Depth Migration of Complex Surfaces with GOCAD:
Study of the Curvature of a Triangulated Surface

Arnaud Berlioux [1]

ABSTRACT

The depth migration of time-migrated surfaces by demigration with GOCAD is inaccurate in the case of complex surfaces, such as folded horizons, in areas where the curvature of the surface reverses its sign. I believe this problem can be avoided by cutting the original complex surface into pieces for which the depth migration is accurate. To do so, I propose an algorithm to study the sign of the curvature of the GOCAD surface in order to divide it up properly before performing the depth migration. Putting the depth-migrated pieces back together, I should then obtain a depth-migrated surface more representative of the true geology than the direct depth-migrated surface.

INTRODUCTION

In an earlier report (Berlioux, 1993) I presented a method for 3-D depth migration of 3-D time-migrated maps with GOCAD. The process consists of two parts: first I use the principle of demigration; then the depth migration is performed.

The demigration of 3-D complex time-migrated surfaces produces even more complex demigrated surfaces. The result of the demigration of a 3-D syncline is a classic example of such complex surfaces: the demigrated surface presents a bow-tie, a triplication that has to be processed carefully before going on with the depth migration.

Such triplications appear when the curvature of the time-migrated surface goes to zero and reverses its sign. Consequently, examining the sign of the curvature of the time-migrated surface associated with a cut algorithm for the same GOCAD object can, I believe, enable us to migrate complex horizons in depth with greater accuracy.

---

[1] **email:** arnaud@sep.stanford.edu

In this paper, I investigate the possibility of using the curvature of a GOCAD surface to characterize its complexity. I also address some of the problems associated with analyzing the sign of the curvature. Finally, I present some results of depth migration of a synthetic complex surface.

TERMS OF THE PROBLEM

In 1992, I built a simple algorithm using GOCAD (Mallet, 1994) to perform map depth migration starting with time-migrated horizons. This algorithm, described in a previous SEP report (Berlioux, 1993), consists of two parts and can be summed up as follows:

- first, the time-migrated surfaces are demigrated in order to retrieve the original zero-offset time and position;
- then, the zero-offset horizons are migrated in depth, taking into account the interfaces previously migrated.

The first step requires the root mean square velocity (V_{rms}) that has been used to produce the time migrated maps. In the second step, the zero-offset horizons are migrated with a user-defined velocity model.

Interpretors feel more comfortable with time-migrated sections that show a better picture of the area under investigation than the premigrated stack sections. Time migration focuses the diffraction, and improves the general aspect of the sections, but often misplaces the horizons. Time demigration, as the inverse operation of time migration, restores these diffractions and retrieves the original position.

A syncline represents a classic example of such complex surfaces. On a stack section, the event corresponding to the syncline presents diffractions that disappear after time migration. Figure 1 shows a 3-D time-migrated syncline. The demigration of this syncline produces a zero-offset surface with a triplication (Figure 2). Since GOCAD represents surfaces as a continuous triangular patch (a set of vertices connected to form contiguous triangles), the demigrated surface presents a bow-tie with two rounded edges that produce the anomalies visible on the depth-migrated surface (Figure 3): the depth-migrated surface has a M shape with two branches going upward to infinity (cut in the figure).

The same kind of behavior can be observed when the curvature of the initial time-migrated surface goes to zero and reverses its sign. When we look at Figure 3, or any depth-migrated complex surface, we can see that in the areas where the curvature of the initial surface has a constant sign, the result of the depth-migration process seems accurate. On the other hand, on the edges where the curvature changes its sign, the process produces strange results (mostly points going upward to infinity). This observation lead me to a new algorithm for the depth migration of complex GOCAD surfaces.

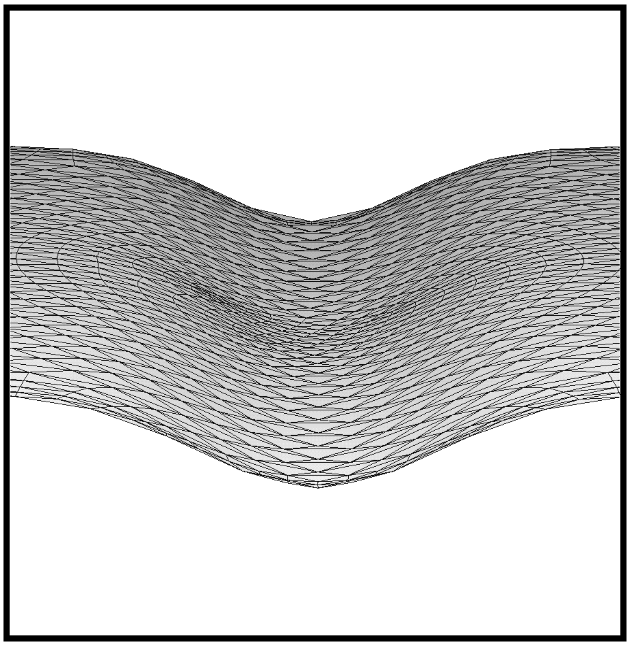

Figure 1: 3-D time-migrated syncline. SYN25 [NR]

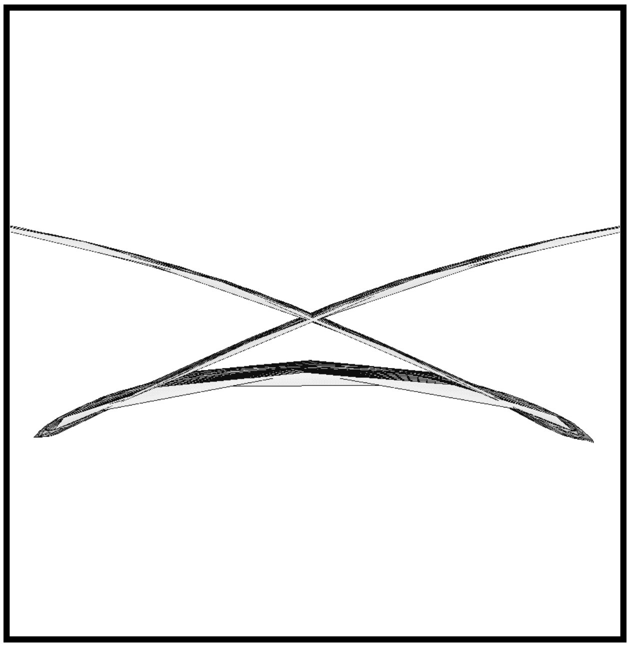

Figure 2: Surface resulting from the demigration of the syncline in Figure 1. SYNdmg25 [NR]

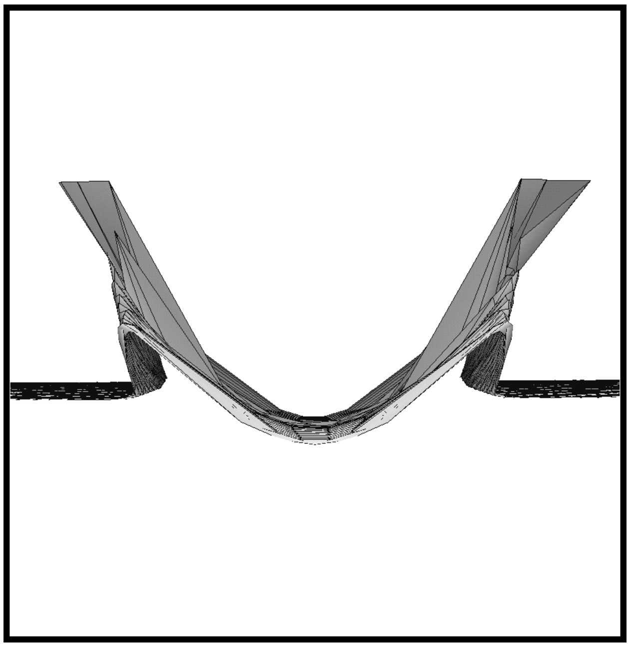

Figure 3: Result of the depth migration after demigration of the syncline in Figure 1. SYNmig25 [NR]

CURVATURE OF A TRIANGULATED SURFACE

The curvature of a continuous surface can be analytically calculated when the position of each point on the surface is given by a set of explicit equations or computed in the case of implicit or complex equations. In the discreet domain, where the position of a point on a surface cannot be simply expressed by equations, the calculation of the curvature requires some work.

Curvature of a discretized surface

The curvature of a discontinuous set of points can be approximated by fitting a patch to this set prior to any calculation, as described in Mortenson's book (1985a). A patch is a curve-bounded collection of points whose coordinates are given by a set of continuous two-parameter single-valued functions of the form

$$\begin{array}{rcl} x & = & x\,(\,u,\,w) \\ y & = & y\,(\,u,\,w) \\ z & = & z\,(\,u,\,w) \end{array} \quad \text{noted} \quad \vec{p} \;=\; p\,(\,u,\,w) \tag{1}$$

where the parametric variables u and w are constrained to the interval $[\,0,\,1]$, and $\vec{p}$ is the position vector. By choosing small subsets of the domain $[\,u_{min},\,u_{max}] \times [\,w_{min},\,w_{max}]$, we can define several particular portions of the surface and thereby study the curvature, for example, around different points of the surface. The chosen domain of operation is $[\,0,\,1] \times [\,0,\,1]$. Any other domain can be normalized to this unit square in the parametric space.

The curvature at a point on a surface is one of the intrinsic properties of a surface, which also include the normal and the tangent plane.

Through a point p on a surface S passes an infinity of curves c for which we can compute the curvature at that point. In the general case, all the curvature values thus calculated are different. Let us consider the curvature of all the curves c that lie on the surface S and pass through the point p with the same tangent line t at p. These curves all have the same center of curvature, radius of curvature, curvature vector, and principal normal curvature.

Among all the possible curves, some are of particular interest: the intersection curves cut on the surface by a plane N passing through the tangent line t, and also those containing the normal $\vec{\eta}$ to the surface at p (Figure 4.) When the plane N rotates around the normal $\vec{\eta}$, the curvature varies and has a maximum and a minimum in two orthogonal directions. These extreme values, denoted as κ_1 and κ_2, are the principal normal curvatures.

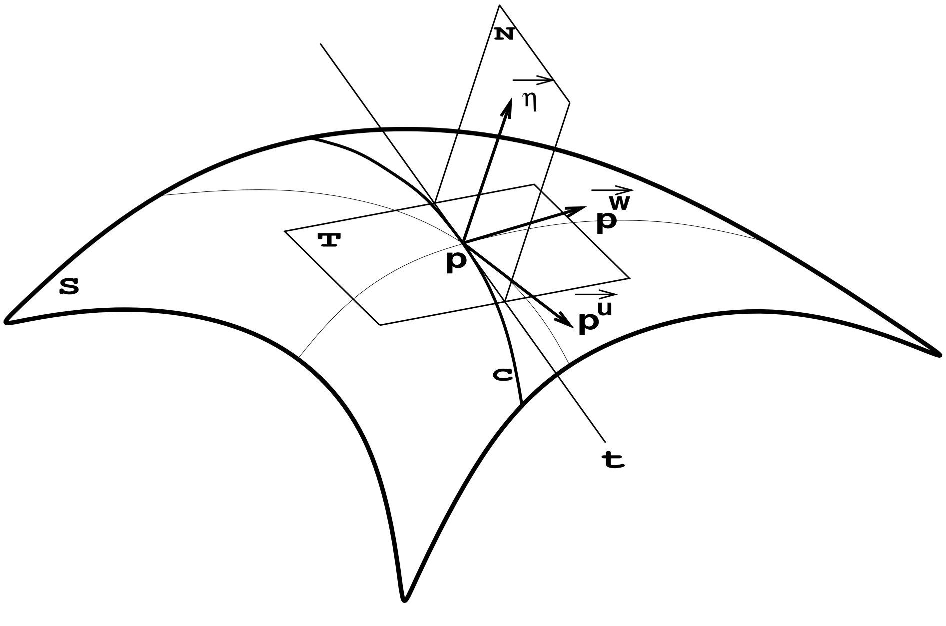

Figure 4: Normal curvature on a surface S at a point p associated with the curve c. curvature [NR]

The derivation of κ_1 and κ_2, and any value of the curvature κ_n in general, is developed in Mortenson's book (1985b). The values κ_1 and κ_2 are useful to define two related measures of the curvature of a surface:

- the Gaussian curvature K,
$$K = \kappa_1 \kappa_2 \tag{2}$$
which is an invariant property of the surface, and

- the mean curvature H,
$$H = \frac{1}{2}(\kappa_1 + \kappa_2) \tag{3}$$

which is more relevant for our problem.

Curvature of a surface with GOCAD

In GOCAD, a triangulated surface can be approximated by a local cubic Bezier patch (Mallet, 1994). The computation of the curvature at any point on a GOCAD surface is based on this patch.

The computation of the two intrinsic properties K and H is available in GOCAD. K, being a constant value for a whole surface, is not useful for our problem. However, the mean curvature H characterizes well the behavior of the dip of a surface. It has the additional advantage that it goes to zero where inflection points are present.

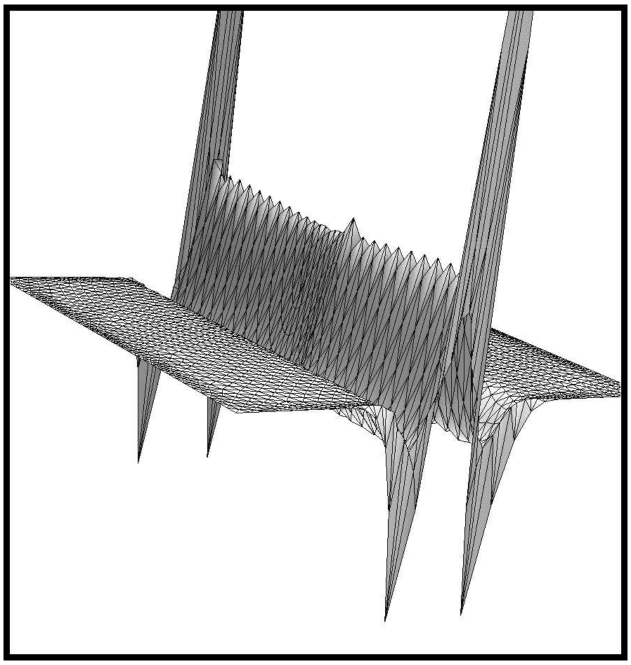

Figure 5: Mean curvature of the syncline surface in Figure 1. SYNcrvH [NR]

Figure 5 represents the mean curvature of the syncline in Figure 1. As described in the first section, I am interested in the sign of the curvature as a means to finding out more precisely where the inflection points are located. Figure 6 is a magnified picture of a surface whose points are at the same x and y locations as on the original time-migrated surface (Figure 1), and the third coordinate is either $+1$ or -1, corresponding to the sign of the curvature at the point. The plateaux of the surface denote a positive sign; the valleys, a negative sign.

MAP MIGRATION OF COMPLEX HORIZONS

This section explains how I modify my original depth-migration algorithm to take into account the information given by the curvature of the initial time-migrated

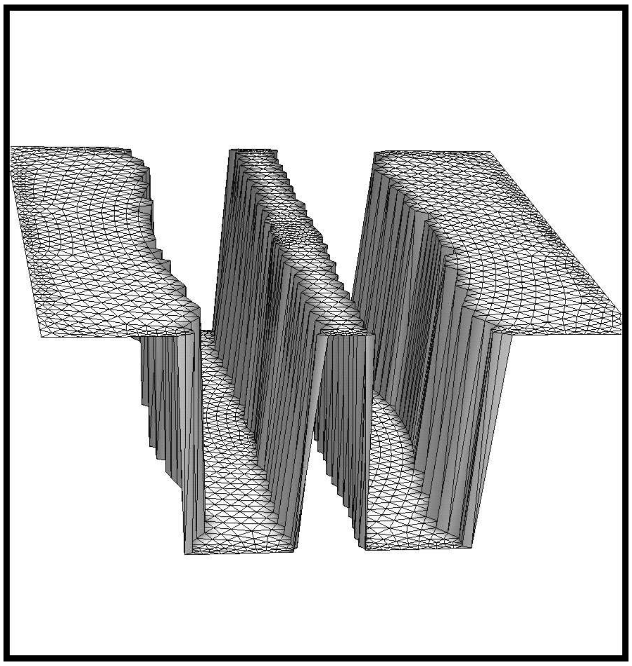

Figure 6: Sign of the mean curvature in Figure 5. SYNcrvHsignP [NR]

surface. I also comment on the results obtained with the surface in Figure 1, and propose some improvements for future work.

Strategy

If the original surface is cut into pieces on which the sign of the curvature is constant, the problem associated with inflection points that occurs with the depth migration process should not arise. Accordingly, I want to cut the surface shown in Figure 6 into plateaux and valleys. The location of the cuts will be mapped to the initial time-migrated surface. Each fragment thus created will be migrated separately.

Therefore, I propose the following algorithm to perform the map migration of complex surfaces with GOCAD:

1. Study the sign of the curvature of the surface.
2. Cut the complex surface into pieces with constant curvature.
3. Demigrate and then migrate in depth each piece separately.
4. Reconstruct the surface in depth from the fragments.

Current status of the project

This subsection presents the results I obtained with the new algorithm for depth migration, starting with the syncline surface in Figure 1.

Figure 7 represents the initial surface after the cut has been applied. The cutting algorithm uses the surface that gives the sign of the curvature (Figure 6) as a reference to indicate where to break the initial surface into pieces. The resulting surface has four cuts corresponding to the four transitions from plateau to valley or valley to plateau visible in Figure 6. The cutting algorithm detects the triangles that have to be cut and removes the connection between the three apices of the triangles.

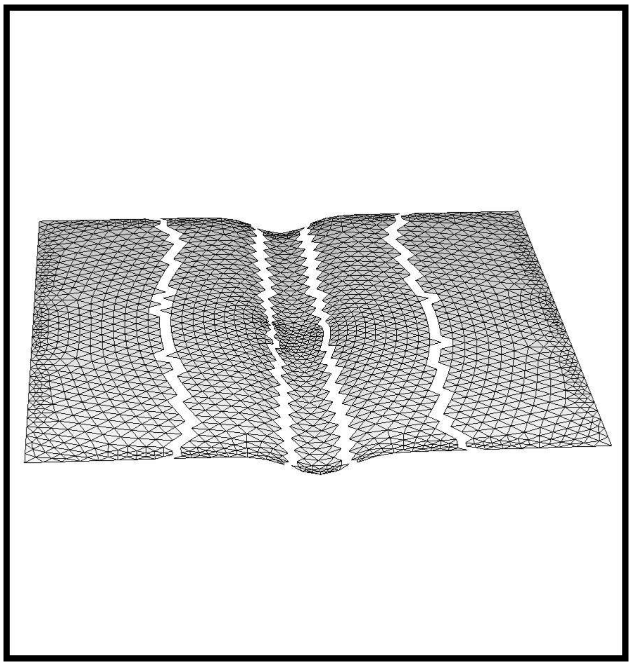

Figure 7: Syncline surface in Figure 1 after the cutting algorithm has been applied. SYNcut [NR]

Running the demigration and depth-map-migration subroutines for this surface produced the results shown in Figures 8 and 9. Figure 8 represents the demigrated syncline surface. The triplication has been cut into several pieces. The left and right branches of the triplication are cut just before the round edges that caused trouble during the depth-migration process. When these edges are present in the central piece of the triplication, they create some mispositioned triangles in the depth-migrated surface.

One of the problems associated with the cutting algorithm is visible in both Figure 7 and Figure 9: the syncline surface should not need to be cut on the left and right part where it is flat. The cut is made automatically; the algorithm cuts the surface using the information given by the sign of the curvature but nothing else. Consequently, the surface in Figure 7 is cut because these regions have been detected as areas of major changes in the curvature.

The effect of the cutting algorithm is particularly visible in Figure 9: the depth image of the syncline itself has been improved in comparison to that shown in Figure 3, but the intervals between the syncline and the flat parts on the left and the right need some more work.

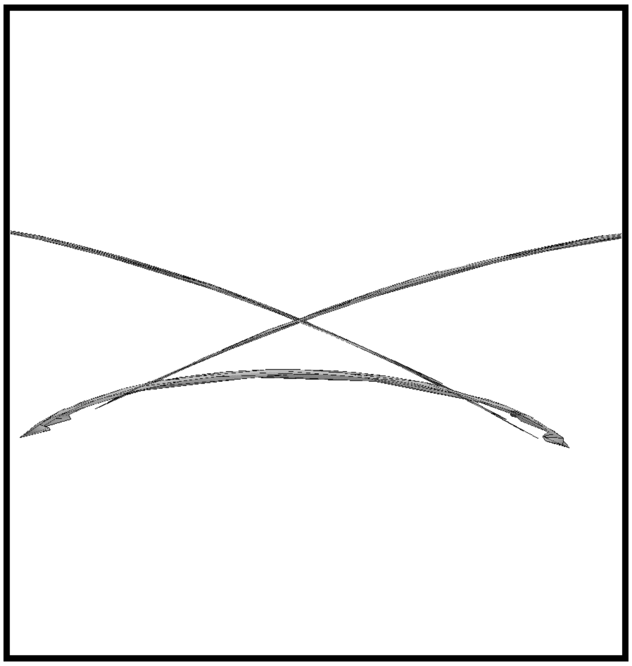

Figure 8: Demigrated surface of the cut syncline in Figure 7. SYNcutdmg [NR]