 3.0$ km/s).

frequently encountered when migrating data associated with large offsets.

Using the Hermite interpolation scheme (which is, theoreretically, more accurate than the Lagrange scheme), the corresponding migrated data associated with a small offset (0.2 km) and with a large offset (3 km) are shown in Figures 19 and 21, respectively. The result is apparently degraded by a kind of random noise as compared with the result obtained by Lagrange interpolation.

In order to understand this surprising result we shall carry out a similar experiment with a survey involving a finer sampling in midpoints (actually the same data than those used in Figure 5). This finer sampling was obtained by adding midpoints to the original acquisition geometry. These additional midpoints are located at the barycenter of each triangle in the triangulation built from the original acquisition *(Fig. 22)* and the added traces correspond to offset such that $\|\vec{h}\| = 3$ km with azimuth equal to zero. With this finer sampling interval, our numerical integration formulas can be expected to be more accurate.

The results obtained by means of Hermite interpolation are shown in Figure 23: despite irregularities in the midpoint distribution, they appear as perfect (these results should be compared with those in Figure 5 obtained by migrating the same data but with the mere diffraction stack formula (4)). Both theory and software thus seem to be correct. These good results may serve as a reference (*i.e.* be considered as correct) and therefore be used to evaluate the error made when computing the results with the original acquisition. The errors *(Figs. 24, 25 and 26)* evaluated for offset 3 km (which gave rise to the strongest artefacts) are the differences between the reference result *(Fig. 23)* and the results in Figures 3, 20 and 21, respectively.

Comparing the observed errors we note that:

– the use of a consistent numerical integration formula drastically improves the quality of the migrated images as compared with those obtained with the mere diffraction stack;

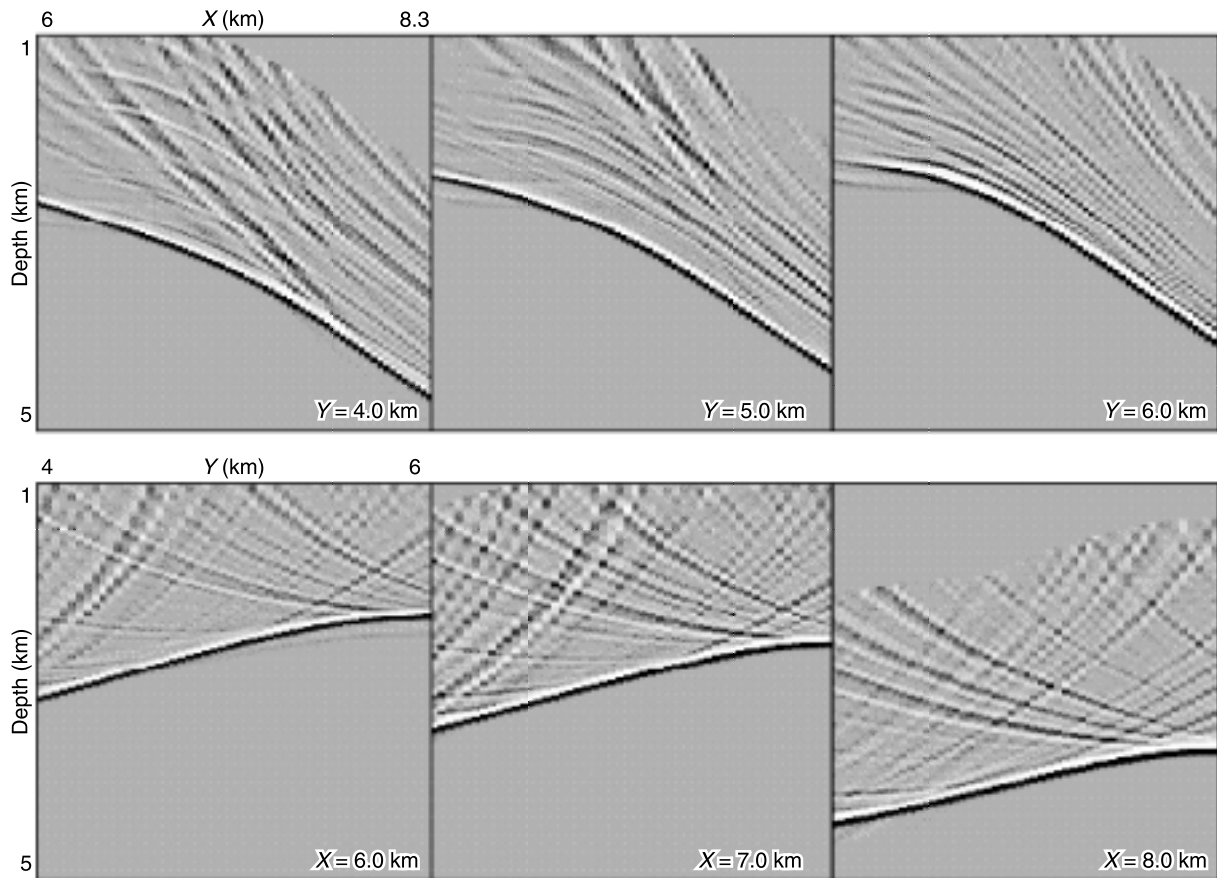– in the vicinity of the reflector, results obtained by means

Figure 20

Migration using a Lagrange numerical integration formula, of large offset (3 km) data associated with an actual marine acquisition geometry *(Fig. 3)*. Constant *Y* sections (top) and constant *X* sections (bottom) in the depth migrated data volume obtained with the correct migration velocity (*v* = 3.0 km/s).

of Hermite interpolation are more accurate than those obtained by Lagrange interpolation;

– at some distance above the reflector, the use of Hermite interpolation reveals many more migration smiles, hence the random noise observed in Figures 19 and 21.

These experimental results are well explained by the theory:

– the Hermite based integration formula can be expected to be more accurate than the formula based on Lagrange interpolation when the function to be integrated is smooth, that is, as shown in Figure 6, at the vicinity of the reflector;

– however, if the function to be integrated is rough (typically the situation shown in Figure 7), that is, at some distance above the reflector, the Lagrange based integration formula turns out to be more robust.

This relative robustness, however, is still not sufficient to achieve a correct numerical integration when we move far above the reflector: the midpoint distribution is then not

dense enough to make the samples correctly represent such a rapidly varying function.

5.2 Application to a Real Data Set

We shall now compare the behavior of the mere diffraction stack and the diffraction stack based on the Lagrange interpolation scheme on a real data set. The Lagrange interpolation scheme was chosen because this technique appeared in the previous section as the most robust.

The real data set came from a marine acquisition in the North Sea. This data set covers a narrow salt body (about 3 km wide). The data acquisition system was a single-source double-streamer. In theory, the spacing between both the streamers and the navigation lines was 75 m. The spacing between both the shots and the receivers was 25 m. From the prestack data, we built a cube of near offset data (offset range [80 m, 100 m]). Figure 27 shows the midpoint distribution of a part of the acquisition and the triangulation that was used for the Lagrange interpolation scheme (again the midpoints
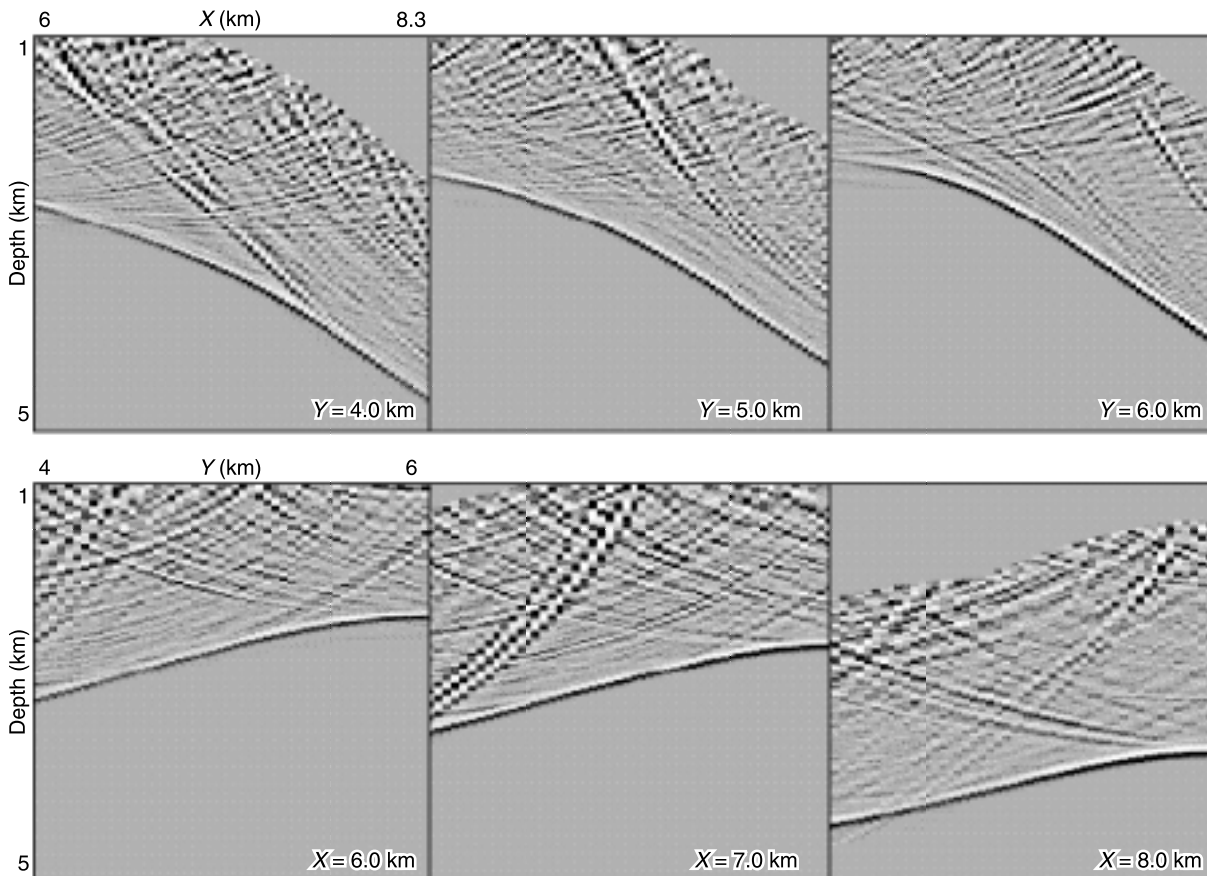
Figure 21

Migration using a Hermite numerical integration formula, of large offset (3 km) data associated with an actual marine acquisition geometry *(Fig. 3)*. Constant *Y* sections (top) and constant *X* sections (bottom) in the depth migrated data volume obtained with the correct migration velocity (*v* = 3.0 km/s).
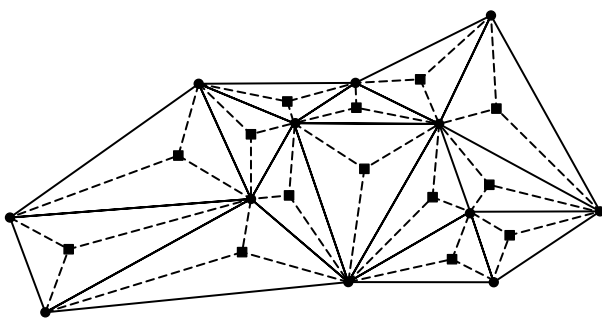

Figure 22

Densification of midpoints in an acquisition. The midpoints in the original acquisition are shown by circles: they are the vertices of the triangles in the original triangulation. New midpoints (squares) are added to the original acquisition as the barycenters of the triangles from the original triangulation. They serve to refine the original triangulation: the refined triangulation is obtained by splitting each triangle in the original triangulation into three pieces.

are the vertices of the triangles). We can clearly observe the non uniform distribution of midpoints in the acquisition. This figure also clearly shows the navigation direction and the ordinate of obstacles met by the ship.

We applied a 3D prestack common offset time migration software to the cube of near offset data. This software carries out a mere diffraction stack but takes account of the true locations of the source-receiver pairs. The migration velocity model is a 1D model with constant gradient. Although this model is quite simple, it provides a reasonable approximation of the velocity distribution in the sediments surrounding the salt body. A cross-line section in the cube of migrated data is shown in Figure 28. Many migration smiles can be observed. This is not surprising given the irregularity in the midpoint distribution. Note that the migration smiles could also have been caused by inaccuracies in our migration velocity model. However the effect of these inaccuracies should not be important in the shallower part of the migrated image.
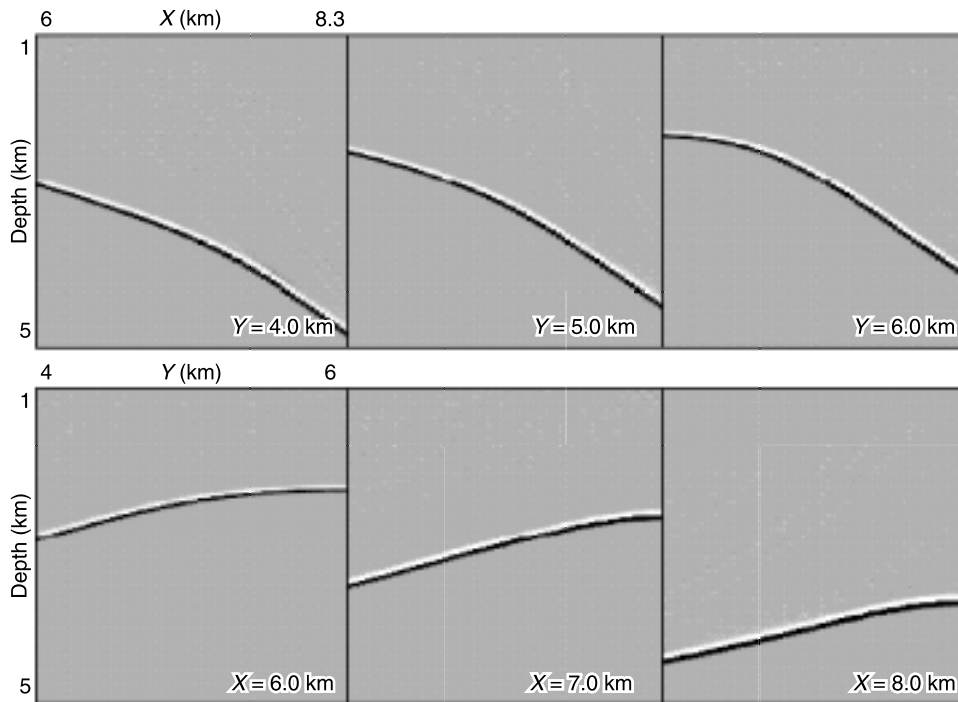
Figure 23

Migration using a Hermite numerical integration formula, of large offset (3 km) data associated with the refined actual marine acquisition geometry. Constant *Y* sections (top) and constant *X* sections (bottom) in the depth migrated data volume obtained with the correct migration velocity (v = 3.0 km/s).
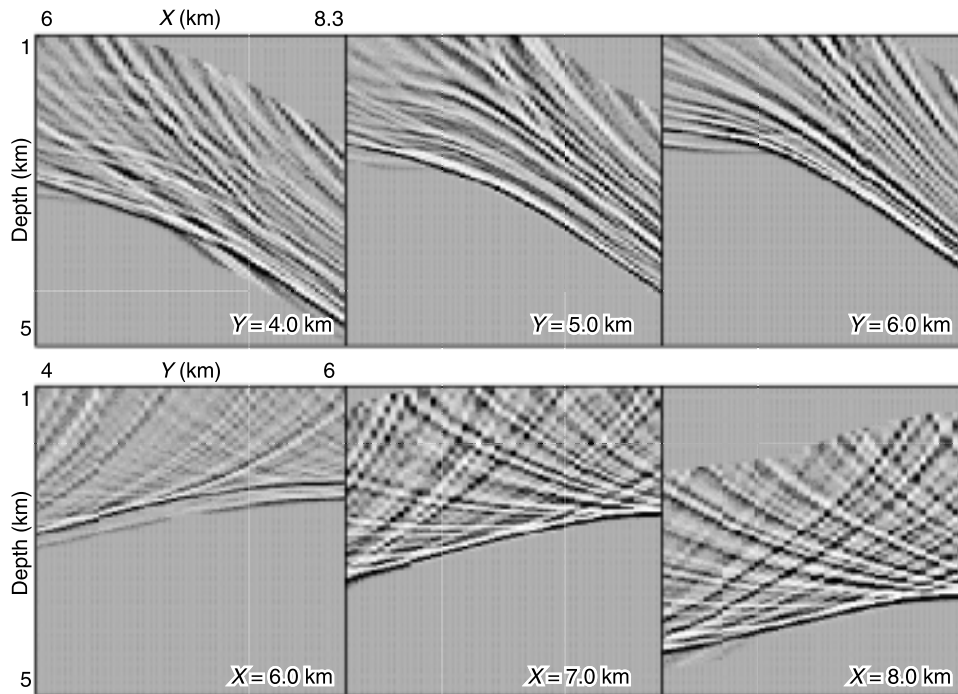

Figure 24

Evaluation of the error involved using the mere diffraction stack formula (4): this error was evaluated by computing the difference between the results shown in Figure 3 (obtained by the mere diffraction stack) and those shown in Figure 23 (used as reference). Constant *Y* sections (top) and constant *X* sections (bottom).
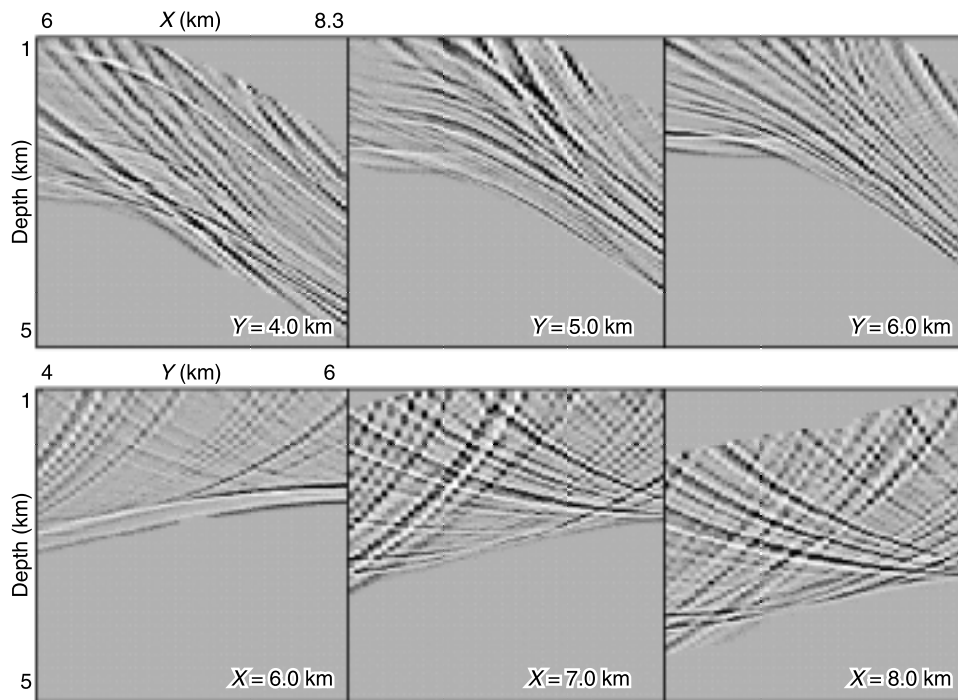
Figure 25

Evaluation of the error involved in the migration scheme based on Lagrange interpolation: this error was evaluated by computing the difference between the results shown in Figure 20 (obtained by numerical integration) and those shown in Figure 23 (used as reference). Constant *Y* sections (top) and constant *X* sections (bottom).
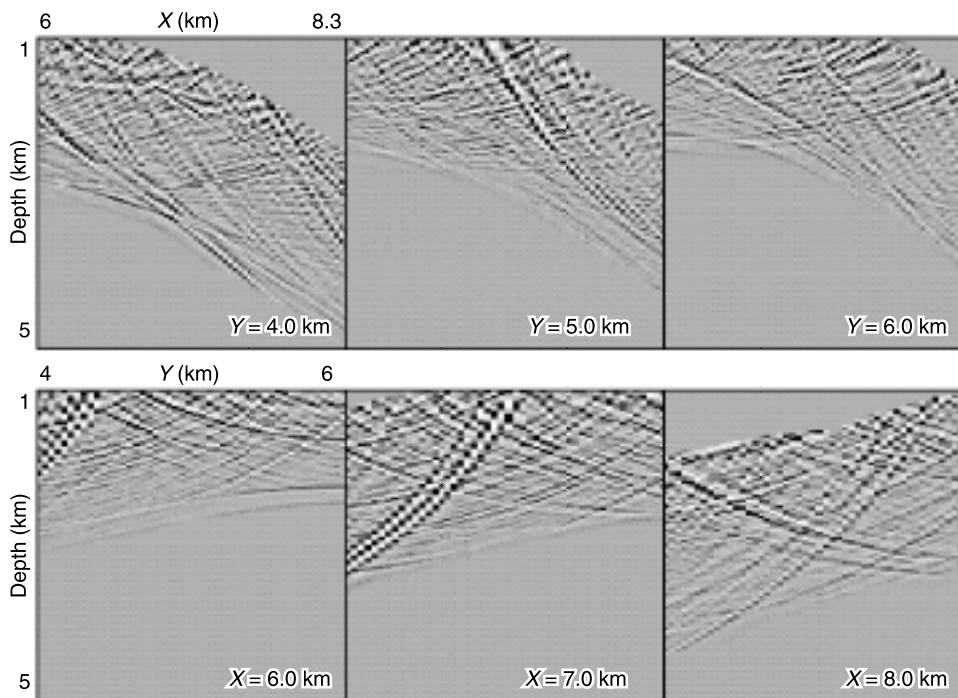
Figure 26

Evaluation of the error involved in the migration scheme based on Hermite interpolation: this error was evaluated by computing the difference between the results shown in Figure 21 (obtained by numerical integration) and those shown in Figure 23 (used as reference). Constant *Y* sections (top) and constant *X* sections (bottom).
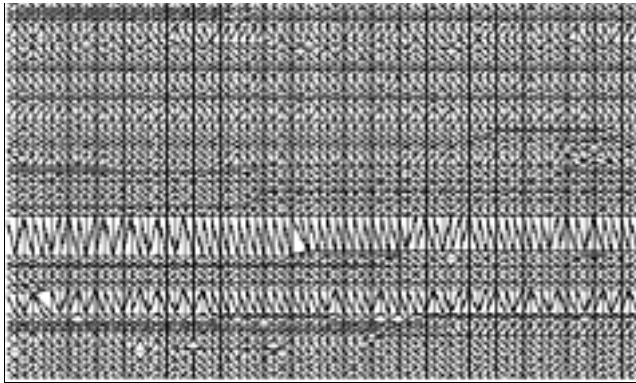
Figure 27

Part of the triangulation of another marine acquisition from the North Sea.

Thus we focus our interest on this shallower part namely in depths less than 1 second two-way travel-time. Figure 29 shows the result obtained by applying the same migration software to the data after they have been scaled (preprocessing associated with the Lagrange interpolation scheme) so as to compensate for irregularities in the midpoint distribution. This figure shows that the migration smiles, instead of disappearing, appear to be strengthened: in fact, large triangles have been generated in those areas where the midpoint distribution is sparse (*i.e.* at the vicinity of acquisition holes), hence giving huge weights to the seismic traces associated with the vertices of these triangles *(Figs. 30 and 31)*

To overcome this difficulty, we set an upper bound, namely 1800, to the weight values. In doing so, the migration result *(Fig. 32, top)* appears somewhat clearer but basically
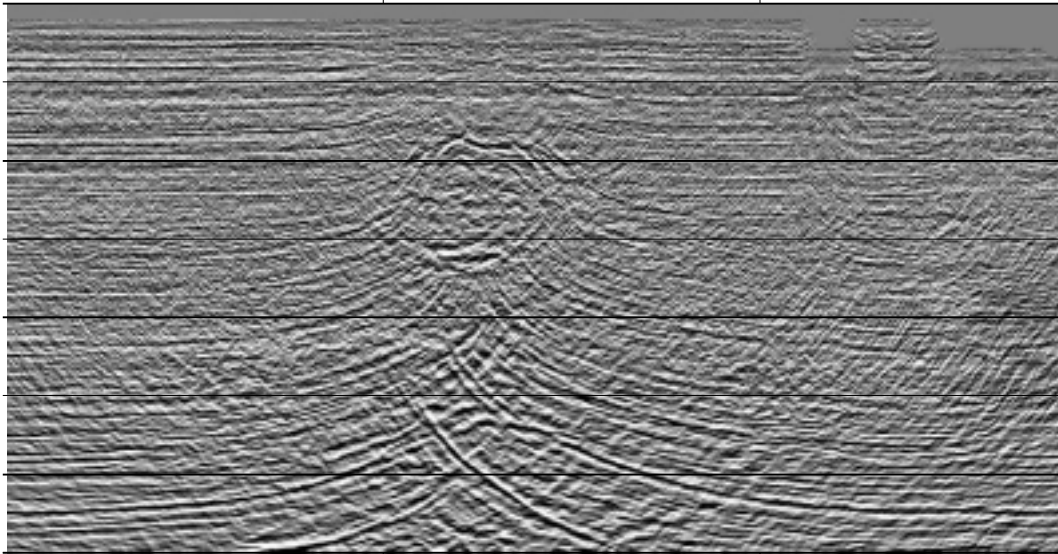

Figure 28

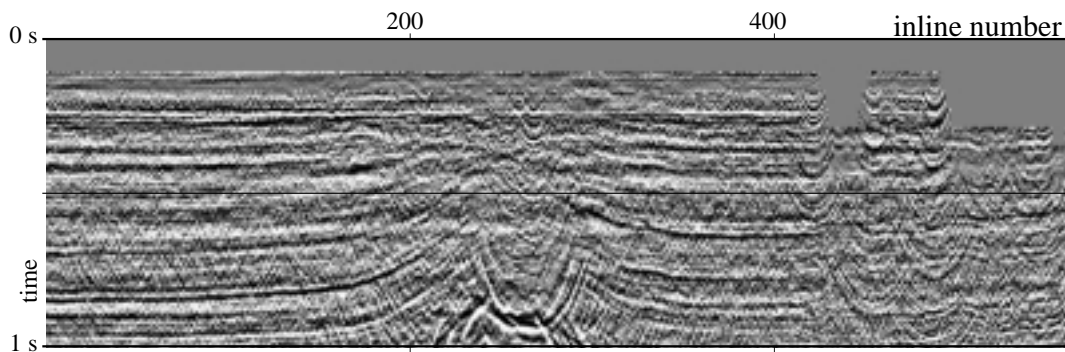Cross-line section in the cube of real data migrated using the mere diffraction stack.

Figure 29

Cross-line section (the same as in Figure 28) in the cube of real data migrated using the scheme based on Lagrange interpolation. The migration smiles appear strengthened compared with the result in Figure 28.
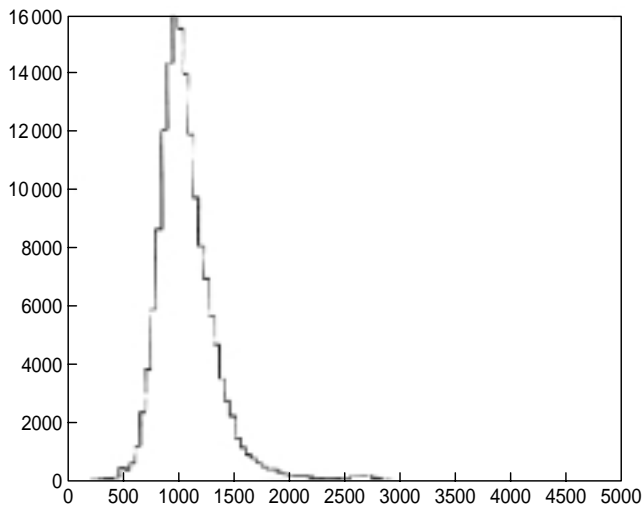
Figure 30

Histogram of the weight distribution (output of the Lagrange interpolation scheme) used to scale the traces in order to compensate for irregularities in the midpoint distribution. Large weight values are associated with midpoints surrounded by large triangles.

quantitative comparison). The difference basically consists, as expected, of migration smiles caused by the irregular midpoint distribution. On average, the amplitude of the difference turns out to be ten times smaller than the amplitude of the individual results: we can thus understand why the results obtained using the mere diffraction stack and those based on a consistent numerical formula yield appeared so similar!

In conclusion, the application of a consistent numerical integration formula to carry out a Kirchhoff migration does not drastically improve the quality of migrated images in practice. Moreover, this approach cannot compensate for large acquisition holes: yet, the weights have to be truncated to avoid spurious migration smiles arising from those areas where the midpoint distribution is sparse. However, once this upper bound has been properly chosen, the method can be applied at negligible additional cost to yield a result at least more satisfactory from a quantitative standpoint. Hence the main context for an application of this approach is quantitative migration (see *e.g.* [10]) for reservoir characterization.

CONCLUSION

quite similar to the one obtained using the mere (unweighted) diffraction stack *(Fig. 28)*. Figure 32 bottom shows the differences between the two results (after scaling to allow

Theoretically speaking, Kirchhoff migration cannot be performed with a mere discrete diffraction stack since, for irregularly sampled data, *i.e.* for a non uniform midpoint distribution in an acquisition, it yields an inconsistent approximation
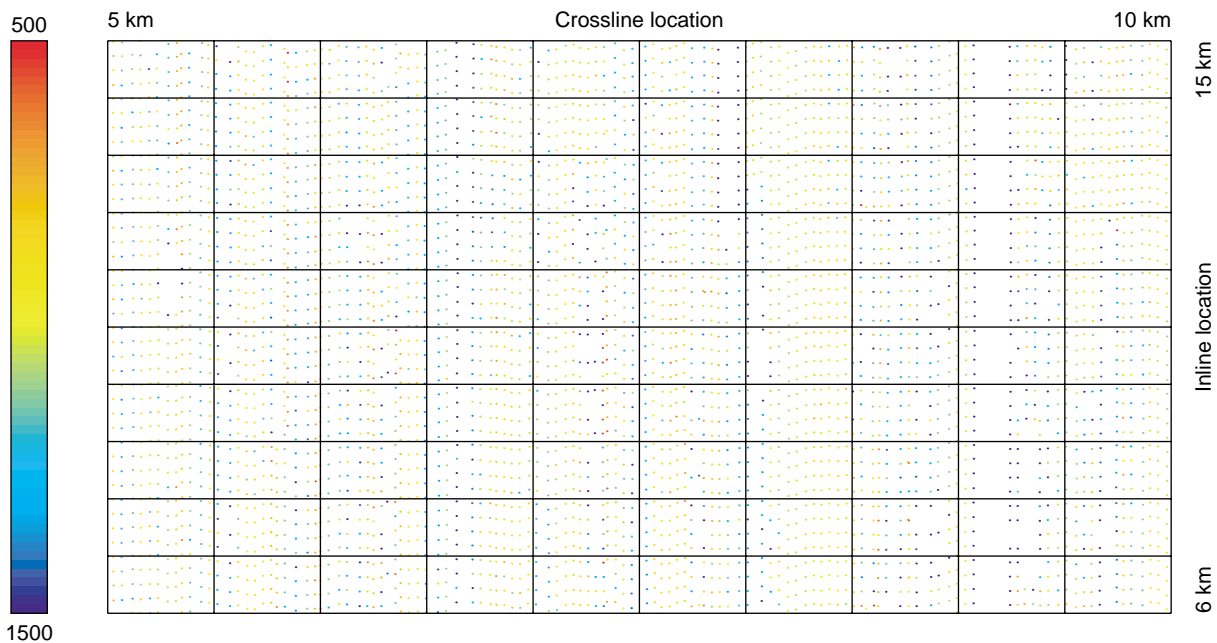

Figure 31

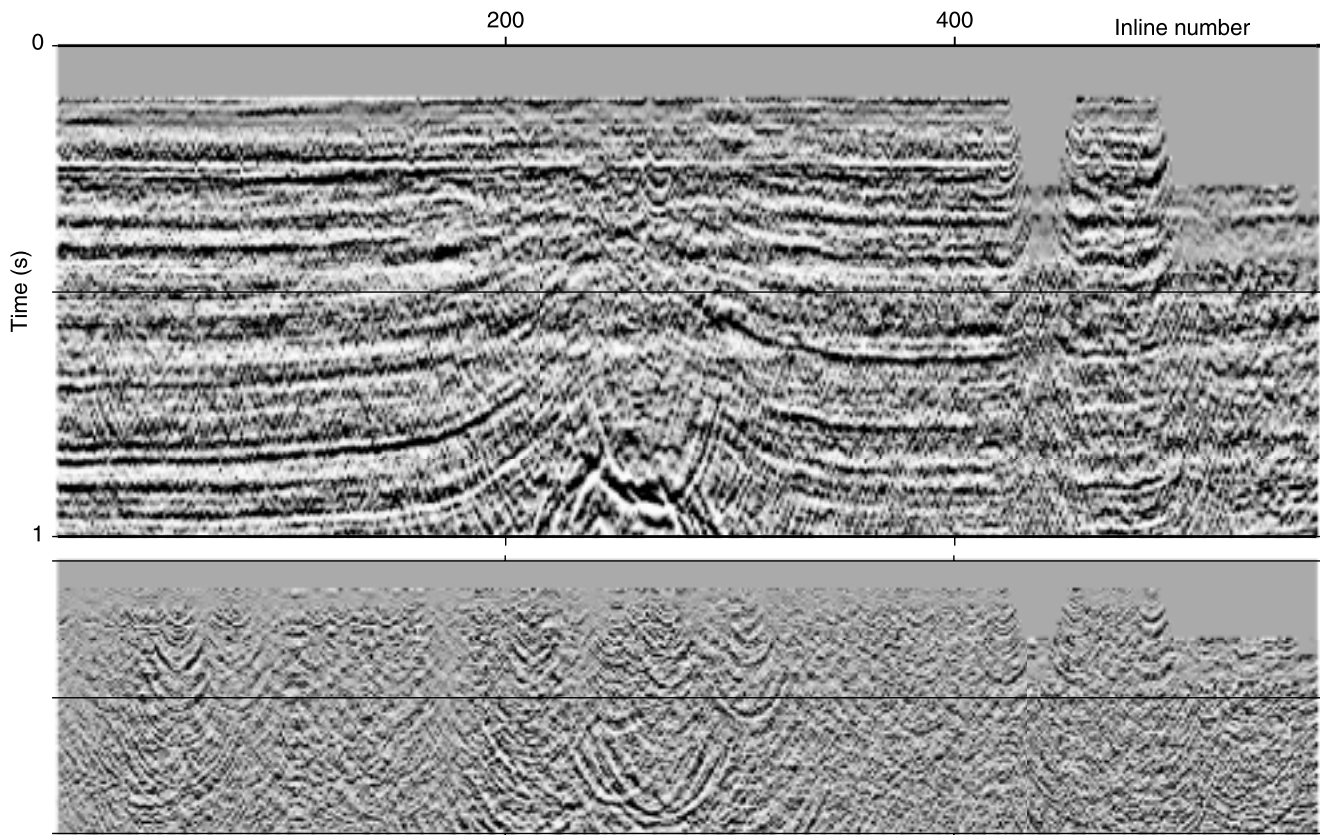Color plot of weight values as a function of midpoint location (zoom).

Figure 32

Top: same as the left part of Figure 29 but with weight values truncated above 1800.
Bottom: difference between the results in Figures 32 top and 28, after normalization of these results.

of the continuous version of the diffraction stack. In fact, using a mere discrete diffraction stack gives a result which is a superposition of elementary migrated events (migration smiles) and this superposition does not give rise to the construction of the migrated event (the envelope of the elementary migrated events). To carry out a migration with non uniformly distributed midpoints, a consistent approximation of the continuous diffraction stack must be used. This can be done by implementing a numerical integration based on interpolation procedures. Two interpolation schemes have been examined: first order Lagrange interpolation and third order Hermite interpolation.

The implementation of these schemes is straightforward: we can use any available code computing a mere discrete diffraction stack after a simple scaling of the input seismic traces, the scaling factor being determined by the selected (Lagrange or Hermite) numerical integration technique. In turn, since this scaling is quite cheap, using the proposed schemes does not really increase the CPU time required to run the migration.

We have shown that, when the correct migration velocity is used, the results obtained with either scheme are very good, provided the midpoint distribution is dense enough to account correctly for the variations of the function to be integrated. This also holds for an erroneous migration velocity.

The technique we have investigated is designed to compensate for irregularities in the midpoint distribution. The difficulty raised by the sparsity of midpoints is another problem. More specifically, variations in the function to be integrated can be extremely rapid and it is therefore quite common, especially when shallow dipping events are to be imaged, that the distance between midpoints is too large to allow correct imaging: the image is then contaminated by migration smiles. This difficulty is often referred to as the aliasing of the migration operator. A number of filters have been designed to remove these artefacts (for instance [4] or [5]) and implemented in industrial migration programs. The counterpart is an unavoidable degradation of the spatial resolution. Since the schemes we propose amount to a data preprocessing, they can, and must, be used in combination with these filters.

From the practical standpoint, the numerical integration formula based on Lagrange interpolation is more robust than

the formula based on Hermite interpolation. An upper bound must also be set for the weights scaling the input seismic traces: lack of care at this stage can be harmful. The improvement provided by the method is in fact slight: its main field of application is for the quantitative determination of reservoir heterogeneities.

Of course, if we are interested in true amplitude imaging, a Kirchhoff migration based on a diffraction stack turns out to be simplistic, and dedicated techniques (see *e.g.* [10]) must be used. These techniques, obviously, also require numerical integration when applied to real, discrete data. The ones presented in this paper also apply to these more sophisticated imaging techniques, provided the function to be integrated is modified accordingly.

Finally, we have concentrated on common offset depth migration, but the techniques proposed are also applicable to time migration or when imaging is carried out in another domain: shot record domain, common angle domain ([11] for instance), etc.

ACKNOWLEDGEMENTS

REFERENCES

1 Assouline, F. and Lailly, P. (2001) Influence of Data Irregularities on 3D Common offset Migration. *Oil & Gaz Science and Technology – Rev. IFP*, **56**, 407–430.

2 Jurado, F., Sinoquet, D. and Ehinger, A. (1996) 3D Reflection Tomography Designed for Complex Structures. *Expanded Abstracts*, *Soc. Expl. Geophys.*, 711–714.

3 Bleistein, N. (1987) On the Imaging of Reflectors in the Earth. *Geophysics*, **52**, 931–942.

4 Lumley, D.E., Claerbout, J. F. and Bevc, D. (1994) Anti-Aliased Kirchhoff 3D Migration. *Expanded Abstracts*, *Soc. Expl. Geophys.*, 1282–1285.

5 Gray, S.H. (1992) Frequency-Selective Design of the Kirchhoff Migration Operator. *Geophys. Prospect.*, **40**, 565–571.

6 Bellman, R., Kashef, B.G. and Casti, J. (1972) Differential Quadrature: A Technique for the Rapid Solution of Nonlinear Partial Differential Equations. *J. Comput. Phys.*, **10**, 40–52.

7 Raviart, P.A. and Thomas, J.M. (1988) *Introduction à l'analyse numérique des équations aux dérivées partielles*, Masson.

8 Raviart, P.A. (1971) *Méthode des éléments finis*, Cours de DEA d'analyse numérique, université Paris VI.

9 Zienkiewicz, O.C. (1971) *The Finite Element Method in Engineering Science*, Mc Graw-Hill.

10 Schleicher, J., Tygel, M. and Hubral, P. (1993) 3D True-Amplitude Finite-Offset Migration. *Geophysics*, **58**, 1112–1126.

11 Xu, S., Chauris, H., Lambaré, G. and Noble, M. (1998) Common Angle Image Gather: A Strategy for Imaging Complex Media. 68th *Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts*, **2**, 1538–1541.

12 Engels, H. (1980) *Numerical Quadrature and Cubature*, Academic Press, Inc.

APPENDIXES

A ERROR ESTIMATE OF LAGRANGE INTERPOLATION IN 1D

We use the P^1 finite element technique to interpolate a function $f(x)$ on $[a, b]$. We have discretized interval $[a, b]$ by introducing some nodes with abscissas x_i, for $i = 1, \ldots, I$, these nodes being such that $x_1 = a$ and $x_I = b$. The nodes may be irregularly distributed on $[a, b]$. For an arbitrary function $f(x) \in C^2([a, b])$, we consider the function $f_{int}(x)$ obtained by a P^1 finite element interpolation technique. To estimate the error involved in this interpolation, we present a classical proof (more precisely an adaptation of a very general proof (Engels, 1980, for instance) to our simple interpolation procedure) based on a study of the error function $\varepsilon(x)$ defined by:

$$\varepsilon(x) = f(x) - f_{int}(x) \quad x \in [a, b] \qquad (A1)$$

On interval $[x_i, x_{i+1}]$ the error function vanishes for $x = x_i$ and $x = x_{i+1}$. On interval $[x_i, x_{i+1}]$ we introduce an arbitrary additional node with abscissa $x = u$ (u given arbitrary but different from x_i and x_{i+1}) and construct a higher order Lagrange interpolated function $f_{int}^1(x)$ defined by:

$$f_{int}^1(x) = f_{int}(x) + \frac{w(x)}{w(u)}\big(f(u) - f_{int}(u)\big) \qquad (A2)$$

where:

$$w(x) = (x - x_i)(x - x_{i+1}) \qquad (A3)$$

We find that:

$$f_{int}^1(x) = f(x) \quad \text{for} \quad x = x_i, \quad x = x_{i+1} \quad \text{and} \quad x = u \quad \text{(A4)}$$

We introduce the higher order error function $\varepsilon^1(x)$ defined by $\varepsilon^1(x) = f(x) - f_{int}^1(x)$. This error function can be rewritten as:

$$\varepsilon^1(x) = \varepsilon(x) - \frac{w(x)}{w(u)}\varepsilon(u) \quad \text{(A5)}$$

Because of (A4), $\varepsilon^1(x)$ has at least 3 zeros in $[x_i, x_{i+1}]$.

Hence, by Rolle's theorem, $\dfrac{d^2\varepsilon^1}{dx^2}(x)$ has at least one zero ξ in $]x_i, x_{i+1}[$. Hence there exists $\xi \in]x_i, x_{i+1}[$ (which, of course, depends on u) such that:

$$\frac{d^2\varepsilon^1}{dx^2}(\xi) = 0 = \frac{d^2\varepsilon}{dx^2}(\xi) - \frac{w''(\xi)}{w(u)}\varepsilon(u) \quad \text{(A6)}$$

and therefore:

$$\varepsilon(u) = \frac{w(u)}{w''(\xi)}\left(\frac{d^2f}{dx^2}(\xi) - \frac{d^2f_{int}}{dx^2}(\xi)\right) \quad \text{(A7)}$$

Noting that $\dfrac{d^2f_{int}}{dx^2}(\xi) = 0$ and that $w''(\xi) = 2$ (see (A3)), we finally obtain $\forall u \in]x_i, x_{i+1}[$:

$$\varepsilon(u) = \frac{w(u)}{2}\frac{d^2f}{dx^2}(\xi) \quad \text{(A8)}$$

The total quadrature error E_L is then:

$$E_L = \left|\int_a^b \varepsilon(u)\,du\right| \le \frac{1}{2}\sum_{i=1}^{I-1}\int_{x_i}^{x_{i+1}}\left|w(u)\frac{d^2f}{dx^2}(\xi(u))\right|\,du \quad \text{(A9)}$$

Using the assumption that $f \in C^2([a, b])$, there exists C such that:

$$\left|\frac{d^2f}{dx^2}(x)\right| \le C \quad \forall x \in [a, b] \quad \text{(A10)}$$

Noting that $\displaystyle\int_{x_i}^{x_{i+1}}|(x - x_i)(x - x_{i+1})|\,dx = \frac{1}{6}(x_{i+1} - x_i)^3$, we obtain:

$$E_L \le \frac{C}{12}\sum_{i=1}^{I-1}(x_{i+1} - x_i)^3 \quad \text{(A11)}$$

Finally, introducing the minimum sampling interval $h = \min_{i=1,\ldots,I-1}(x_{i+1} - x_i)$ and assuming that the discretization is such that the ratio between the largest and the smallest sampling intervals is bounded by a bound B in $O(1)$ (B, which is of course less than $(b - a)/h$, must be in $O(1)$ otherwise the nodes would not yield a genuine discretization of $[a, b]$), we obtain:

$$E_L \le \frac{C}{12}\sum_{i=1}^{I-1}B^3h^3 \le \frac{C(b - a)}{12}B^3h^2 \quad \text{(A12)}$$

B ERROR ESTIMATE OF HERMITE INTERPOLATION IN 1D

We use a finite element technique to implement a Hermite interpolation of a function $f(x)$ on $[a, b]$. We have discretized interval $[a, b]$ by introducing some nodes with abscissas x_i, for $i = 1, \ldots, I$, these nodes being such that $x_1 = a$ and $x_I = b$. The nodes may be irregularly distributed on $[a, b]$. Function f being given in $C^4([a, b])$, we denote by $f_{int}(x)$ the corresponding interpolation function. It is given by:

$$f_{int}(x) = \sum_{i=1}^{I}\left(f(x_i)e_i^0(x) + f'(x_i)e_i^1(x)\right) \quad \text{(B13)}$$

where $e_i^0(x)$ and $e_i^1(x)$ are third degree polynomials on each interval $[x_i, x_{i+1}]$ defined by:

$$e_i^0(x_j) = \delta_{ij}; \quad \frac{de_i^0}{dx}(x_j) = 0 \quad \forall i, j \in [1, I] \quad \text{(B14)}$$

$$e_i^1(x_j) = 0; \quad \frac{de_i^1}{dx}(x_j) = \delta_{ij} \quad \forall i, j \in [1, I] \quad \text{(B15)}$$

Our goal is to estimate the error involved in the interpolation procedure and, from this to derive the error involved in the associated quadrature formula. To do this, we use an adaptation of a very classical technique ([12] for instance).

We introduce the error function $\varepsilon(x)$ defined by:

$$\varepsilon(x) = f(x) - f_{int}(x) \quad x \in [a, b] \quad \text{(B16)}$$

and we analyze this function on an interval $[x_i, x_{i+1}]$ (with i arbitrary $\in \{1, \ldots, I - 1\}$).

On interval $]x_i, x_{i+1}[$ we introduce an arbitrary additional node with abscissa u and construct another interpolation function $f_{int}^1(x)$ defined by:

$$f_{int}^1(x) = f_{int}(x) + \frac{w^2(x)}{w^2(u)}\left(f(u) - f_{int}(u)\right) \quad \text{(B17)}$$

with $w(x) = (x - x_i)(x - x_{i+1})$.

We find that:

$$f_{int}^1(x) = f(x) \quad \text{for} \quad x = x_i, \ x = x_{i+1} \quad \text{and} \quad x = u \quad \text{(B18)}$$

and that:

$$\frac{df_{int}^1}{dx}(x) = f'(x) \quad \text{for} \quad x = x_i \quad \text{and} \quad x = x_{i+1} \quad \text{(B19)}$$

Besides, $\varepsilon^1(x) = f(x) - f_{int}^1(x)$ satisfies:

$$\varepsilon^1(x) = \varepsilon(x) - \frac{w^2(x)}{w^2(u)}\varepsilon(u) \quad \text{(B20)}$$

Now, because of (B18) and (B19), $\varepsilon^1(x)$ has at least one single and 2 double distinct zeros, *i.e.* at least 5 zeros in $[x_i, x_{i+1}]$. By recursive application of Rolle's theorem, we

determine that $\dfrac{d^4\varepsilon^1}{dx^4}$ has at least one zero ξ (which, of course, depends on u) in $]x_i, x_{i+1}[$. Hence:

$$\frac{d^4\varepsilon^1}{dx^4}(\xi) = 0 = \frac{d^4\varepsilon}{dx^4}(\xi) - \frac{\dfrac{d^4w^2}{dx^4}(\xi)}{w^2(u)}\varepsilon(u) \qquad (B21)$$

and using $\dfrac{d^4w^2}{dx^4}(\xi) = 4!$ we obtain:

$$\varepsilon(u) = \frac{w^2(u)}{4!}\big(f''''(\xi) - f''''_{int}(\xi)\big) \quad \forall u \in]x_i, x_{i+1}[\qquad (B22)$$

and considering that f''''_{int} is identically zero, we end up with:

$$\varepsilon(u) = \frac{w^2(u)}{4!}f''''\big(\xi(u)\big) \quad \forall u \in]x_i, x_{i+1}[\qquad (B23)$$

The total quadrature error E_H is then:

$$E_H \le \int_a^b |\varepsilon(u)|\, du = \frac{1}{4!}\sum_{i=1}^{I-1}\int_{x_i}^{x_{i+1}} w^2(u)\,\big|f''''\big(\xi(u)\big)\big|\, du \qquad (B24)$$

As $f \in C^4([a, b])$, $\big|f''''\big(\xi(u)\big)\big|$ is bounded by some constant D.

Noting that:

$$\int_{x_i}^{x_{i+1}} w^2(u)\, du = \frac{(x_{i+1} - x_i)^5}{30} \qquad (B25)$$

we obtain:

$$E_H \le \frac{D}{4!30}\sum_{i=1}^{I-1}(x_{i+1} - x_i)^5 \qquad (B26)$$

Introducing h and B as defined in Appendix A, the same reasoning leads to the result:

$$E_H \le \frac{D}{4!30}(b - a)B^5h^4 \qquad (B27)$$

C SCALING ARGUMENT

The α_k's are the solution of system (17)–(20) for given x_k's. If, by scaling of discretization, each x_k becomes λx_k, each α_k becomes α_k/λ. Besides, h is the characteristic size of the intervals involved in the discretization: this quantity can be viewed as the result of a scaling by h of an original discretization. This shows that the α_k's are $O(h^{-1})$, so that, for a function $f(x) \in C^4([a, b])$, the quantity $\sum_k \alpha_k x_k^4 \dfrac{d^4f}{dx^4}(\xi_k)$ is $O(h^3)$.

D OVERALL ERROR ESTIMATE OF HERMITE INTERPOLATION IN 1D INCLUDING NUMERICAL DERIVATION

When a numerical derivation procedure is used to carry out a Hermite interpolation, the interpolation function obtained is $\bar{f}_{int}(x)$ whereas it would have been $f_{int}(x)$ if the derivatives $f'(x_i)$, (for $i = 1, \ldots, I$) had been available. Regarding numerical integration, we compute $\int_a^b \bar{f}_{int}(x)\, dx$ instead of $\int_a^b f_{int}(x)\, dx$.

The overall error $\bar{E}_H$ is then given by:

$$\bar{E}_H = \left| \int_a^b \big(f(x) - \bar{f}_{int}(x)\big)\, dx \right|$$

$$\le \left| \int_a^b \big(f(x) - f_{int}(x)\big)\, dx \right|$$

$$+ \left| \int_a^b \big(f_{int}(x) - \bar{f}_{int}(x)\big)\, dx \right| \qquad (D28)$$

Then

$$\bar{E}_H \le E_H + E_D \qquad (D29)$$

where E_H is the error evaluated in Appendix B. E_D is the error arising from the numerical evaluation of the derivatives, *i.e.*:

$$E_D = \left| \int_a^b \big(f_{int}(x) - \bar{f}_{int}(x)\big)\, dx \right|$$

$$= \left| \sum_{i=1}^I \big(f'(x_i) - f'_{num}(x_i)\big)\int_a^b e_i^1(x)\, dx \right| \qquad (D30)$$

where $f'_{num}(x_i)$ is the numerical evaluation of $f'(x_i)$ by finite differences. We have seen that, when this evaluation makes use of two interpolation nodes in the vicinity of x_i, the error $|f'(x_i) - f'_{num}(x_i)|$ is $O(h^2)$. Besides, a straightforward calculation shows that:

$$\int_a^b \big|e_i^1(x)\big|\, dx \le \frac{1}{6}B^2h^2 \qquad (D31)$$

with B and h as defined in Appendix A.

Hence finally E_D is $O(h^3)$, and, according to the estimation of E_H given in Appendix B, so is $\bar{E}_H$.