

The visualisation of integrated 3D petroleum datasets in ArcGIS

Alistair Ford

Abstract

This paper describes the use of ArcGIS and the 3D Analyst extension for the integration of multiple petroleum datasets as part of research undertaken by the University of Newcastle upon Tyne, UK, on behalf of Shell International Exploration and Production. The integration of datasets from both surface and subsurface application areas is essential to present a complete picture for decision-making, project overviews and data quality assurance processes. This paper will illustrate how such integration is possible using the ESRI Geodatabase for true 3D feature storage (using multipatch features), and examine the 3D spatial analysis and visualisation tools available within 3D Analyst.

Introduction

The recent advances in interoperability and standards, in addition to the breakout of GIS from a specialist application towards being a widespread desktop tool, has meant that it is now in an ideal position to provide a powerful data integration tool in many industries. Many industries however, such as meteorology, environmental management or oil and gas, deal with complex 3D spatial data. In order to provide full integration GIS must develop a capability to handle the complex three-dimensional (3D) data types and spatial representations which exist in the various disciplines within these industries. This paper will examine the current state of 3D data storage in the ESRI suite of products, and give examples of the incorporation of numerous disparate data types into a comprehensive pilot dataset within ArcGIS. Using the oil and gas industry as a case study, it will also describe how this data can be analysed and visualised within the ESRI environment as used by Shell.

The systems integration problem – is 3D GIS the solution?

The use of 3D modelling, analysis and visualisation software in the oil and gas industry is widespread. Amongst geoscientists, 3D views of the world are an important tool for increasing understanding, improving communication and simulating complex geological and petrophysical structures and processes. Amongst planners and engineers 3D visualisation is an important way of assessing the impact of projects, examining numerous implementation options and ensuring health and safety requirements are met. Be it on land, under the sea or under the ground, the ability to visualise, analyse and interact with objects of interest in 3D is of great value.

Most of these 3D visualisation, analysis, and modelling capabilities are today provided by specialised software applications, with functionality geared towards specific analyses, provided on specialist hardware (such as UNIX-based geological modelling software). Whilst these applications are not viewed as GIS by their users, there is no doubt that the majority deal with geo-spatial data, spatial representations of real world objects and with spatial analysis. These systems could therefore be viewed as simply 'specialised GIS'.

Despite this proliferation of three-dimensional visualisations and the enabling software, one problem is still faced by geoscientists; integration. The separation of people and systems is a huge barrier to true cross-organisation working and team interaction, and the difficulties in encouraging integration and cooperation in the geosciences are well documented (Abel et al, 1994; Breunig, 1999). This is mainly due to the software-led and discipline-led development of proprietary data formats, spatial models and representations, in a very similar vein to that which mainstream GIS industry historically followed. Some strides have been taken towards

integrating various subsurface modelling packages (such as the OpenSpirit initiative), but the true integration of all data available from surface and subsurface is still not directly possible.

Conversely, mainstream GIS vendors, such as ESRI, have begun to move away from the strictly proprietary data formats which were historically used. Geo-interoperability Initiatives have been established in recent years (such as OGC, GML) to provide open, extensible and standardised spatial representation that encourages interoperability amongst all parties in the spatial community. Other advances, such as the inclusion of the Safe Software FME extension into ArcGIS 9.0, allow the conversion of data between numerous formats from disparate vendors and applications, enabling a vast array of data to be incorporated into GIS projects.

In addition to these advances in geo-spatial data standards and interoperability, the use of GIS is also becoming more widespread. The migration of GIS from specialist workstations and complex software to a user-friendly application capable of running on almost any desktop PC has meant that in large organisations its use is no longer the preserve of cartographers, planners and data managers. In the petroleum industry, for example, GIS is now used by many branches of the organisation from geologists and petrophysicists through to project managers.

With these advances in flexibility and status, along with the traditional strengths of coordinate system support and spatial analysis, mainstream GIS now finds itself in an ideal position to serve as the primary integration toolkit for spatial data (and especially in the oil and gas industry). In order to provide full integration however, GIS must be capable of handling the complex three-dimensional (3D) data types and spatial representations which exist in the various disciplines within the industry.

Required objects and possible representations

Many GIS users consider '3D GIS' to be simply the provision of a 3D view of standard 2D and 2.5D spatial data. This 3D aspect is often seen to be a cosmetic addition to GIS, as whilst the data may be visualised in three dimensions, the majority of the spatial analysis, editing, and other common GIS tasks are undertaken in a strictly 2D environment. Whilst simply viewing data in 3D may be sufficient for many applications, some disciplines require a 'true 3D' spatial representation of important objects due to the complex nature of features to be modelled or to allow for spatial analysis in three-dimensions.

Most literature and thought on 3D data modelling for vector GIS (e.g. Carlson, 1987; Molenaar, 1990; de la Losa and Cervelle, 1999) presents models which are built up from a set of common primitives. Data structures constructed from these primitives are either object oriented or topology oriented, depending on whether the application which will use the data is expressly concerned with visualisation or analysis of the data. Topological representations expressly store the relationships between any object and its neighbours, whilst object oriented models store the structure of the objects, deriving the topology, and therefore spatial relationships as and when required (Zlatanova et al, 2002). Since the integration of data in this case study is aimed at providing a high-level management tool without replicating any of the specialist analysis present in the originating software, the emphasis should be towards the object oriented representation of features.

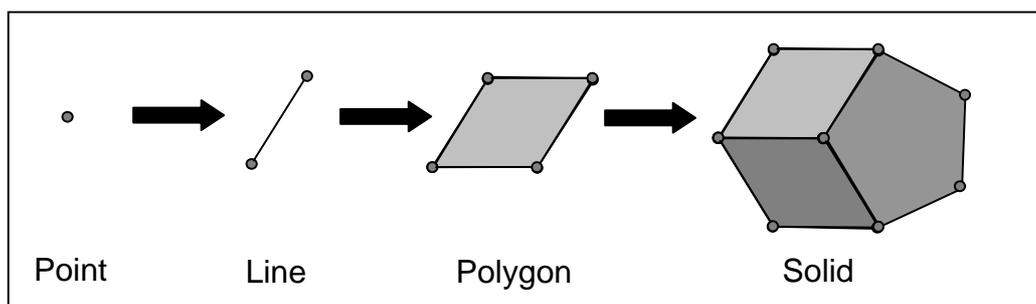


Fig. 1: – Basic 3D geometric primitives

The basic components of any model can be summarised as points (or nodes), lines (or edges), faces (or polygons) and solids (see *Fig. 1* above). It is quite evident therefore that the geometric primitives of a 3D data model are essentially those in a standard 2D or 2.5D data model.

Point Features

Point features can be represented by the most basic spatial primitive and can be viewed as a three-dimensional location in space. As with 2D GIS, point features rarely exist in the real world as there are very few entities that are zero-dimensional (i.e. have no area or volume) but are an extremely useful construct for creating higher-dimensional spatial representations. The point feature can be used to represent data points; locations where measurements are made or values exist. Examples of data points in the geological case study are geological markers (or well picks), where a boundary exists between two lithological layers along a well.

Line Features

Line features are a collection of point features which are linked together to form a one-dimensional feature. As with point features, line features are abstractions of real world features, providing a suitable representation for simple storage and analysis. Examples of linear features in this case study are well trajectories, pipelines and cables. In 3D space a line is still a one-dimensional feature even though it does not sit on a flat surface as it has only length, but not width or height.

Point and line features are special, since they do not exist in the real world (nothing in the real world has zero width or height) but they are key to constructing more complex three dimensional features.

Polygon Features

Polygon features are segregated portions of space which define areas with common attributes. A polygon is a two-dimensional feature which consists of one or more rings of lines forming an enclosed boundary. They are used to represent planes or surfaces; features which have a spatial extent in two-dimensions, but no thickness. In 3D space, polygons are often viewed as two-and-a-half-dimensional (2.5D) features, as the points from which they are built may contain three-dimensional coordinates (giving the feature height variation) but the feature itself is still 2D. Examples of polygonal features in the geological world are fault planes and lithological horizons. A DEM can also be viewed as collection of contiguous polygonal features and is therefore implicitly 2.5D; it is a surface with zero thickness but with height variation.

Solid Features

Solids are 'true 3D' features consisting of a number of polygonal faces defining an enclosed boundary within which is a region of common attributes. Unlike the above features, solids are not restricted to 2.5D representation allowing complex entities such as oil and gas reservoirs to be represented in a GIS database. The creation of solid features, however, is an extremely difficult process. Whilst a polygonal feature can be represented by defining the order of points which create the boundary, determining the order of points which form a solid feature is particularly complex.

3D Feature storage in ArcGIS

As outlined above, there is a raft of literature which outlines data models for 3D features in GIS databases (e.g. Carlson, 1987; Molenaar, 1990; de la Losa and Cervelle, 1999), although most of this presents excellent theoretical storage methods which could be implemented in a bespoke GIS systems. Most large organisations, however, are committed to large commercial GIS and DBMS contracts (e.g. Shell Exploration and Production with ESRI and Oracle). Any 3D data storage and associated functionality must therefore be developed within the restrictions of the current database and GIS technologies.

Some proposals have been made for the storage of 3D objects in commercial DBMS (e.g. Oracle; see Slotter and Oosterom, 2002), although this results in a solution which is reliant on a single database vendor's technology. In contrast, GIS data storage engines, such as ArcSDE, are designed to be independent of database technologies. Also, ArcSDE allows simpler integration of numerous datatypes and with spatial data in mind.

In order to fully integrate the various surface and subsurface datasets, GIS must be capable of handling 'true 3D' data. Whilst most commercial GIS software (ESRI included) allows the assignment of the third dimension in coordinates and includes some level of 3D visualisation, the features themselves are often limited to 2.5D. This means that whilst the locations of a feature may contain a 'z' value, the structure of the feature itself may not be true 3D; the feature is restrained to a simple surface. For example, a polygon feature may be constructed from numerous points with z values, but two points making up the polygon may not have the same (x,y) value. This is because all topological operators acting on the polygon at the time of its creation (to ensure the polygon has no edges which cross, for example) act solely in two dimensions.

One example of this is the attempt to construct a box from a polygonal representation in an ESRI geodatabase. In the real world, a box can be created from a simple 'net' (see Fig 2.0), which itself can be represented in a GIS as a polygon feature. This feature is perfectly valid in 2D, and even in 2.5D (Fig 2.1), but once an attempt is made to 'wrap up' the polygon to form the box, the topological operators will remove any 'duplicate' (x,y) points which form the vertical sides of the box (despite the fact that they are distinct vertices with different z values) as the feature is invalid.

This shows that in a 2.5D system, a feature must meet with the demands of a 2D oriented world. It is obvious from this that the creation of true 3D feature using standard geo-spatial objects such as points, lines and polygons will not be possible. The provision of a 'true 3D' feature in a geodatabase is required to enable the storage of complex 3D objects alongside the more common 2D or 2.5D spatial objects.

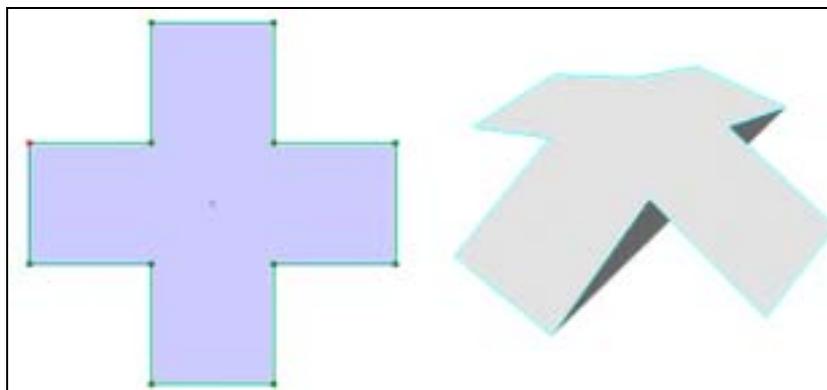


Fig 2.0: Polygon box net

Fig 2.1: 2.5 box net

Multipatch features

In the ESRI geodatabase there exists an elusive feature which could achieve such a goal. The 'multipatch' feature is designed for the on-the-fly creation of 3D symbology for visualisation. The multipatch feature is also currently ESRI's answer to true 3D geodatabase objects, building on the OpenGL 3D primitives of triangles, which can be created in strips and fans (see Fig 3.0). A multipatch feature is simply an additional ESRI geometry type (much like points, polylines or polygons) and can therefore be assigned in the geometry field of a feature in a shapefile or geodatabase. This way, spatial entities can be represented with true 3D features and still be queried, selected and have full attribute data and associated symbology within ArcMap and ArcScene.

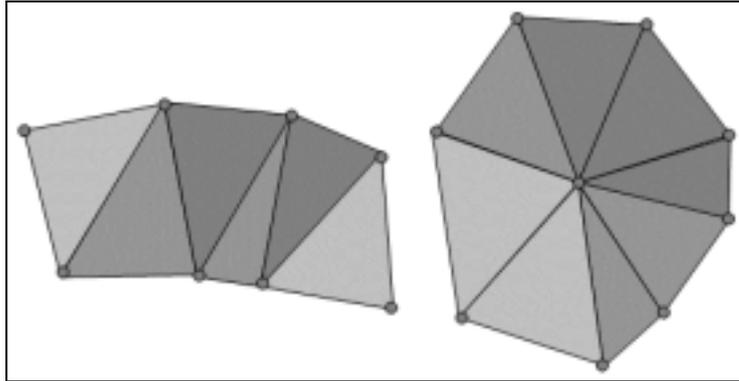


Fig 3: Triangle Strip and Triangle Fan

The multipatch is essentially a solid feature, consisting of a number of polygon faces which define a boundary. The multipatch feature can therefore be utilised as a means of storing complex three-dimensional features in a geodatabase. Although multipatch features are specifically designed for display purposes, it can be seen that they are able to fulfil the role of a 'true 3D' spatial representation.

The multipatch feature cannot be created through the normal edit sessions within ArcMap as with the standard ESRI geometry types. These features must be created using ArcObjects scripting, either from existing geometries (this is how buildings are extruded in ArcScene and ArcGlobe) or from raw data sources. The following section outlines some of the issues with multipatch creation.

Multipatch creation using ArcObjects

After examining a number of potential applications, it was decided that in the pilot ArcGIS dataset, the use of multipatches for storing true 3D features would be tested on reservoir and pipeline objects. An oil or gas reservoir is a complex 3D object, with irregular shape and structure. Whilst the geological interpretation and modelling of such objects was beyond the scope of this study, the storage of the resultant reservoir objects in a geodatabase was seen to be an excellent test of the multipatch as a 3D representation.

A pipeline could be stored as a line feature in geodatabases, but for some 3D applications this representation is insufficient. For example, when carrying out analysis of sub-sea pipelines for exposure (areas where a pipeline on the seabed has become exposed due to ocean currents and is therefore in danger) the inclusion of a true 3D representation can be combined with a detailed seabed surface (from a swathe bathymetry survey, for example) to give a visual indication of areas of dangerous exposure (see Fig. 4).

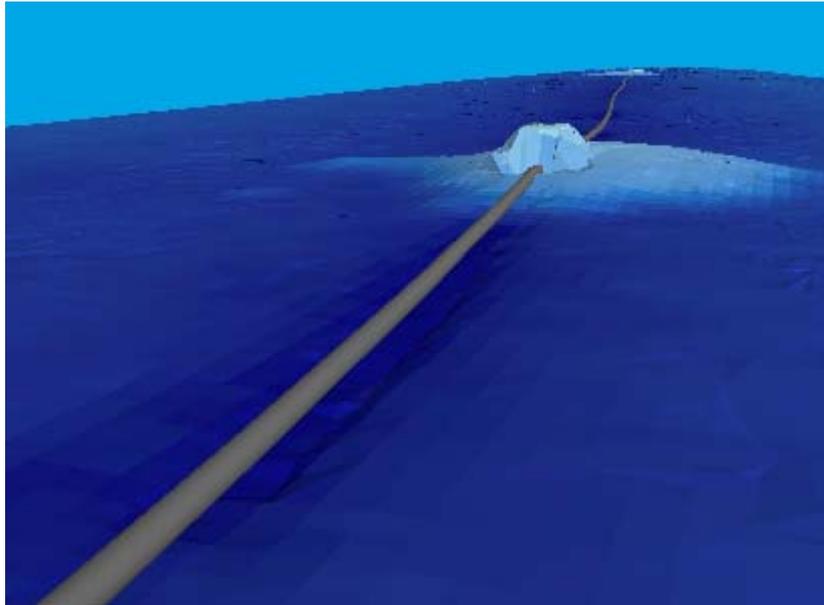


Fig. 4: 3D pipeline created with multipatch feature

Since they cannot be created directly in ArcMap, most multipatch features will be created from existing data. In the oil and gas industry, and particularly the geological application area, most data transfer is done in the form of ASCII text files representing features as a series of x,y,z points. These files can then be loaded through ArcCatalog, either as a data table or an XYZ Featureclass. It is extremely important to know and understand the order of the x,y,z points in these featureclasses, as this impinges directly on the way that the multipatch is built up from the raw data. For example, for data concerning a pipeline it is reasonable to assume that the x,y,z points will be ordered such that each point is sequenced along the feature. Conversely, with a complex geological feature, knowledge the structure of the input data is vitally important. As most native data formats used by the modelling and interpretation software in which reservoir models will be created is not open source, it is often required

As multipatch features can be assigned to the geometry field of a geodatabase record, their creation through ArcObjects initially follows the same method as creating any standard ESRI feature. Initially the geodatabase `FeatureClass` must be created, along with the associated `SpatialReference`. Since the multipatch is often created from an input `FeatureClass`, the `Spatial Reference` can be derived directly from this as shown below:

```
Dim pGeoDataSet As IGeoDataset
Set pGeoDataSet = pInputFeatureClass

Dim pSR As ISpatialReference
Set pSR = pGeoDataSet.SpatialReference
```

The multipatch is a three-dimensional feature, so it is essential that the Z Domain in the `SpatialReference` of the output feature class is set. The `GeometryType` property of the `Geometry` definition in the `Shape` field of the `Feature Class` must be set to `esriGeometryMultiPatch`.

Once the output `Feature Class` is created, it can then be populated with the 3D features. As outlined above, a multipatch feature consists of a number of triangle strips or triangle fans. In ArcObjects, these must be created initially and populated with the input points in the correct order, and then added to the multipatch feature. The order in which the points are added depends on whether the triangle strip or triangle fan is used. A triangle strip assumes that every point added creates a triangle in conjunction with the previous two points (see *Fig. 5*). It was found that with most applications, the triangle fan results in the simplest implementation.

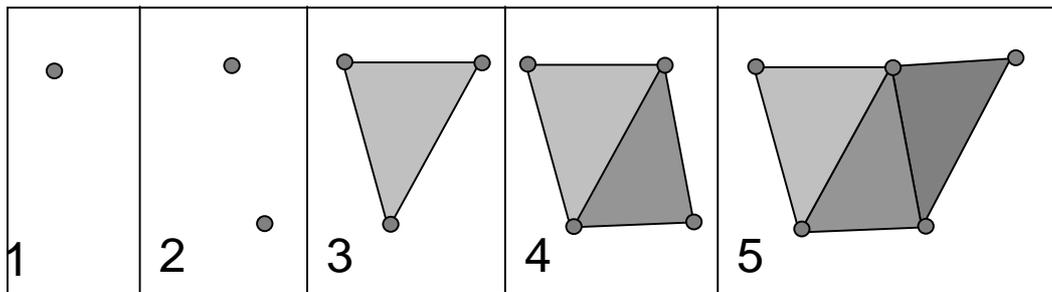


Fig. 5: Triangle strip creation

Initially an `IMultiPatch` feature is declared in `ArcObjects`, along with an `IGeometryCollection` and a `IPointCollection`.

```
Dim pMultiPatch As IMultiPatch
Dim pGCol As IGeometryCollection
Dim pStrip As IPointCollection
```

```
Set pMultiPatch = New MultiPatch
Set pGCol = pMultiPatch
Set pStrip = New TriangleStrip
```

The Geometry Collection object is used as an interface to the Multipatch feature, since it is essentially a collection of Triangle Strip and Triangle Fan geometries. Each Triangle Strip or Fan is essentially a collection of Point features, hence the need for the `IPointCollection` object.

Each point in turn is then added to the Point Collection in the correct order, until the Triangle Strip or Fan is completed. It is essential for the points which make up a triangle in the object to be ordered clockwise in order for `ArcObjects` to determine which side of each triangle is the outside and compute normals (these are required for correct 3D visualisation later).

```
pStrip.AddPoint pTempPoint1
pStrip.AddPoint pTempPoint2
pStrip.AddPoint pTempPoint3
```

```
pGCol.AddGeometry pStrip
```

Once all points are added to the Triangle Strip or Fan, this geometry is then added to the Geometry Collection (and hence the Multipatch feature). The final step is to ensure that the Multipatch is Z Aware (as it is a three-dimensional object) and add the feature to the output Feature Class through an Insert Feature Buffer as normal.

The Pipeline Object

The pipeline object was created from a series of points representing the centreline of the pipe, extracted from swathe bathymetry data from an inspection survey. These were read into ArcGIS through ArcCatalog and a 3D Point Feature Class created from the x,y,z coordinates. In `ArcObjects`, an `IVector3D` object was created for each pair of points defining the pipeline, and a second vector created perpendicular to this with a magnitude equal to the required diameter of the pipe. The second vector was then rotated around the first, creating a new `IPoint` object a distance equal to the diameter away from the centreline of the pipe. This created a new ring of points at each vertex along the pipe which could then be added in the correct order to a new Multipatch feature to create a 'tube' representation (see *Fig. 6*)

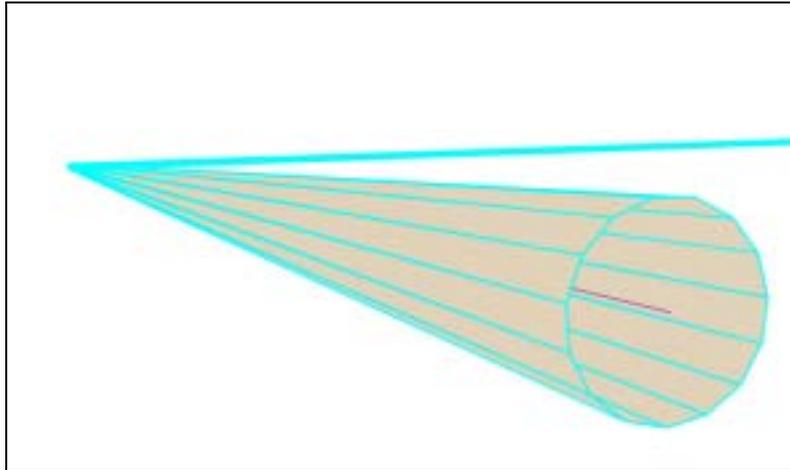


Fig. 6: Multipatch pipeline created from points defining centreline

Reservoir Object

Whilst the creation of a pipeline object was relatively straightforward, an oil or gas reservoir feature has greater complexity. It was decided to attempt the storage of a reservoir model created in Schlumberger's geological modelling software Petrel, as this is widely used and Windows-based. Petrel uses 'faulted grids' in its reservoir models, which allow the user to define irregular variations throughout a solid structure. These grids are composed of three-dimensional cells of irregular shape and size, the number of rows, columns and layers throughout an object being constant rather than cell size.

A Petrel model was exported in Eclipse ASCII grid format, which defines the two opposite corner points of each cell. These, in conjunction with the corner points of neighbouring cells, can be used to build up the irregular structure of the 'faulted grids'. The corner points of each cell were loaded into ArcGIS as XYZ Points as before. A number of nested loops were then written in ArcObjects parse each cell and the corresponding neighbours in the correct order, adding each point to a Triangle Strip as before. A Multipatch feature was created for each cell, allowing each an individual record in the geodatabase table, and there allowing the assignment of attributes on a cell-by-cell basis. A layer of one such 'faulted grid' is shown in Fig. 7.

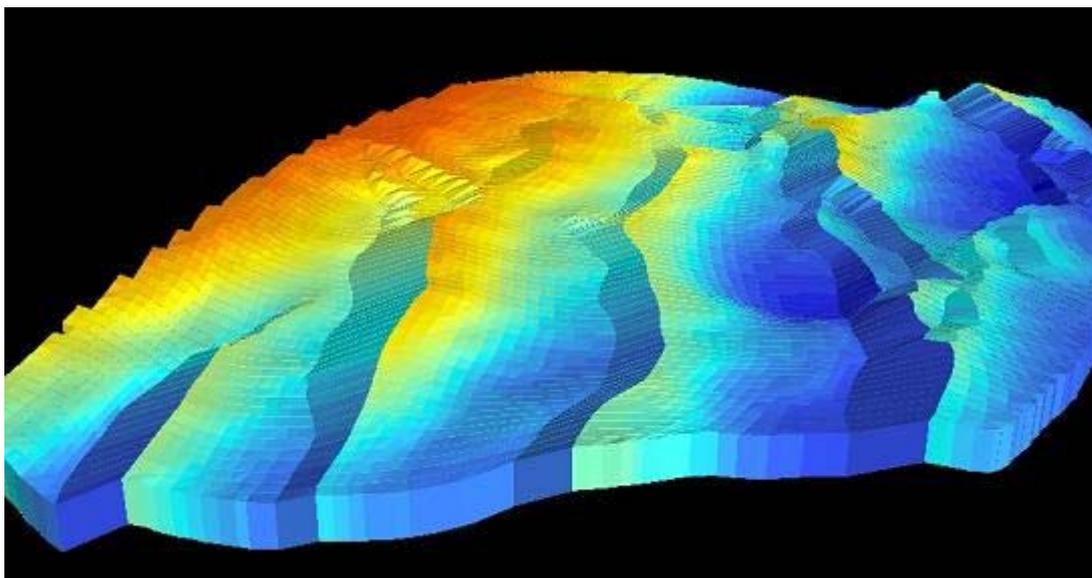


Fig. 7: Faulted grid stored as Multipatch features in an ESRI Geodatabase

Once Multipatch features have been created within the ESRI geodatabase, their behaviour in ArcMap or ArcScene is essentially identical to any other feature type. The symbology of a Multipatch feature may be set through the normal Layer Properties dialog, allowing the user to alter the display of the Multipatch dependent upon any of the attributes in the geodatabase table. Whilst in most geological modelling software each cell is assigned only one property (or attribute), ArcGIS allows the storage of multiple attributes about each cell and the interactive display of these properties through symbology. For example, the grid in *Fig. 7* is displaying the maximum depth of each grid cell whilst each cell may also have information on permeability of the rock at each cell, flow properties or the name of the geological layer (see *Fig. 8*).

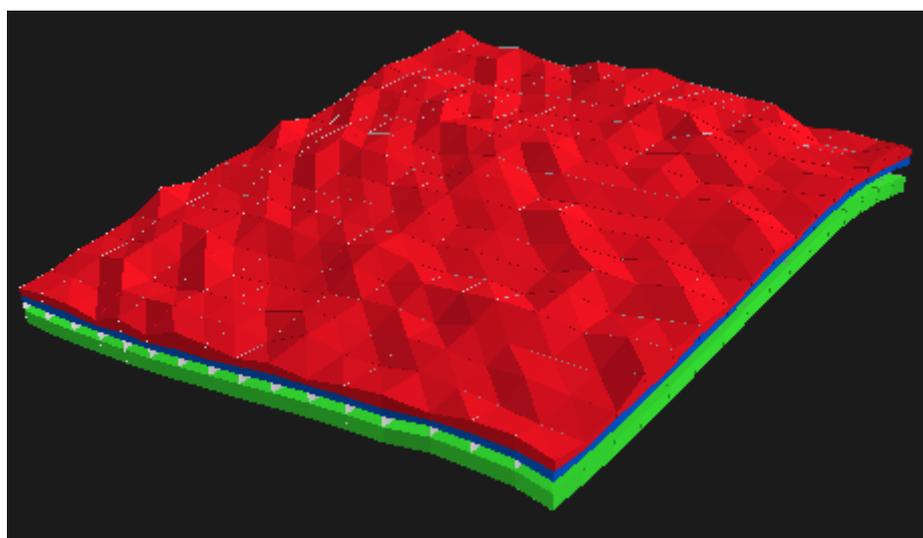


Fig. 8: Cells displayed by geological layer

Issues and limitations

It can be seen from previous sections that the multipatch feature allows for the storage of complex three-dimensional features in an ESRI geodatabase. Whilst this is the first step towards true three dimensional objects in a GIS, there is still a number of limitations with this approach. These are outlined below.

Data volumes

The geological application was deliberately chosen as a case study as it offers the greatest challenge to GIS developers, not only because of the complexity of the objects to be modelled but also due to the vast quantities of data to be stored. Each cell in the 'faulted grid' model above is represented by 12 triangles in a multipatch feature. A grid with a reasonably coarse resolution may still consist of 50 x 50 x 10 cells, which would require 300,000 triangles to represent it in this manner. It is instantly evident that this is an impractical means of storing such large quantities of data. Similarly, 1.5km of pipeline stored as a simple line feature is barely 1Mb of data, whilst the same pipeline stored as a multipatch feature (even with a low level of detail) can amount to over 100Mb.

There are numerous other data types in the oil and gas industry, such as seismic data, storage of which results in huge data volumes. Data storage in most geological packages has remained relatively simple in a bid to improve performance (often using flat ASCII files for storage), whilst GIS this has been less of a concern. One solution to this problem would be to mimic the geological modelling software and limit storage of such data to simple point features, creating the complex multipatch representations on-the-fly when complex visualisation or analysis is needed. This would require more work at runtime but would reduce the storage cost. It is obvious that the storage of all pipeline features as true 3D objects is impractical, whereas storage of a line feature representing a pipe, with an numerical attribute

representing the pipe diameter, would allow for on-the-fly creation of the required section of pipeline at the time of visualisation or analysis.

3D Symbology

The introduction of 3D symbology at ArcGIS 9.0 has rendered the creation of multipatch representations of features for visualisation trivial. For example, voxel data (uniform 3D arrays of x,y,z points with attributes) can be stored as a grid of points and symbolised using 3D symbology, thus vastly saving storage costs. In order to carry out 3D spatial analysis, however, it is essential for the 3D geometry of objects to be existent and accessible rather than simply visualised.

One area where 3D symbology is particularly powerful is in the creation of standardised libraries of 3D objects, which can be included in any visualisation. Complex three-dimensional objects, such as oil platforms, refineries, well heads or pipe components, can now be created from this common library and simply stored as point features with the necessary attributes to assign location and orientation to the 3D object.

Representing variation

One interesting problem in the geosciences is the representation of spatial variation of attributes. It was noted above that in some geological modelling software, variation is represented by dividing an object into 'cells'. In GIS, features are traditionally represented by either vector (discrete) or raster (continuous) data. In some cases however, this is insufficient. For example, a reservoir is a discrete entity with boundaries, but the attributes of this object vary spatially throughout. It is not sufficient therefore to store a single object with one attribute for permeability, as this attribute itself is varying dependent on the location within the object.

In one-dimensional objects (i.e. line features) it is currently possible to represent this. The use of dynamic segmentation allows the spatial variation of attributes along the length of a line. Through this technique it is possible to illustrate how the characteristics of a linear feature vary through space. For example, along a well feature it is possible to illustrate the variation of lithological layer or even returns from a well log. This can be achieved by referencing a table of data defining the attributes at given distances along a line to the measures along the line geometry itself.

In the current GIS mindset however, it is not possible to account for both raster and vector representations of a 2D or 3D object. Whilst variation in 2D space can be represented by raster features and in 3D space by voxels, it is not possible to combine both of the representations into one feature in a database. The need for this 'hybrid' representation is clearly one area of future work into 3D object modelling.

Conclusion

This paper has discussed some of the many issues which must be resolved in order to include true three-dimensional objects in an ESRI geodatabase. It has outlined the data modelling considerations and basics of 3D object modelling, before describing the methodology for implementing such a model in the geodatabase using multipatch features. It can be seen from this paper that there is much work to be done to provide comprehensive 3D functionality in commercial GIS software. In addition to data storage, advances must be made in visualisation and spatial analysis in three dimensions before any GIS can truly call itself "3D".

Acknowledgements

The author wishes to thank Professor David Parker, Mr Philip James and Dr David Fairbairn at the University of Newcastle upon Tyne's School of Civil Engineering and Geosciences, and Dr Thierry Gregorius and other staff of Shell International Exploration and Production for their assistance and support. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Petrel is a trademark of Schlumberger Information Services.

References

- Abel, D. J., Kilby, P. J. and Davis, J. R.** (1994), The systems integration problem, *International Journal of Geographical Information Systems*, 8 (1), 1-12.
- Breunig, M.** (1998), An approach to the integration of spatial data and systems for a 3D geo-information system, *Computers and Geosciences*, 25 (1), 39-48.
- Carlson, E.**, (1987), Three dimensional conceptual modelling of subsurface structures, *Technical Papers of ASPRS/ACSM Annual Convention, Baltimore, Vol. 4 (Cartography)*, 188-200
- de la Losa, A., Crevelle, B.** (1999), 3D Topological modeling and visualisation for 3D GIS, *Computers and Graphics*, 23(4), 469-478.
- Molenaar, M.**, (1990), A formal data structure for 3D vector maps in: *Proceedings of EGIS'90, Vol. 2, Amsterdam, The Netherlands*, 770-781.
- Sloter, J. a. v. Oosterom., P** (2002), Incorporating 3D geo-objects into a 2D geo-DBMS, *ACSM-ASPRS 2002 Annual Conference Proceedings*.
- Abdul-Rahman, A., Zlatanova, S., and Shi, W.** (2002), Topology for 3D spatial objects, *Proc. of International Symposium and Exhibition on Geoinformation 2002*, Kuala Lumpur

Author Information

Alistair Ford
PhD Student in Geomatics
School of Civil Engineering and Geosciences
University of Newcastle
Newcastle upon Tyne
United Kingdom
Email: a.c.ford@ncl.ac.uk