Synthetic model building using a simplified basin modeling approach

Robert G. Clapp

ABSTRACT

Generating a realistic synthetic model is a challenging problem in a geophysical research environment. Achieving the right balance between being complex enough to be realistic while still simple enough that a new algorithm can be debugged is hard to achieve. I propose a different way to generate synthetic models, by allowing the user to specify a series of geologic events. The result of each event is approximated on the current model. This approach has the benefit of allowing complex models to be built is easily extendable to multiple model parameters, and allows the user to "turn off" events, allowing the construction of simpler models by stages. A geologic event-based modeling strategy proves to be useful to build simple to quite complex models.

INTRODUCTION

Synthetic models play an important role in geophysical research. Over the last 25 years, SEG and company generated synthetics (Versteeg, 1993) have been used extensively by researchers throughout the world. These synthetics have found such wide use for two reasons. First, generating realistic synthetic models is difficult, particularly 3-D models. Tools such as Gocad (Cain et al., 1998) allow the user to specify layers, faults, and other geologic features before creating a 3-D grid to ease the process, but still require significant experience to use effectively. Second, industry wide synthetics provide a common benchmark to compare results from different companies. These models are useful for final testing of a finished algorithm, but often are too complex to be used in algorithmic development.

Another approach to generating synthetic models is following the methodology of basin modelers. Basin modeling takes a more geologic approach to describing models than the more computer science-based approach followed by GoCad and its competitors. Basin modeling attempts to model the geologic history of a given piece of the earth to explain its current properties.

In this paper I build on the tool introduced in Clapp (2013) for generating more realistic synthetic models. Specifically I improve the fault, emplacement, and river erosion models while adding unconformities and compressional folding. In this paper I describe these new modules, introduce a python script that simplifies the model building process, and provide a complete list of all of the parameters for each model building module.

FAULTS

The simplified view of faulting that you see in introductory geology textbooks where you a see bulk shift of all the layers on one side of the fault move up, or down, or side to side, does not accurately reflect how faulting actually occurs at the large scale. Movement away from the fault location decays away from fault surface in all directions. In addition fault planes are not the planar surfaces that one often sees depicted in illustrations.

In `ModelCreate` faults are created on a cylndrical coordinate system. The user describes the beginning and ending of the fault plane in terms of a rectangular coordinate system. These are then mapped into a polar coordinate system by defining the angular range represented by the beginning and ending points of the fault surface. Using this methodology, a user can make a fault virtually planar by defining a small angular range and curved using a larger angular range. The user defines the amount of movement in terms of angular shift. Figure 1 graphically illustrates how the fault is described.

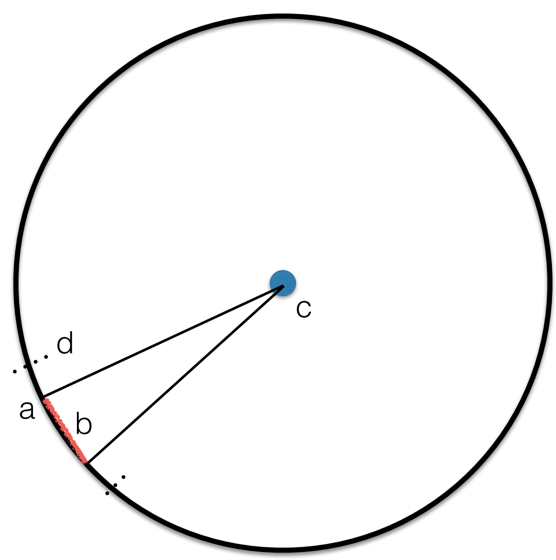

Figure 1: Fault movement is defined by an initial point 'A' and a point 'b' defined by distance away from 'a' in z and x. In addition the user defines the arc length separating points a and b (the red line). Using the arc length along with points 'a' and 'b' a circle is implied with center 'c'. Fault movement is then described in terms of this circle. The user specifies the die out of fault movement 'd' in terms of arc length along with the shift in radians along the circle.

The decline in movement along the fault is also defined in terms of the cylindrical coordinate system in terms of radial distance, angular range, and perpindicular distance. This coordinate system can then be rotated to an arbitrary azimuth. Figure 2 shows two faults. The fault on the left shows a nearly linear fault, using a small angular range. The fault on the right shows a more curved fault surface using a large angular range.

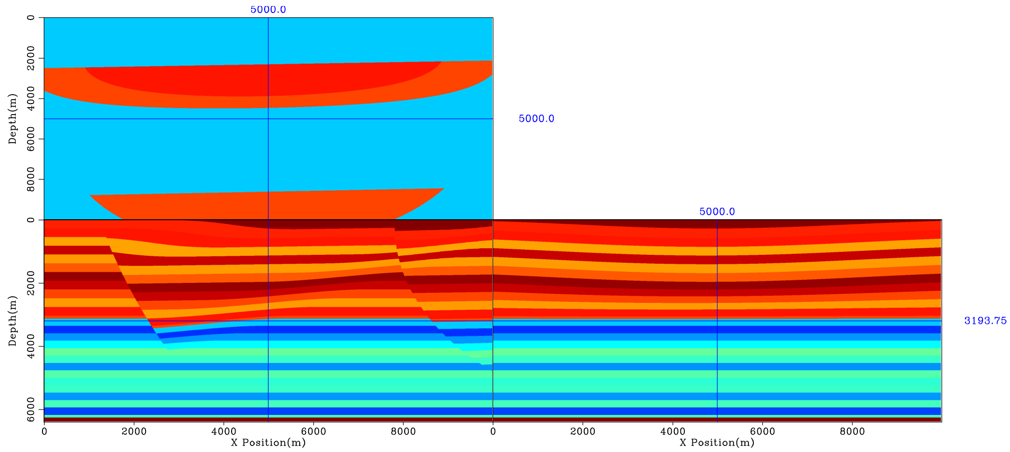

Figure 2: Two examples of faults. In the lower left fault a distant center point is calculated by specifying a small angular range compared to shift in rectangular coordinated. The upper right fault is curved by specifying a large angle compare rectangular distance resulting in a smaller circle (see explanation in Figure 1.

RIVER CHANELS

The previous version of `ModelCreate` (Clapp, 2013) was capable of producing river channels but required significant parameterization by the user to get reasonable shapes. As a result I overhauled the module and added additional complexity. The user now specifies a general direction for the river in terms of azimuth. The user also specifies a wavelength and wave amplitude which gives the river its sinusoidal like appearance In addition the user specifies a randomness factor. This factor is used to add semi-random other wavelengths to the river's path, creating a more realistic pattern. In addition the user can specify a number of levels in the river's paths. This parameter takes into account the slope of the river's path to create deeper river bed sediments that are consistent with the river's curvature. Figure 3 shows an example of using these features. The left panel shows several different rivers. The right panel shows a smaller region of the model where deeper river channels can easily be seen.

EMPLACEMENT

Describing salt flow, which would follows the spirit of `ModelCreate` , is quite challenging and is not something I have undertaken at this time. Instead I allow bodies to be emplaced into the geologic scenario. Just replacing one earth property with another does not produce a pleasing model in all cases, because sediments are affected by the flowing in of salt. As a result the emplacement module offers the ability to warp the surrounding structure. To do this I first smooth the emplaced body. This creates a field that is at its maximum inside the body and dies out away from the body. This field is used the shift sediments up. Shifts are greatest close to the salt

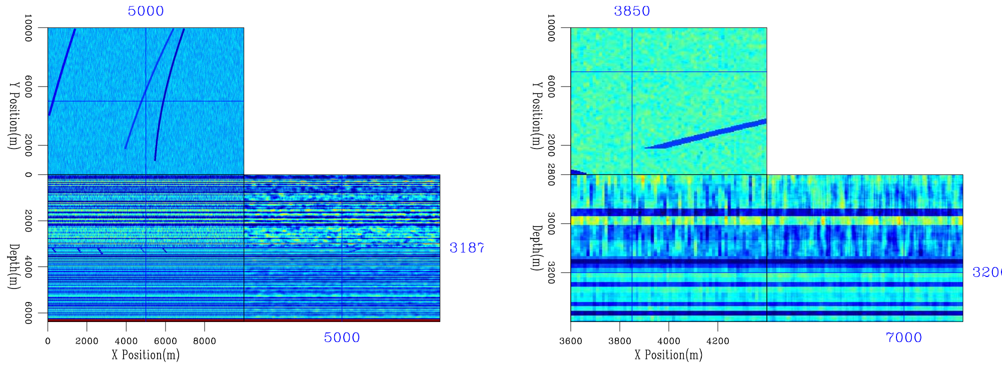

Figure 3: The left panel shows an example of constructing several different river channels. The right panels shows a blowup around the river chanels. Note the stacked river channel feature where the river channel has moved over time.

body and die off as you get further from body. Note how the sediments are drawn up to the sides along the salt and the structure of the salt top and bottom can be seen around the salt in Figure 4. In addition I allow the user to change the earth's properties under the salt (e.g. underpressure). Again this is a smooth field that dies off away from the salt.

FOLDING AND EROSION

In the original version of `ModelCreate` I allowed the user to create hills. Geologically, hills alone are usually associated with some compressional event that also leads to valleys, and there is often some level of spatial consistency because the entire region is under the same compessional regime. As a result I added a module that allows the user to describe a compressional stress direction and the amount of uplift associated with it. These two values, along with the average wavelength of the hill/valley pattern and a description of randomness inline and crossline are used to calculate a spatial field of vertical shifts up and down. Figure 5 shows an example of using this geomodule. Note how the hill/valley pattern, while showing a dominant wavelength, also has spatial variation.

I also added a third erosion option (to go along with a river channel and bowls), a planar erosion event. With the unconformity option, all of the model down to a certain depth is scraped off, to be replaced with later sediment deposits. Figure 6 shows the result of eroding the hill/valley pattern off to certain depth (note how some of anticlines are cut) and then adding additional layering over it.

GEOMODEL.PY

Creating a realistic geologic model using `ModelCreate` can involve hundreds of different geologic events. Given that each event has up to 12 parameters it can quickly

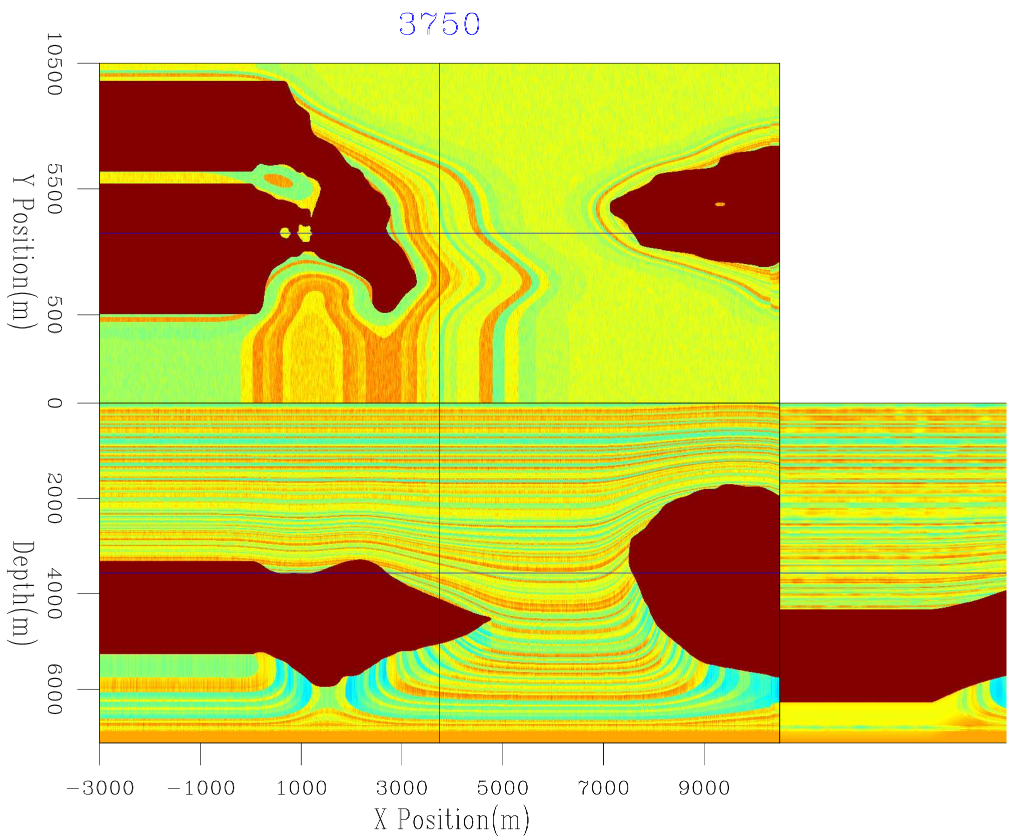

Figure 4: An example of emplacing a salt body into a model. In this case the sediments around the salt have been modified so that they are affected by the emplacement. In addition a low velocity region has been added below the salt.

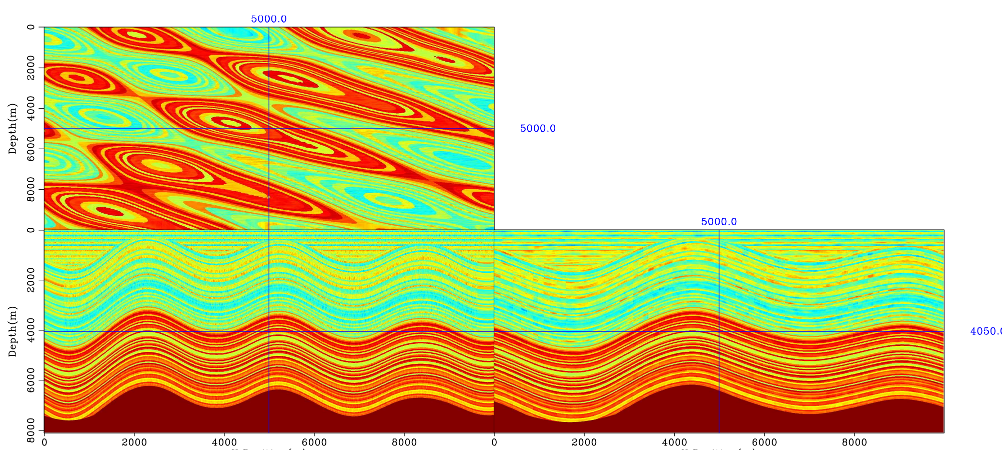

Figure 5: An example of introducing compressionalforces and folding the layers. Note how the folding pattern is somewhat repeatable but still varies significantly. In addition, but harder to see, the thickness of the various beds change spatially.

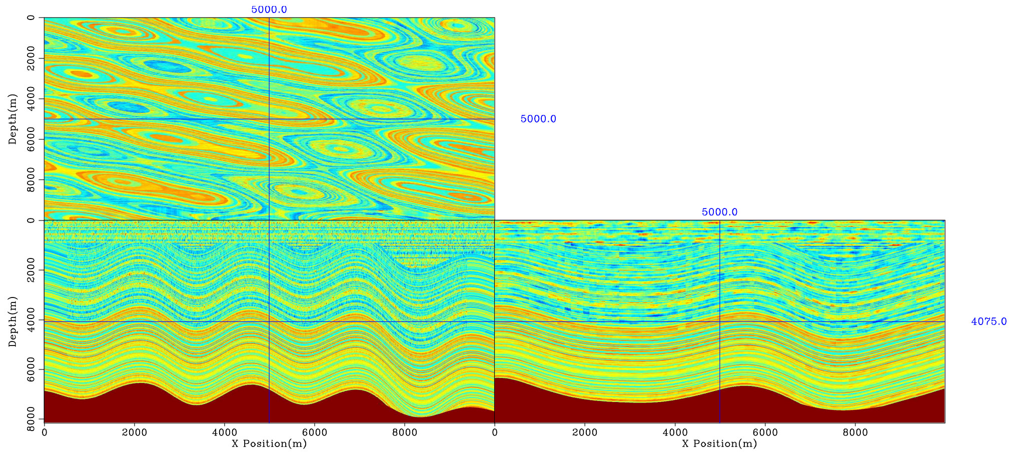

Figure 6: An example of a horizontal unconformity event. Note how the top of some of the anticlinal structures has been eroded off.

become a daunting task. To make it a more reasonable task, `Geomodel` is a python library to ease the model building process.

The basic idea of the library is that specifying every parameter to every module at every step is too time consuming. A better idea is to specify a reasonable range of possible parameters for each geologic event (controlled by a `geomodule`). When the user asks for an event to be added to the geologic history, the parameters are randomly chosen from the acceptable range the user specified. The user at anytime has the option of changing the possible parameters range.

For example, to create the fault model shown in Figure 2 I began by importing the `Geomodel` library and making a copy of the library default parameter set.

```
import Geomodel
myDefaults=Geomodel.defaults()
```

I then changed some of the default parameters for the deposit and fault modules.

```
myDefaults.change_param_ranges("deposit",
 ["thick:500:500","dev_pos:.2:.4","dev_layer:.2:.29","layer:11:39"])
myDefaults.change_param_ranges("fault",["begz:.1:.3","begy:.1:.9",
 "perp_die:.5:.9","dist_die:1.2:16","theta_shift:1.:1.2"])
```

I specified that I wanted to create a new model using my default parameter set, the sampling in z, the axes in x and y, and the parameter file I wanted to write out to.

```
mod=Geomodel.model(myDefaults,6.25,800,0,12.5,800,0,12.5,3000,20,"temp.P")
```

I then changed the velocity range I wanted for my first layer and added it to the model.

```
mod.change_param_ranges("deposit",["prop:2200:2600"])
mod.add_event("deposit")
```

I created a new layer, changed the velocity range, and deposited another layer.

```
mod.new_layer()
mod.change_param_ranges("deposit",["prop:2300:2300"])
mod.add_event("deposit")
```

I next change some of the parameters for the fault model, created a fault event, changed the parameters again, and created another fault event.

```
mod.change_param_ranges("fault",["begx:.1:.4","dz:1000:2000","daz:500:502",
  "azimuth:1:3","deltaTheta:4:5","theta_die:2:4"])
mod.add_event("fault")
mod.change_param_ranges("fault",["begx:.55:.9","dz:2000:4000","daz:500:502",
"azimuth:1:3","deltaTheta:32:36","theta_die:20:23"])
mod.add_event("fault")
```

Finally, I finished my geologic model and wrote its description to the parameter file.

```
mod.finish()
```

Using this scheme, complex models can be created. Figure 7 shows three slices through a 3-D model containing six layers, eighteen faults, two compressional events, fifteen gaussian anomalies, and five river channels.

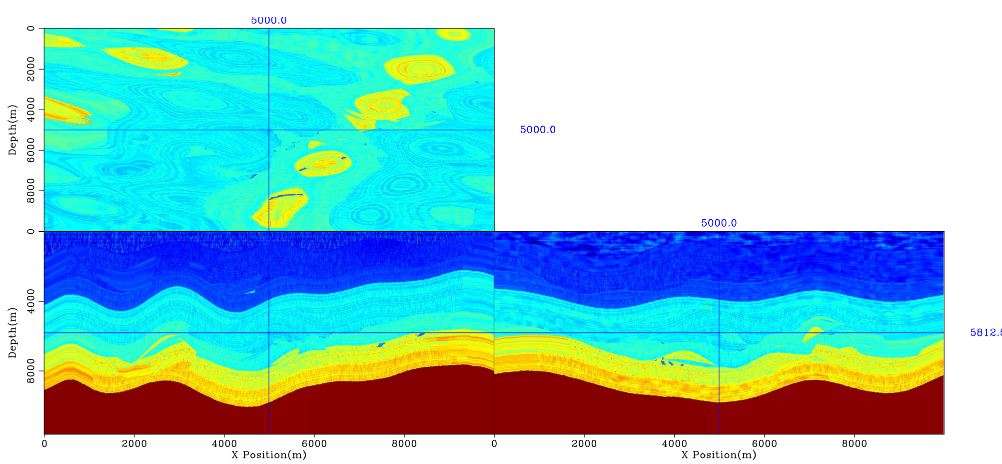

Figure 7: Three views of a complex synthetic containing deposition, river channels, faults, compressional folding, and gaussian anomalies.

CONCLUSIONS

Complex synthetic models are useful not only for benchmarking but in the code development process. Complex models can be built up by simulating a series of geologic events such as deposition, erosion, compressional events, and faulting. `ModelCreate` and `Geomodel` used together allow for relatively easy construction of complex models.

REFERENCES

Cain, G., G. Cambois, M. Gehin, and R. Hall, 1998, Reducing risk in seismic acquisition and interpretation of complex targets using a Gocad-based 3-D modeling tool: 68th Ann. Internat. Mtg, 2072–2075, Soc. of Expl. Geophys.

Clapp, R. G., 2013, Synthetic model building using a simplified basin modeling approach: SEP-Report, **149**, 273–280.

Versteeg, R. J., 1993, Sensitivity of prestack depth migration to the velocity model: Geophysics, **58**, 873–882.