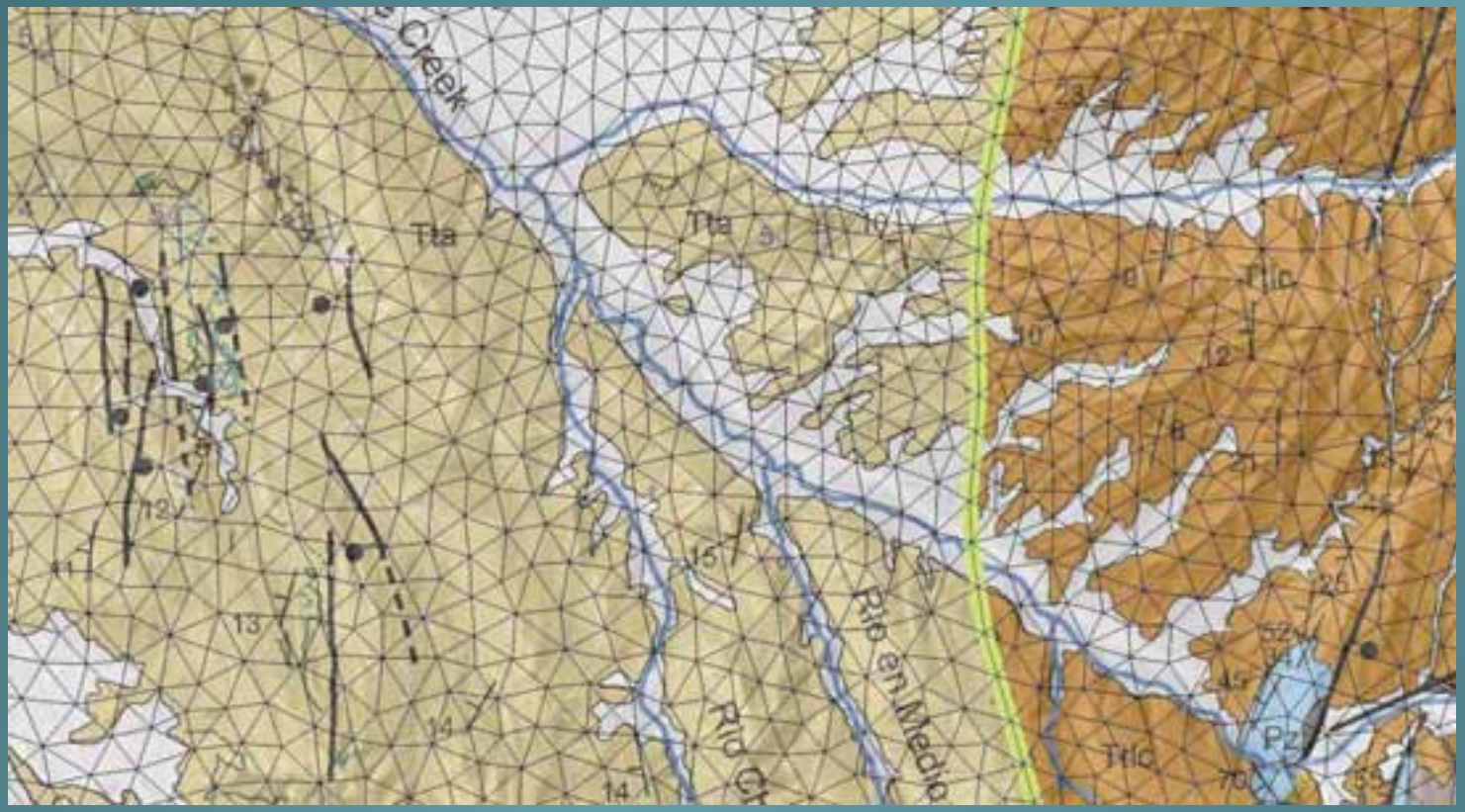
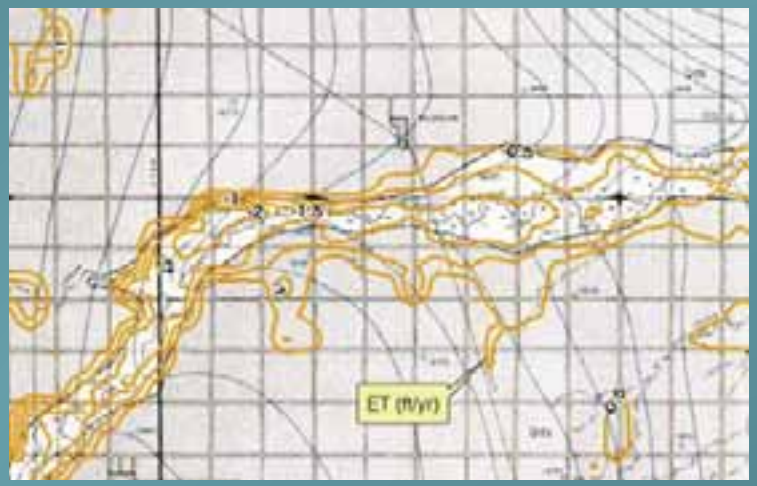
Getting Out of Squaresville: MODFLOW USG Tools for ArcGIS

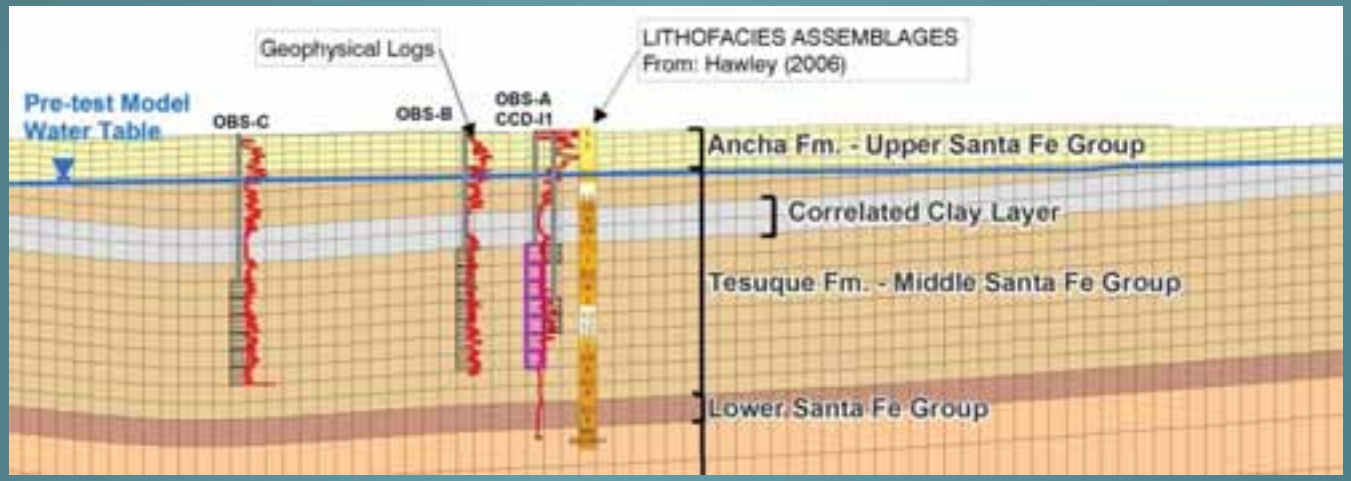
Steven E. Silver, GISP



Why Use ArcGIS for Hydrologic Modeling ?

1) Access map-base and data





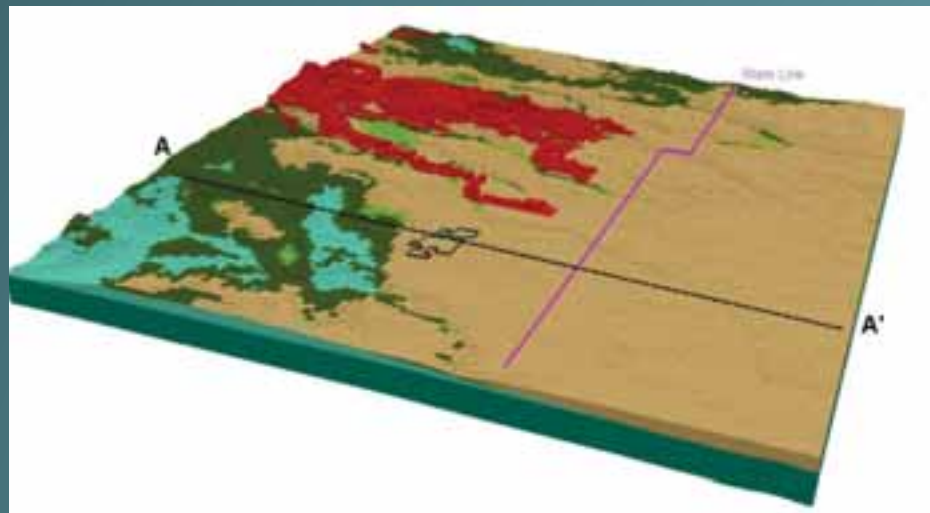
Why Use ArcGIS for Hydrologic Modeling ?

2) Flexibility to meet new classes of problems

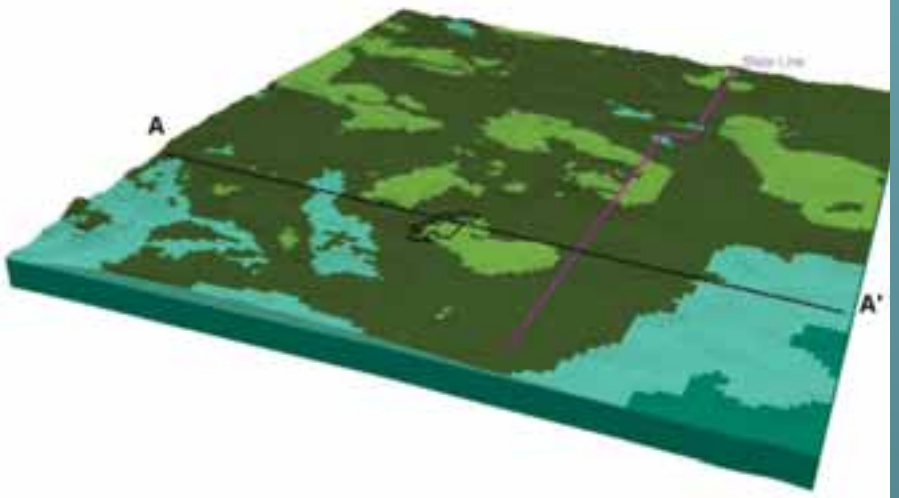
Use well known MODFLOW input output patterns and model concepts and flexibly apply geo-spatialy aware scripting tools to achieve novel analysis and visualizations of hydrologic model and related data.

Why Use ArcGIS for Hydrologic Modeling ?

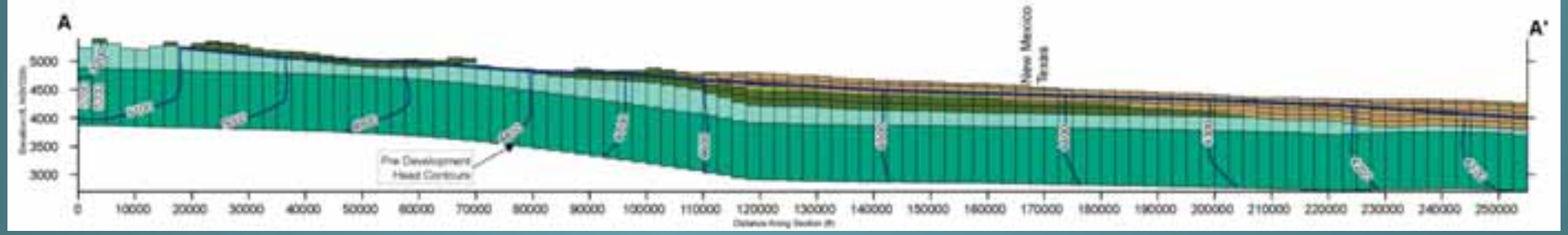
3) Results look like Hydrogeology.



HYDROGEOLOGIC SOLIDS MODEL OUTCROP



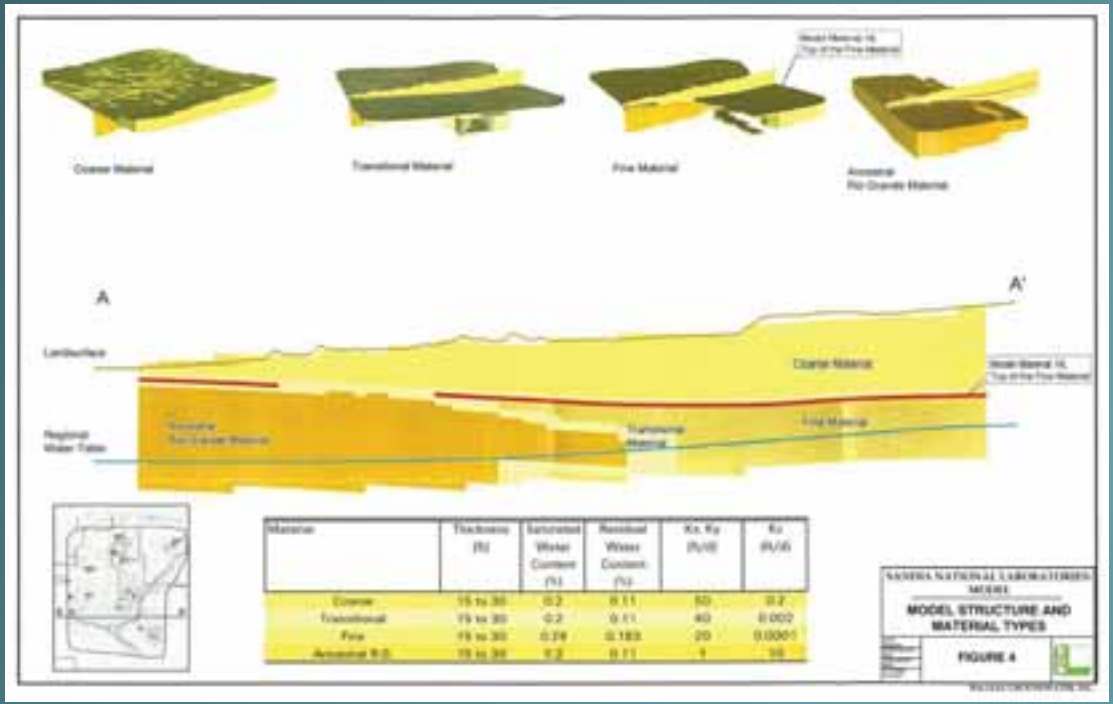
HYDROGEOLOGIC SOLIDS MODEL BEDROCK SUBCROP

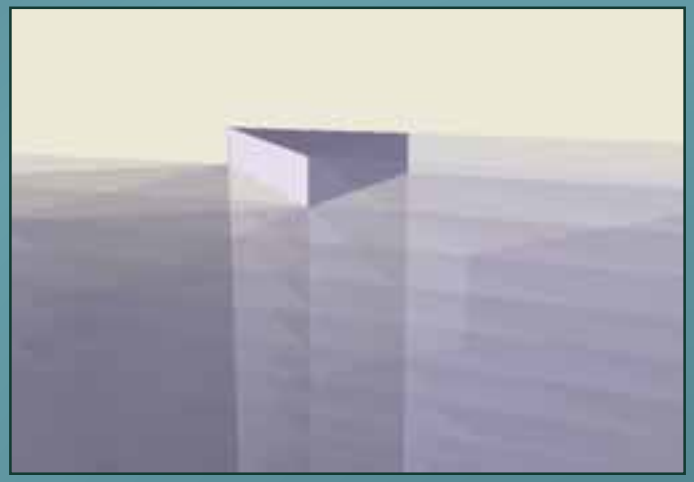


What is MODFLOW-USG?

"A new version of MODFLOW, called MODFLOW–USG (for UnStructured Grid), was developed to support a wide variety of structured and unstructured grid types, including nested grids and grids based on prismatic triangles, rectangles, hexagons, and other cell shapes. Flexibility in grid design can be used to focus resolution along rivers and around wells, for example, or to subdiscretize individual layers to better represent hydrostratigraphic units."

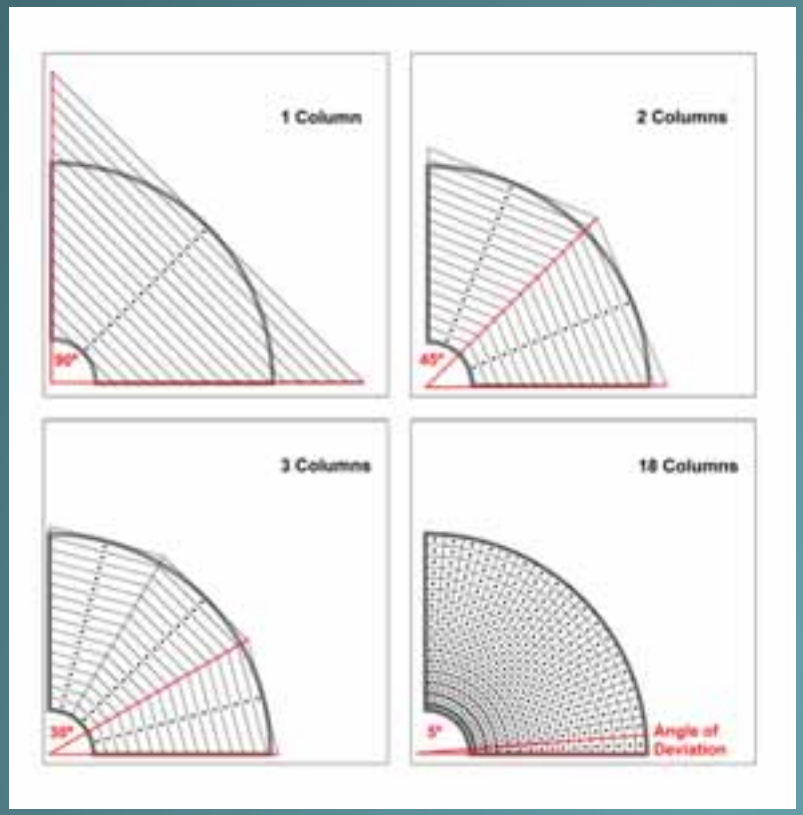
Background: Finite Element Models in ArcGIS





Shape	EL	NODE1	NODE2	NODE3	NODE4	NODE5	NODE6	MAT	X	Y	Z	LAY	TOP	MAT_D	RECHAR
MultiPatch M	137	907	1129	1133	713	902	906	8	1560	1471	5331.	9		Coarse	
MultiPatch M	137	907	1133	1135	713	906	908	8	1559	1471	5331.	9		Coarse	
MultiPatch M	137	921	907	1135	722	713	908	8	1559	1471	5319	9		Coarse	
MultiPatch M	137	908	1136	1137	714	909	910	7	1558	1472	5362	7		Coarse	
MultiPatch M	137	1136	908	1131	909	714	903	7	1559	1472	5367.	7		Coarse	
MultiPatch M	138	908	1137	915	714	910	717	7	1558	1471	5354.	7		Coarse	
MultiPatch M	138	1138	909	1139	911	715	912	6	1558	1472	5399.	6	top	Coarse	arroyo
MultiPatch M	138	909	1138	910	715	911	716	6	1558	1472	5394.	6	top	Coarse	arroyo

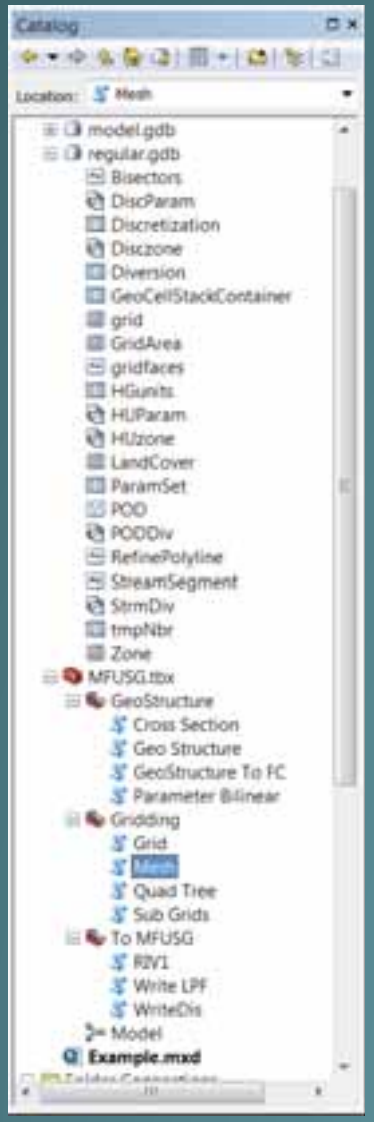
Background: Integrated Finite Difference in MODFLOW-88 and ArcGIS

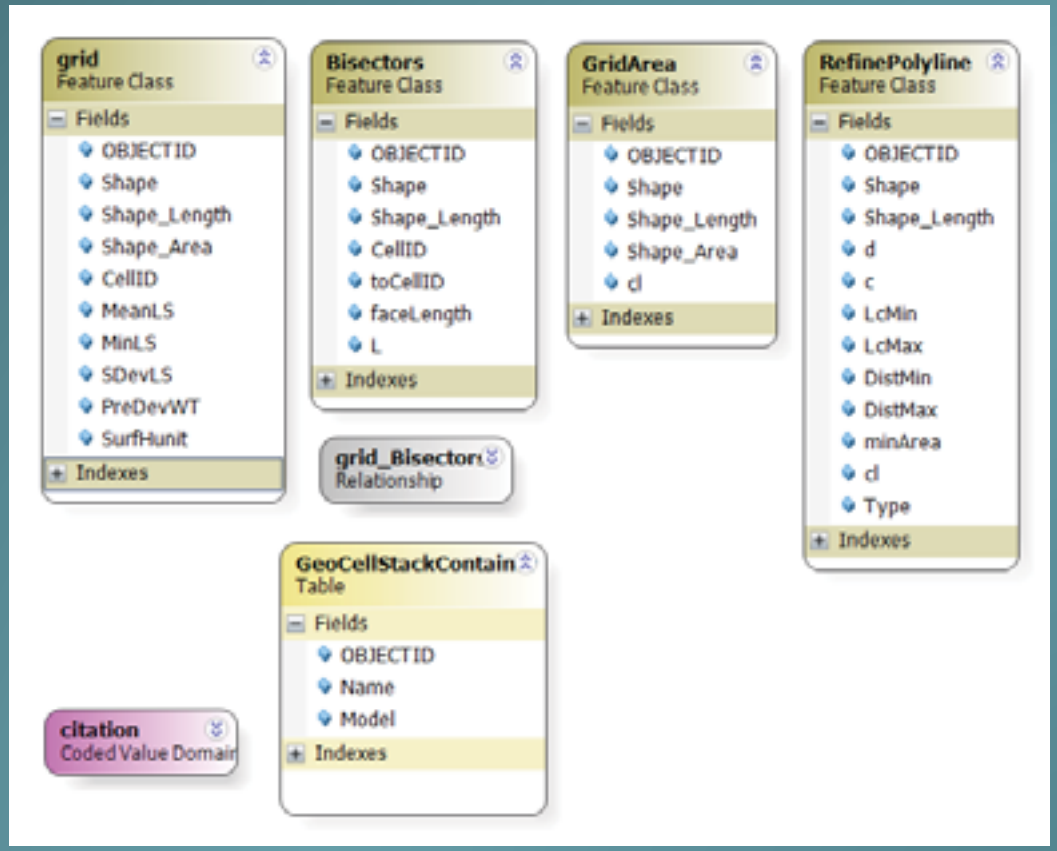


```
Columns = iCols
Rows = iRows

iID = 0
iRowCol = (iRows + 1) * (iCols + 1)

For iELM = 1 To (iRows * iCols) * iLayers
        lnode = lnode + 1
        Dim pElm As clsElement
        Set pElm = New clsElement
        pElm.CreateQuadElement iELM, 8, _
                               lnode, lnode + 1, _
                               lnode + iCols + 2, lnode + iCols + 1, _
                               lnode + iRowCol, lnode + iRowCol + 1, _
                               lnode + iCols + iRowCol + 2, lnode + iCols +
        ElementList.Add pElm
        If (iELM Mod iCols) = 0 Then lnode = lnode + 1
Next iELM

' Create Nodes
Dim dVec As IVector3D
Dim dRay As IRay

dCurrZ = 10
Dim dRadiusDelta As Double
dRadiusDelta = ((dDeltaR * Tan(PI / (2 * (iRow(iCols) + 1)))) * 2 + dDeltaR *
Debug.Print dRadiusDelta
For iL = 1 To iLayers + 1
  dRadius = dRadius1 + (iRows * dDeltaR)
  For iR = 1 To iRows + 1
  dAng = dCircum
    For iC = 1 To iCols + 1
      iIndex = iIndex + 1

      Set dVec = CreateVector3DPolar(dAng, 0#, 1#)
      Set dRay = CreateRay(pOrigin, dVec)

      Dim pNode As clsNode
      Set pNode = New clsNode
      Dim NodePnt As IPoint
      Dim nn As Double
      Dim naides As Double
      naides = ((2 * PI) / dDeltaC)
      nn = 0.5 / (Cos((PI / naides)) / (2 * dRadius))
      Set NodePnt = dRay.GetPointAtDistance(nn)
      NodePnt.z = dCurrZ

      pNode.CreateNode2 iIndex, NodePnt
      NodeList.Add pNode
      dAng = dAng + dDeltaC
    Next
    dRadius = dRadius - dDeltaR
  Next
  dCurrZ = dCurrZ - dDeltaZ
Next iL
```

Romero, D.M., and Silver S.E., 2006, Grid cell distortion and MODFLOW's integrated finite-difference numerical solution: Ground Water v. 44, p. 797–802.

Implementation: Toolset and Schema





Implementation: Code (ArcPy + Python)

geocell.py - C:\GoogleDrive\USGTools\Tools\geocell.py

File Edit Format Run Options Windows Help

```
class GeoCellStackContainer:
    def __init__(self):
        self.GeoStacks={}
        self.layerCount={}
        self.cellID2stack={}
    def __iter__(self):
        self.n1 = 0
        self.n2 = 0
        return self
    def addStack(self,GeoStack):
        self.GeoStacks[GeoStack.stackID]=GeoStack
    def numberByKIJ(self,startNumber):
        n=startNumber
        for nstack in self.GeoStacks:
          gstack = self.GeoStacks[nstack]
          for geoCell in gstack.stack:
           geoCell.CreateID(n)
           n += 1
    def numberByIJK(self,startNumber,maxLayer):
        n=startNumber
        self.layerCount = {}
        for lay in range(1,maxLayer+1):
          if not lay in self.layerCount: self.layerCount[lay] = 0
          for nstack, gstack in self.GeoStacks.iteritems():
            # TBD delete empty geostack else:
            geoCell = gstack.returnLayer(lay)
            if geoCell:
              self.cellID2stack[n]=[lay,nstack]
              geoCell.CreateID(n)
              n += 1
              self.layerCount[lay] += 1
    def createTopology(self,avgFace):
        for nstack in self.GeoStacks:
          gstack = self.GeoStacks[nstack]
          stackiter = zip(gstack.adjacent,gstack.faceWidths,gstack.branchLengths
          lastCell = None
          for geoCell in gstack.stack:
            for n,w,l in stackiter:
              if n in self.GeoStacks:
                adjCell=self.GeoStacks[n].returnLayer(geoCell.layer)
```

Ln: 383 Col: 39

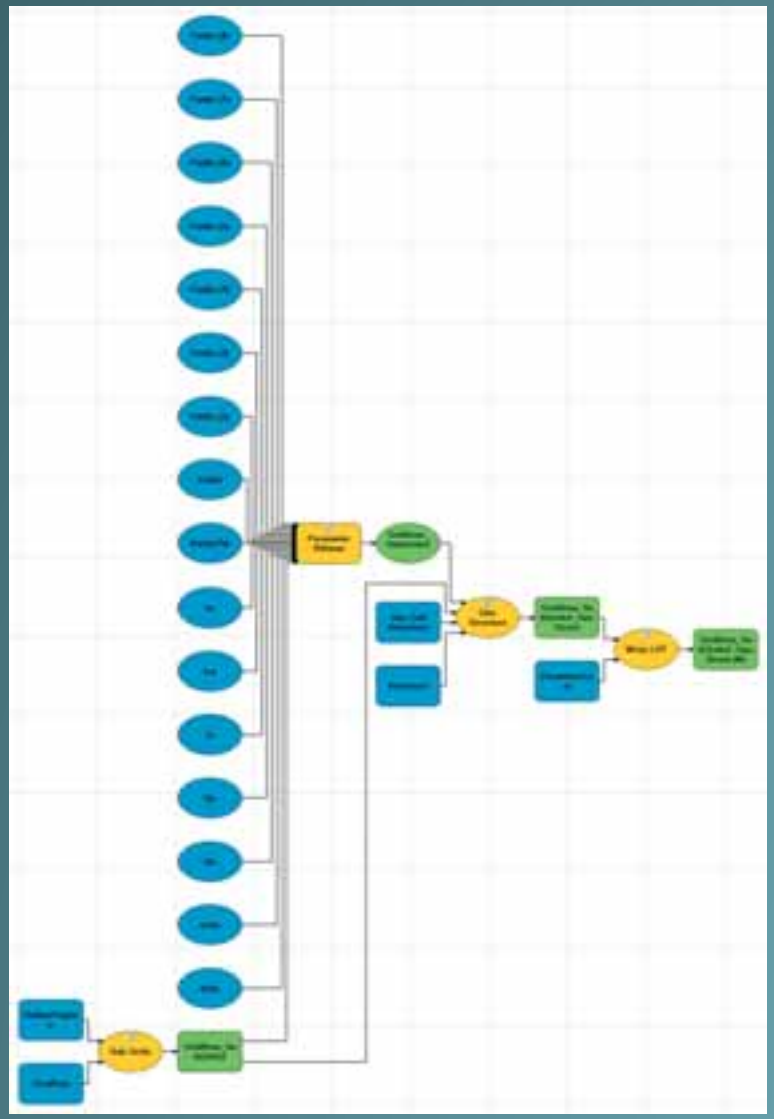
*makeGrid.py - C:\GoogleDrive\USGTools\Tools\makeGrid.p...

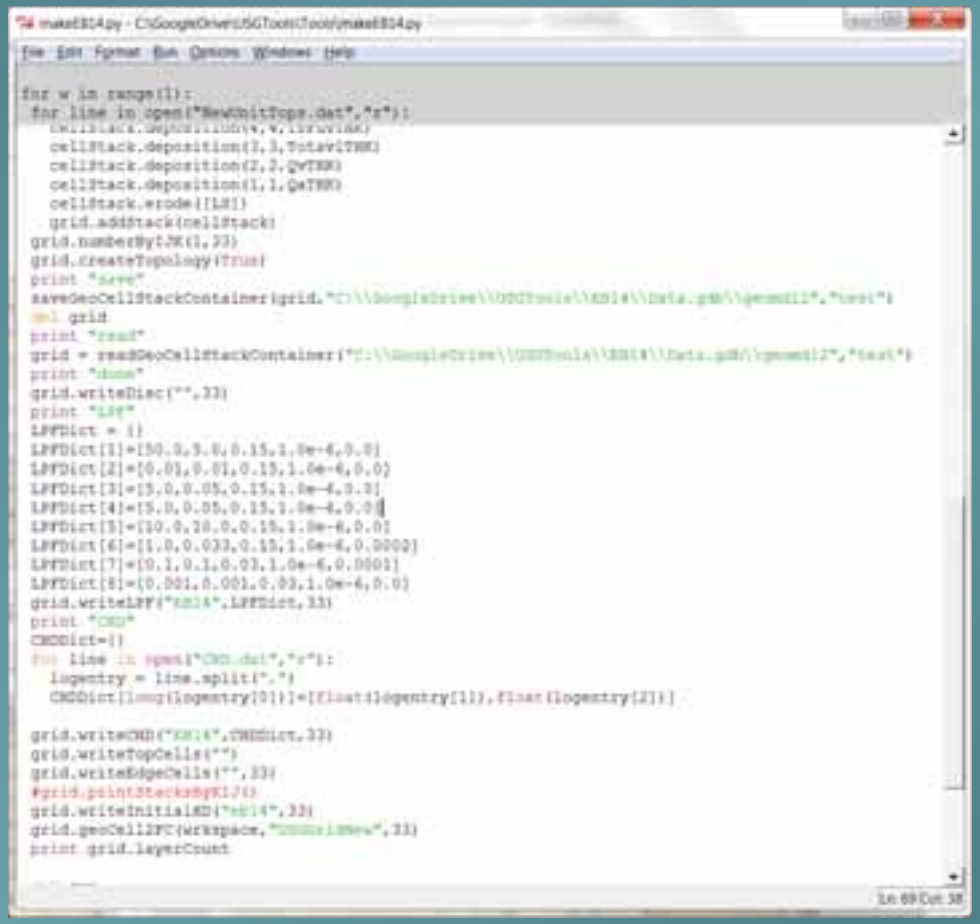
File Edit Format Run Options Windows Help

```
from math import *
import numpy as np
from geocell import *
import subprocess
import tempfile

n = NodeContainer()
cellID = 0
def distance(p1,p2):
        return ((p1.X-p2.X)**2+(p1.Y-p2.Y)**2)**0.5

def midPoint(p1,p2):
        return arcpy.Point((p1.X+p2.X)/2,(p1.Y+p2.Y)

def addtoIndex(poly,cellID):
   n.addPolygon(cellID)
   for g in range(0,poly.getPart(0).count-1):
     n.getPolygonSegment(poly.getPart(0)[g],poly.get

def getL(A,B,P):
     C1 = midPoint(A,B)
     C2 = perpendicularToPointOnLine(A,B,C1)
     Cn = closestPoint(C1,C2,P)
     return distance(C1,Cn)

def getlpolyline(A,B,P,spatial_reference):
     C1 = midPoint(A,B)
     C2 = perpendicularToPointOnLine(A,B,C1)
     Cn = closestPoint(C1,C2,P)
     xsarray = arcpy.Array()
     xsarray.add(C1); xsarray.add(Cn)
     return arcpy.Polyline(xsarray,spatial_reference

def closestPoint(A,B,P):
  ap = [P.X - A.X, P.Y - A.Y]
  ab = [B.X - A.X, B.Y - A.Y]
  atb = ab[0]**2 + ab[1]**2
  apdotab = ap[0] * ab[0] + ap[1] * ab[1]
  t = apdotab / atb
  return arcpy.Point(A.X + ab[0] * t, A.Y + ab[1] *

def perpendicularToPointOnLine(A,B,P):
        uvr = np.dot([[0.0,-1.0],[1.0,0.0]],[[A.X-B.
        return arcpy.Point(P.X-uvr[0][0],P.Y-uvr[1][
```

Ln: 81 Col: 40

Implementation: Use





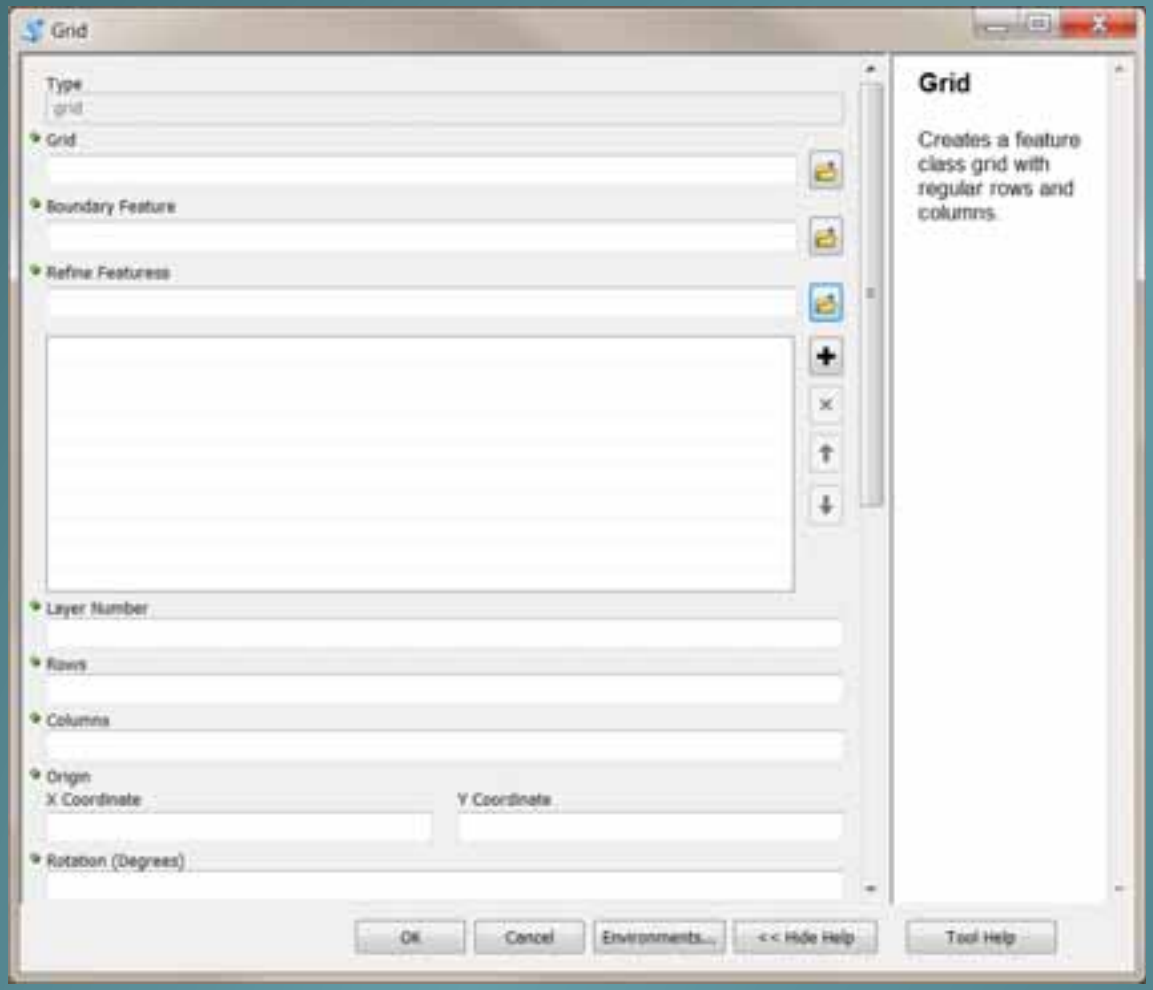
```
for w in range(1):
 for line in open("NewUnitTops.dat","r"):
   cellStack.deposition(3,3,TotalTHK)
   cellStack.deposition(2,2,QvTHK)
   cellStack.deposition(1,1,QaTHK)
   cellStack.erode([LH])
   grid.addStack(cellStack)
 grid.numberByIJK(1,33)
 grid.createTopology(True)
 print "save"
 saveGeoCellStackContainer(grid,"C:\\GoogleDrive\\USGTools\\KH14\\Data.gdb\\geomdl2","test")
 del grid
 print "read"
 grid = readGeoCellStackContainer("C:\\GoogleDrive\\USGTools\\KH14\\Data.gdb\\geomdl2","test")
 print "done"
 grid.writeDisc("",33)
 print "LPF"
 LPFDict = {}
 LPFDict[1]=[50.0,5.0,0.15,1.0e-4,0.0]
 LPFDict[2]=[0.01,0.01,0.15,1.0e-4,0.0]
 LPFDict[3]=[5.0,0.05,0.15,1.0e-4,0.0]
 LPFDict[4]=[5.0,0.05,0.15,1.0e-4,0.0]
 LPFDict[5]=[10.0,10.0,0.15,1.0e-4,0.0]
 LPFDict[6]=[1.0,0.033,0.15,1.0e-4,0.0002]
 LPFDict[7]=[0.1,0.1,0.03,1.0e-4,0.0001]
 LPFDict[8]=[0.001,0.001,0.03,1.0e-4,0.0]
 grid.writeLPF("KH14",LPFDict,33)
 print "CHD"
 CHDDict={}
 for line in open("CHD.dat","r"):
   logentry = line.split(",")
   CHDDict[long(logentry[0])]=[float(logentry[1]),float(logentry[2])]

 grid.writeCHD("KH14",CHDDict,33)
 grid.writeTopCells("")
 grid.writeEdgeCells("",33)
 #grid.printStacksByKIJ()
 grid.writeInitialHD("KH14",33)
 grid.geoCellFC(wrkspace,"[illegible]",33)
 print grid.layerCount
```

Grids and Meshes

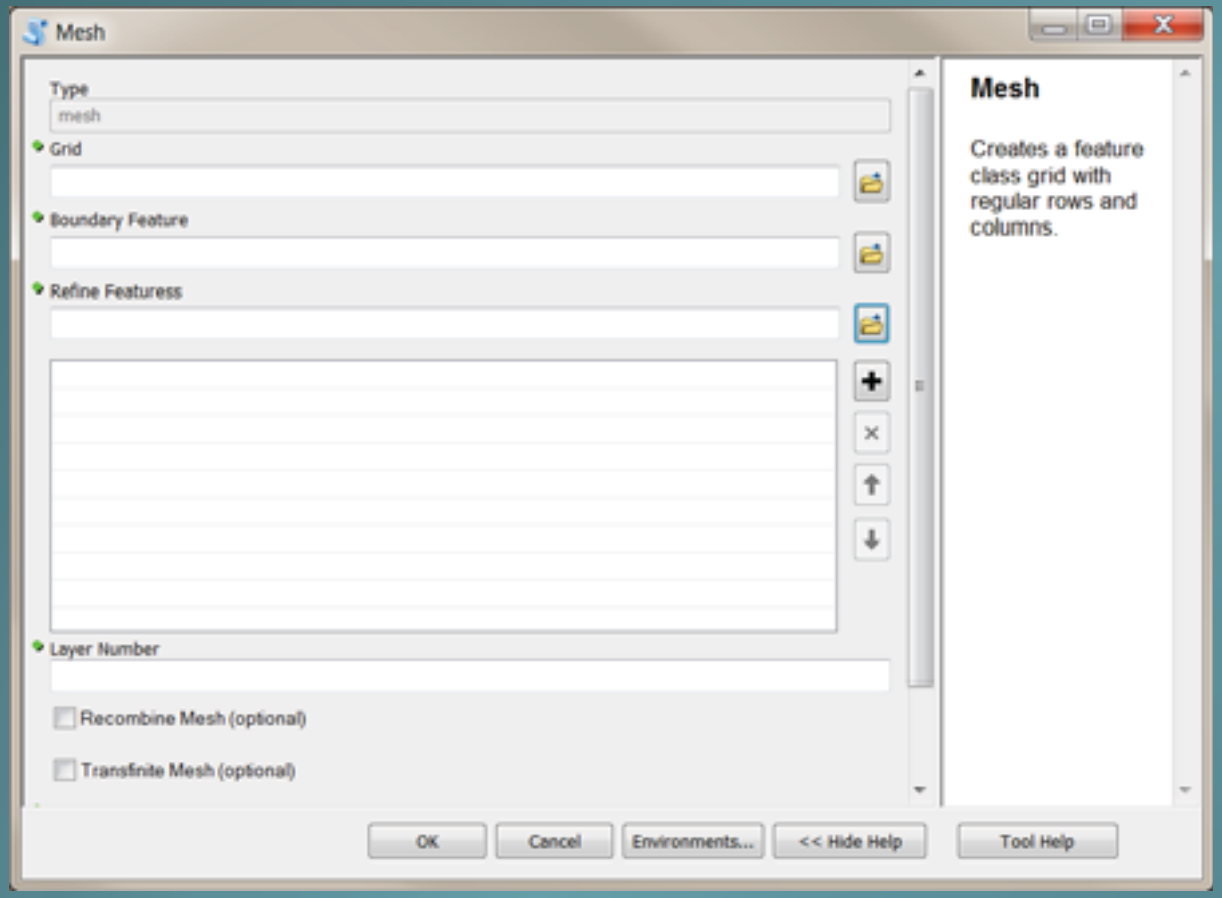
"The term "unstructured grid" simply means that the number of connections may be variable for each cell. For example, with mesh-centered triangular finite-elements, a node may be common to several elements, and this connectivity may vary for each node. This variability results in an unstructured system of equations. Similarly, in CVFD schemes the connectivity of a cell depends on the number of shared faces, which may vary for each cell."

USGS Techniques and Methods 6–A45 pg. 8

Grids and Meshes: Tools

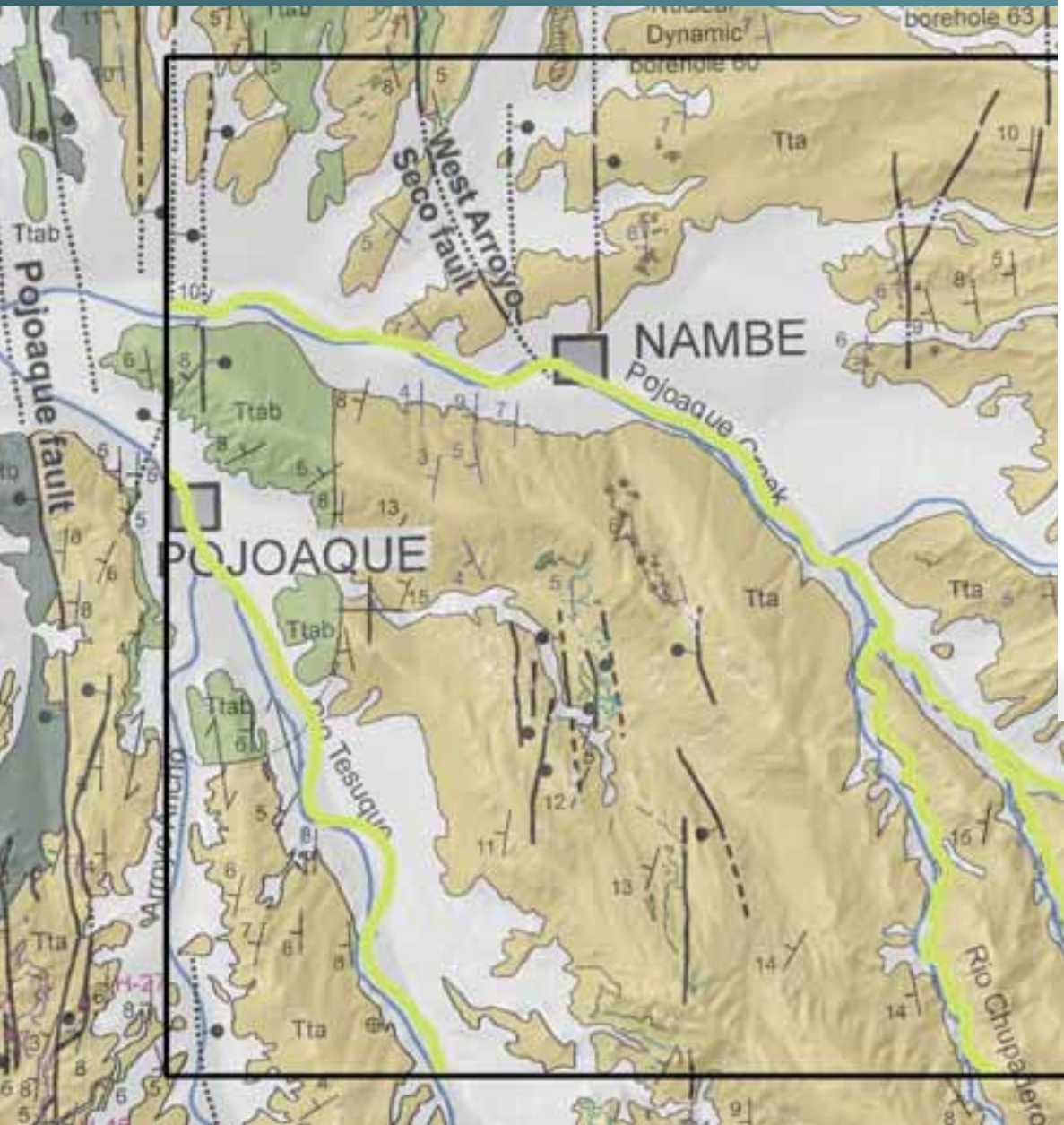


Grids and Meshes: Tools



Mesh tool utilizes Gmsh:
C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering 79(11), pp. 1309-1331, 2009

Grids and Meshes: Refinement and Boundary



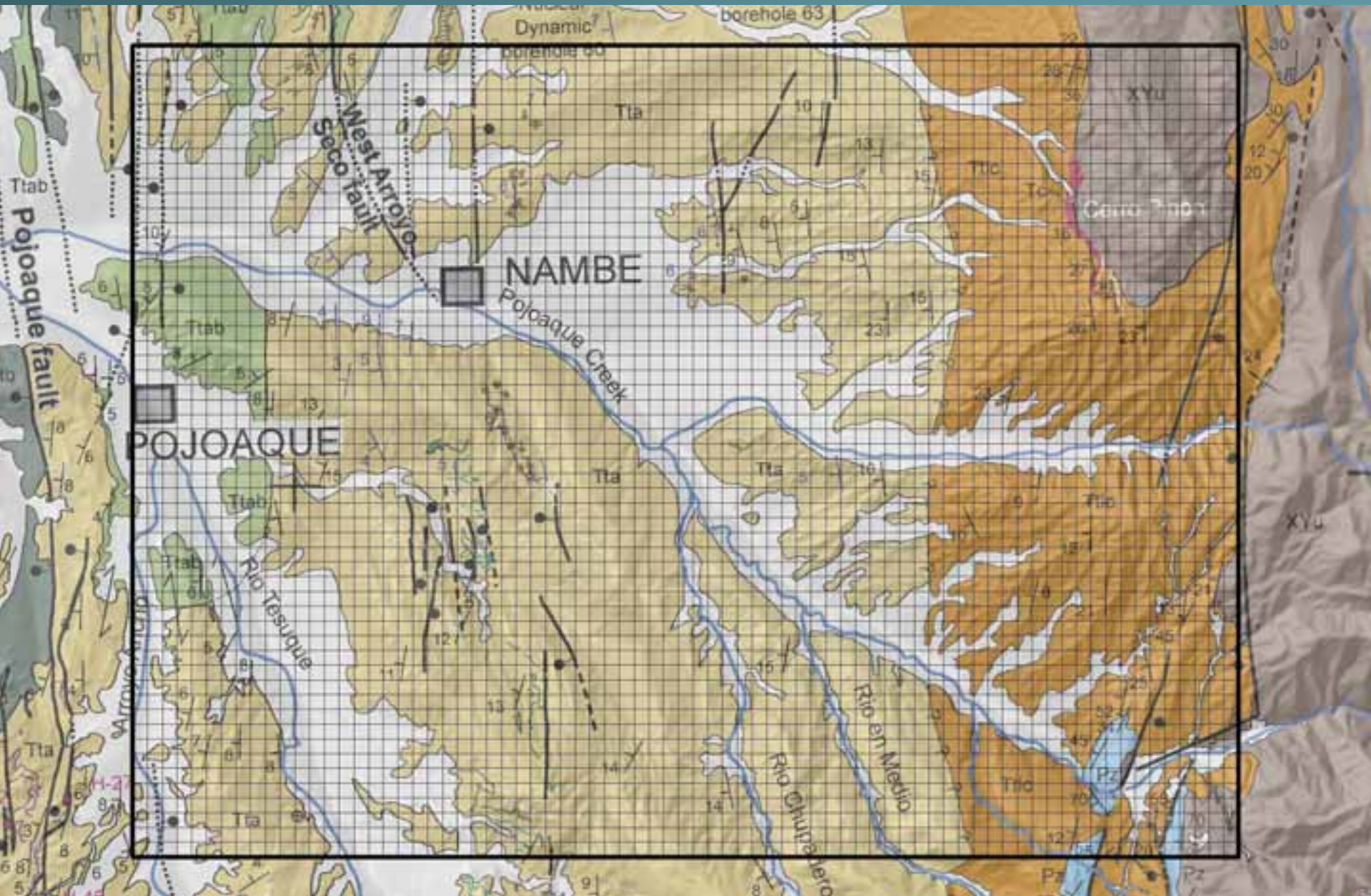
GridArea
Feature Class

- Fields
  - OBJECTID
  - Shape
  - Shape_Length
  - Shape_Area
  - cl
- Indexes

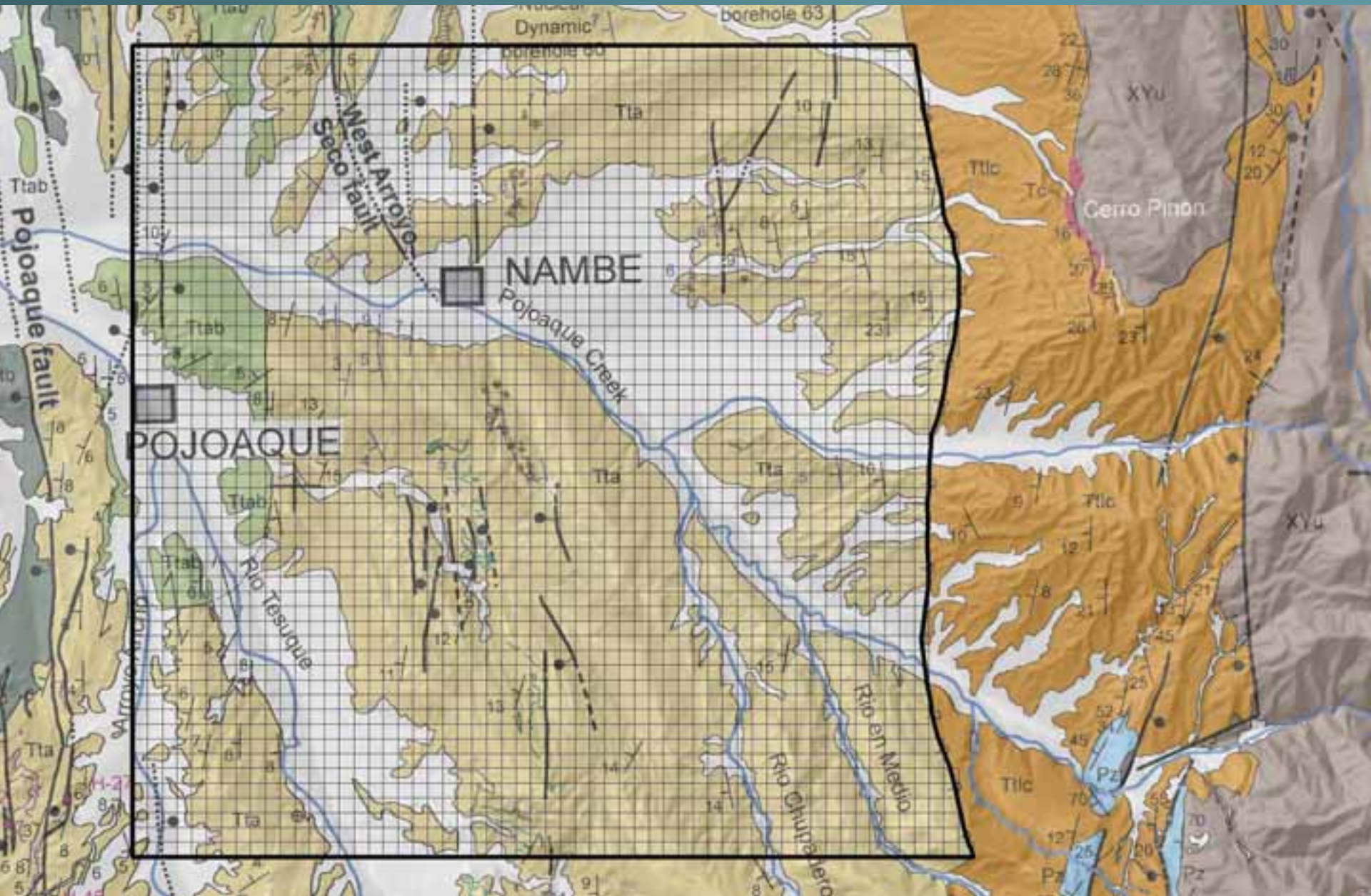
RefinePolyline
Feature Class

- Fields
  - OBJECTID
  - Shape
  - Shape_Length
  - d
  - c
  - LcMin
  - LcMax
  - DistMin
  - DistMax
  - minArea
  - cl
  - Type
- Indexes

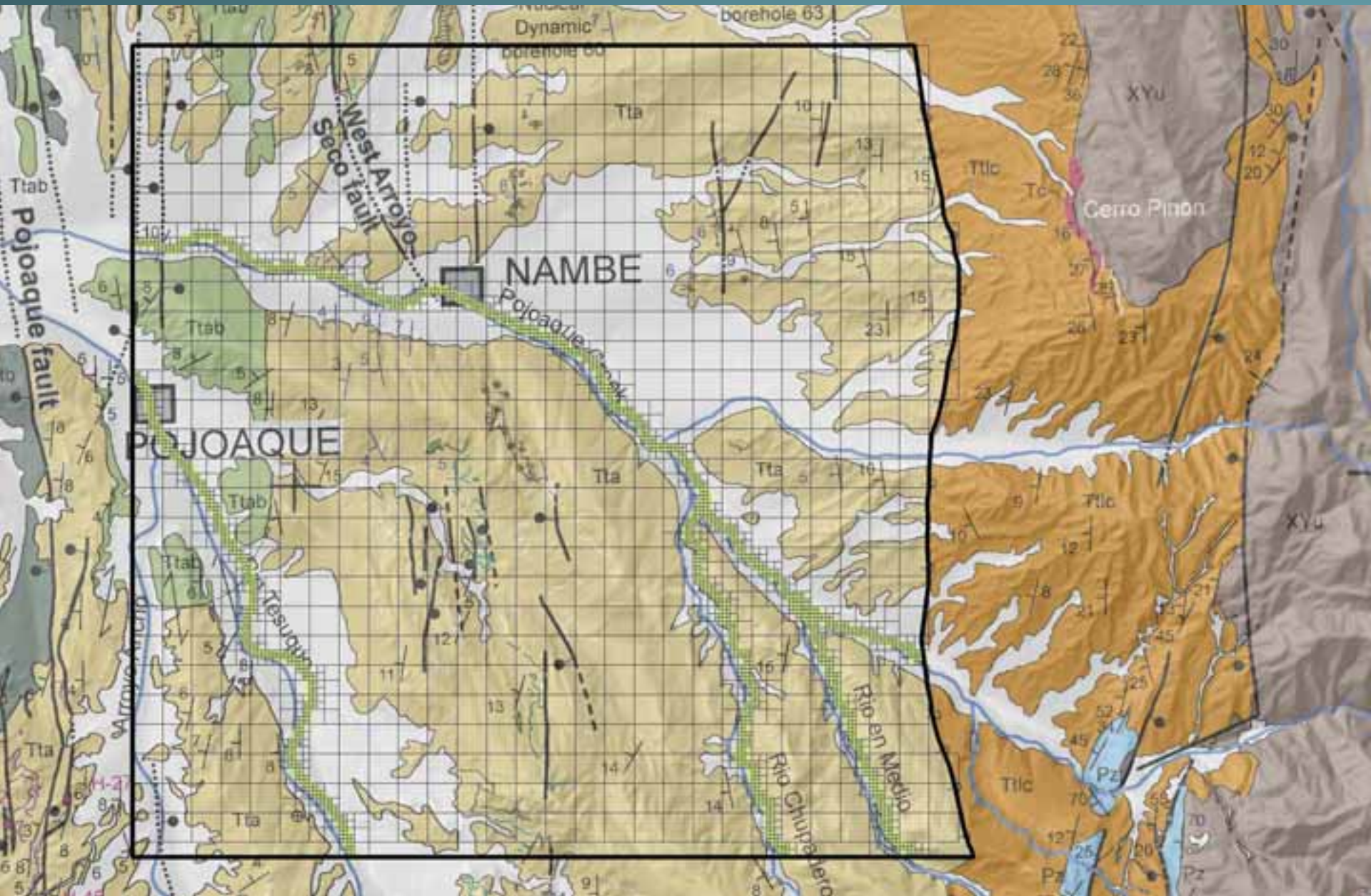
Grids and Meshes: Grid



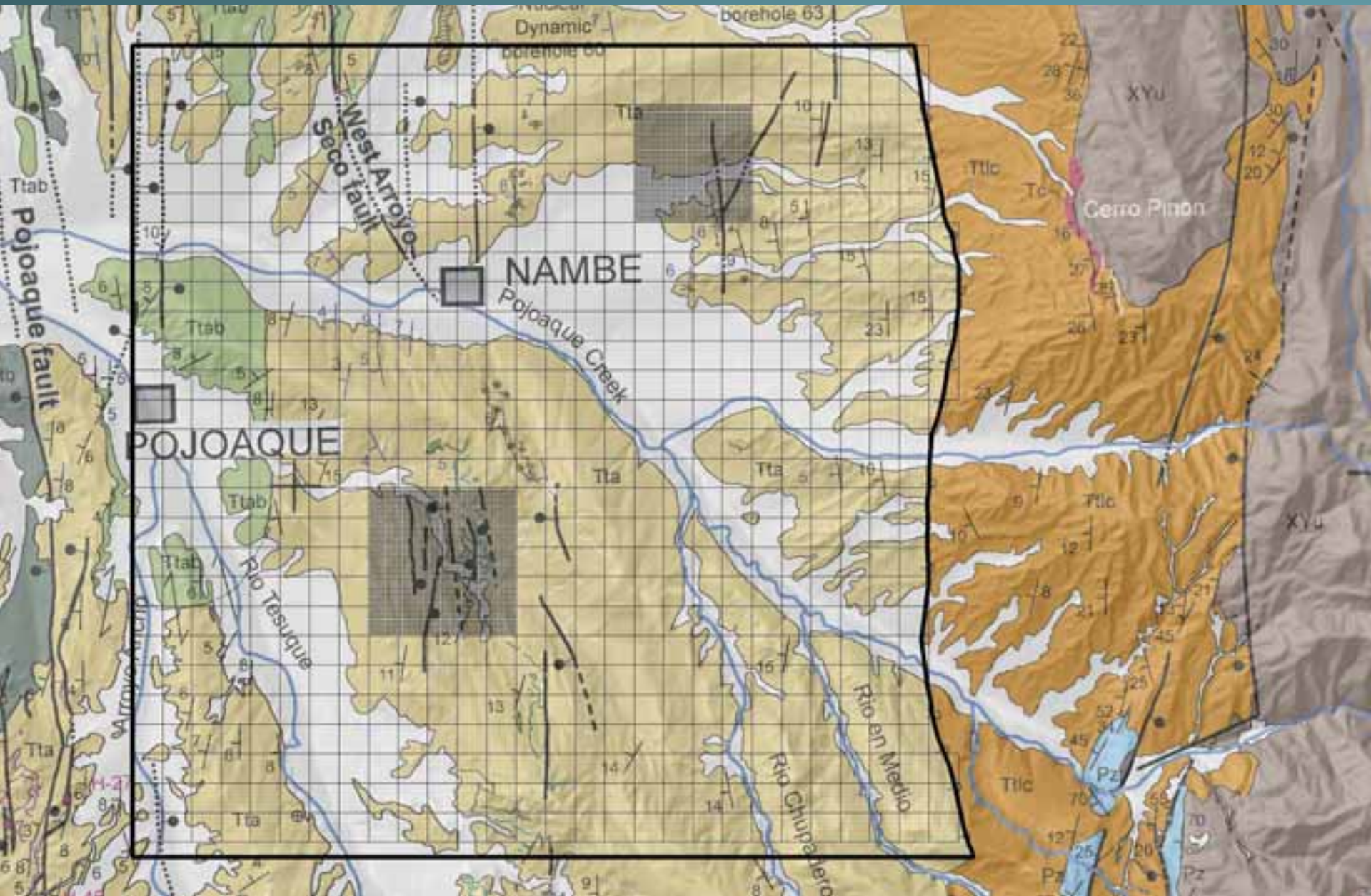
Grids and Meshes: Grid with Irregular Domain



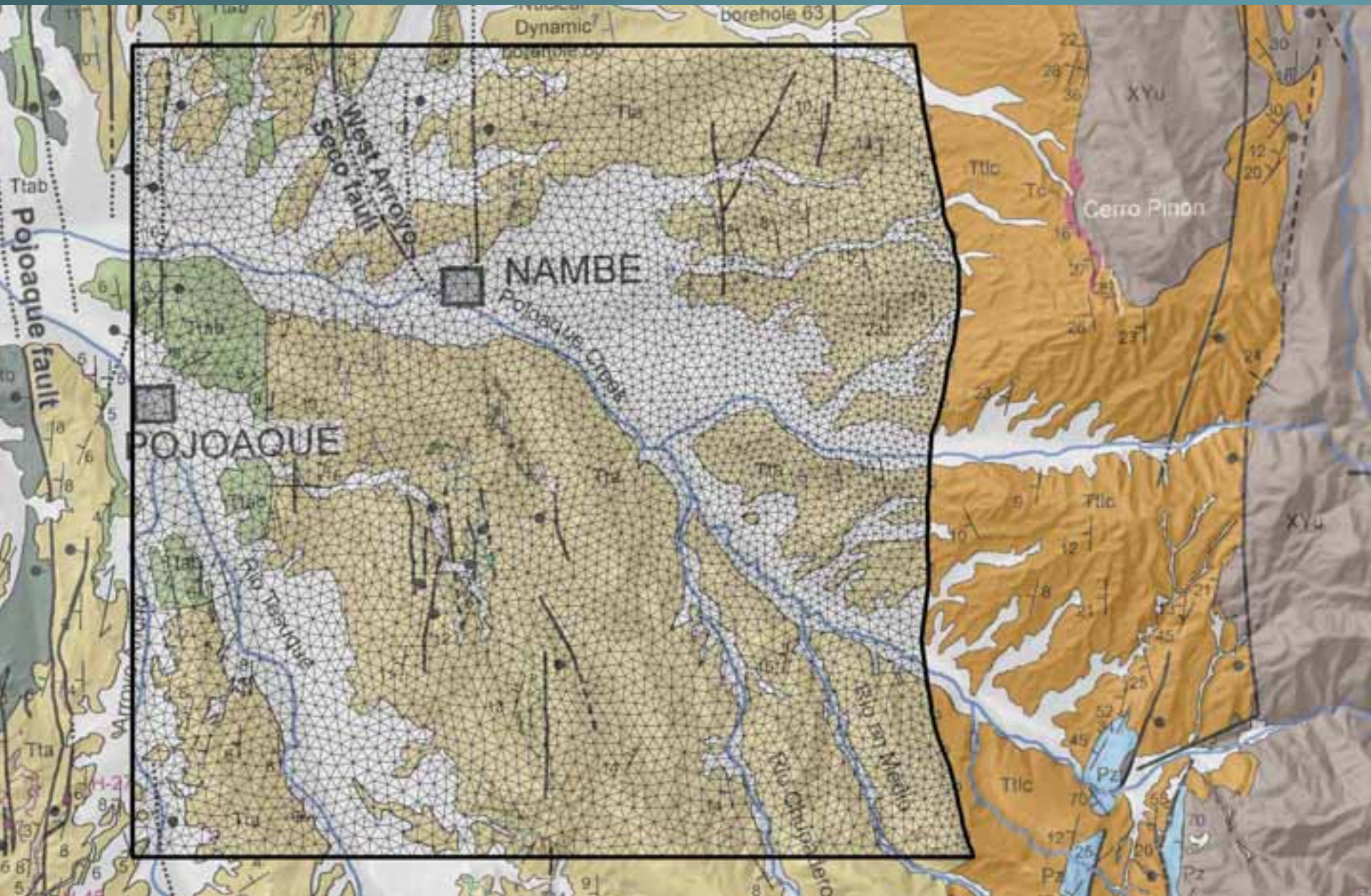
Grids and Meshes: Grid with Quad-Tree Refinement



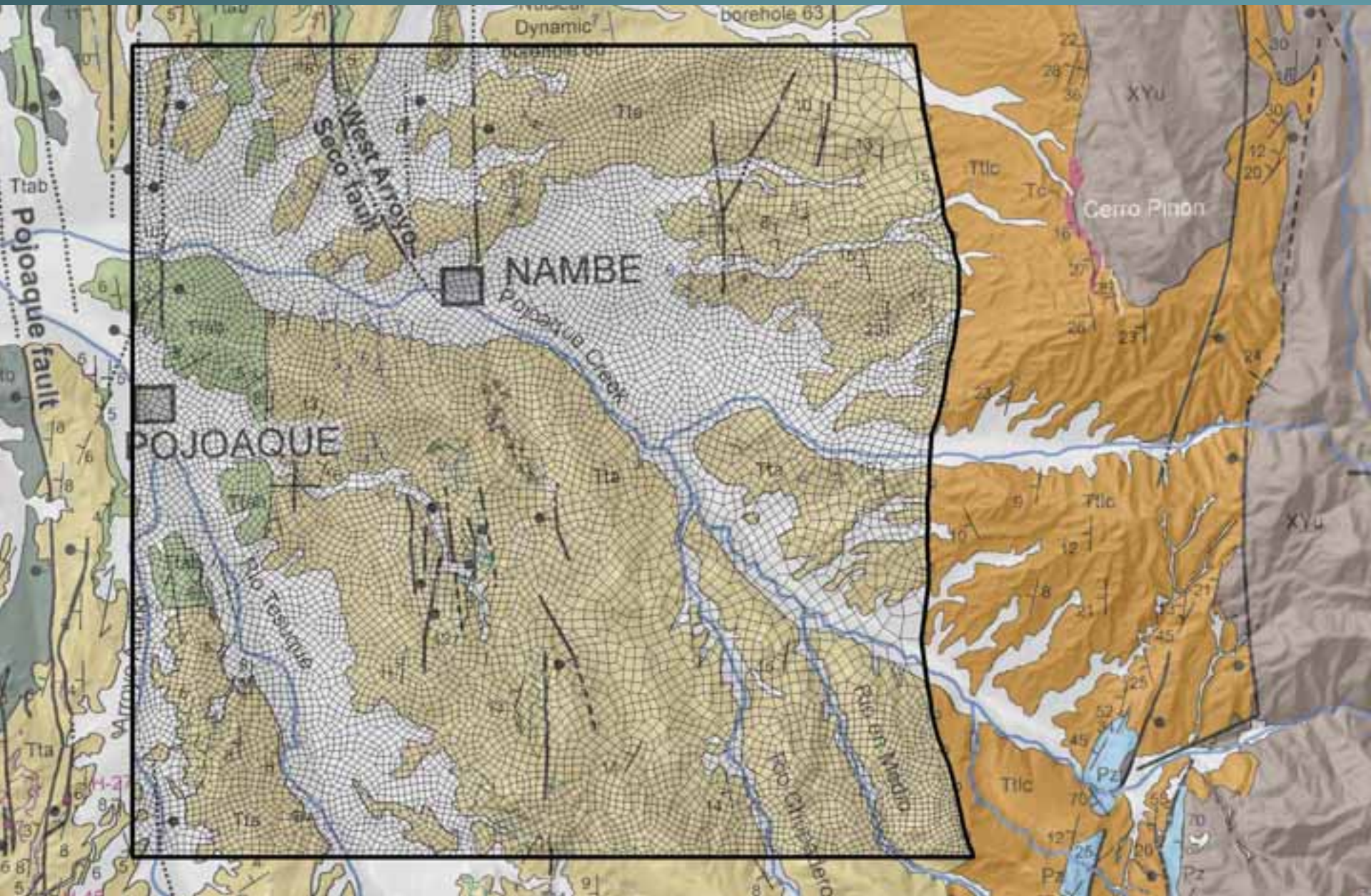
Grids and Meshes: Grid with Sub-Grids



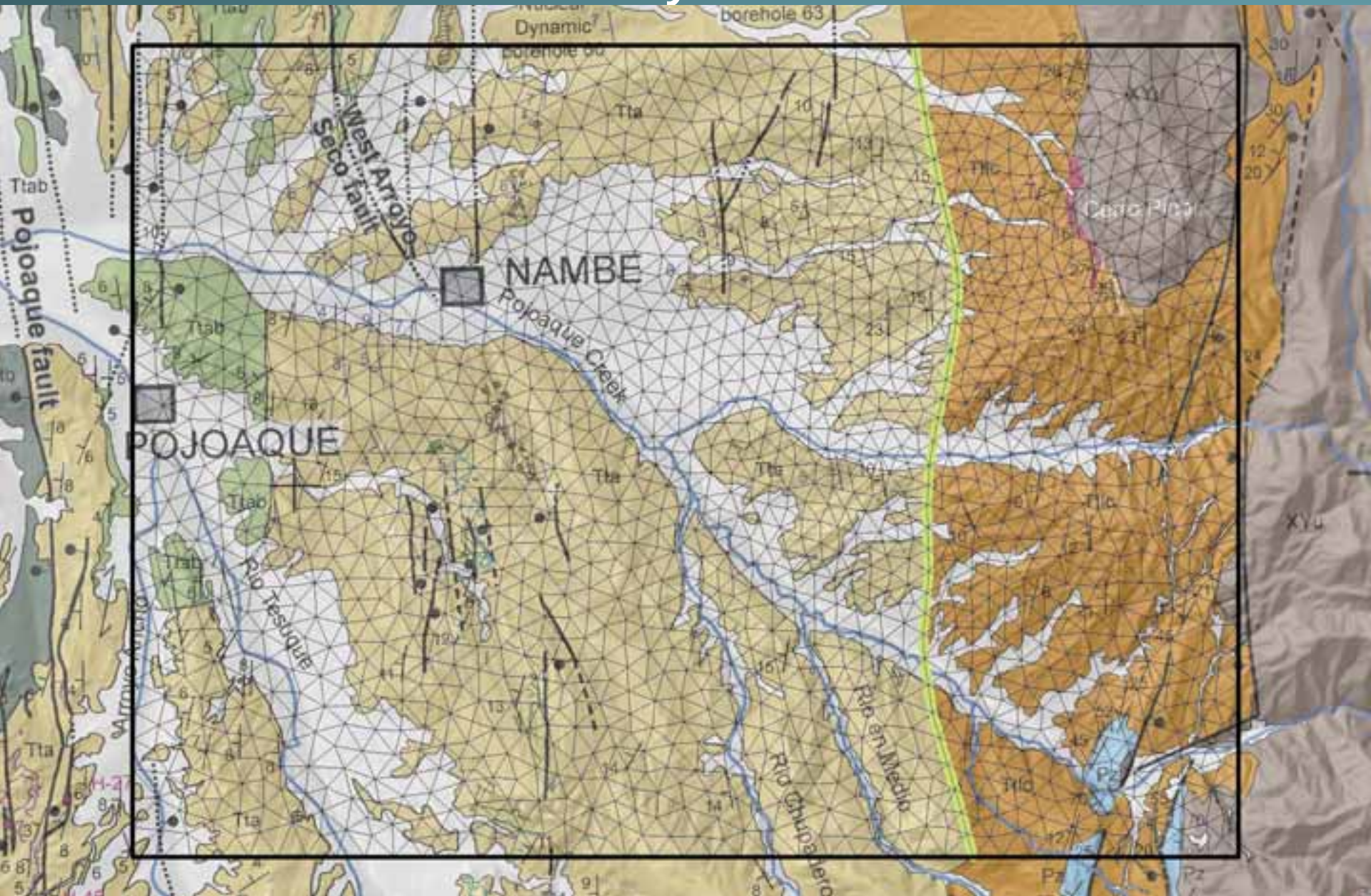
Grids and Meshes: Triangular Mesh



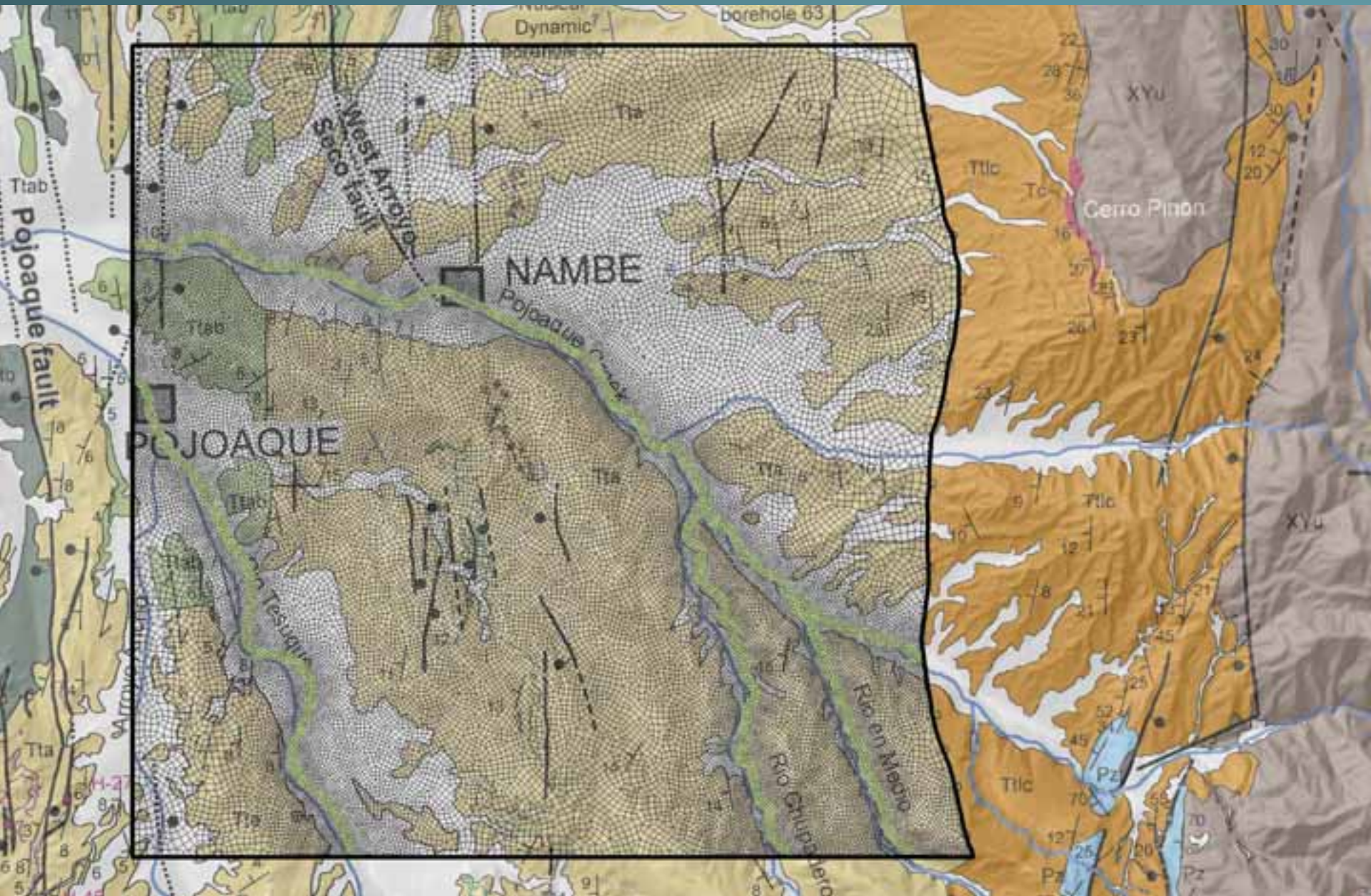
Grids and Meshes: Quadrilateral Mesh



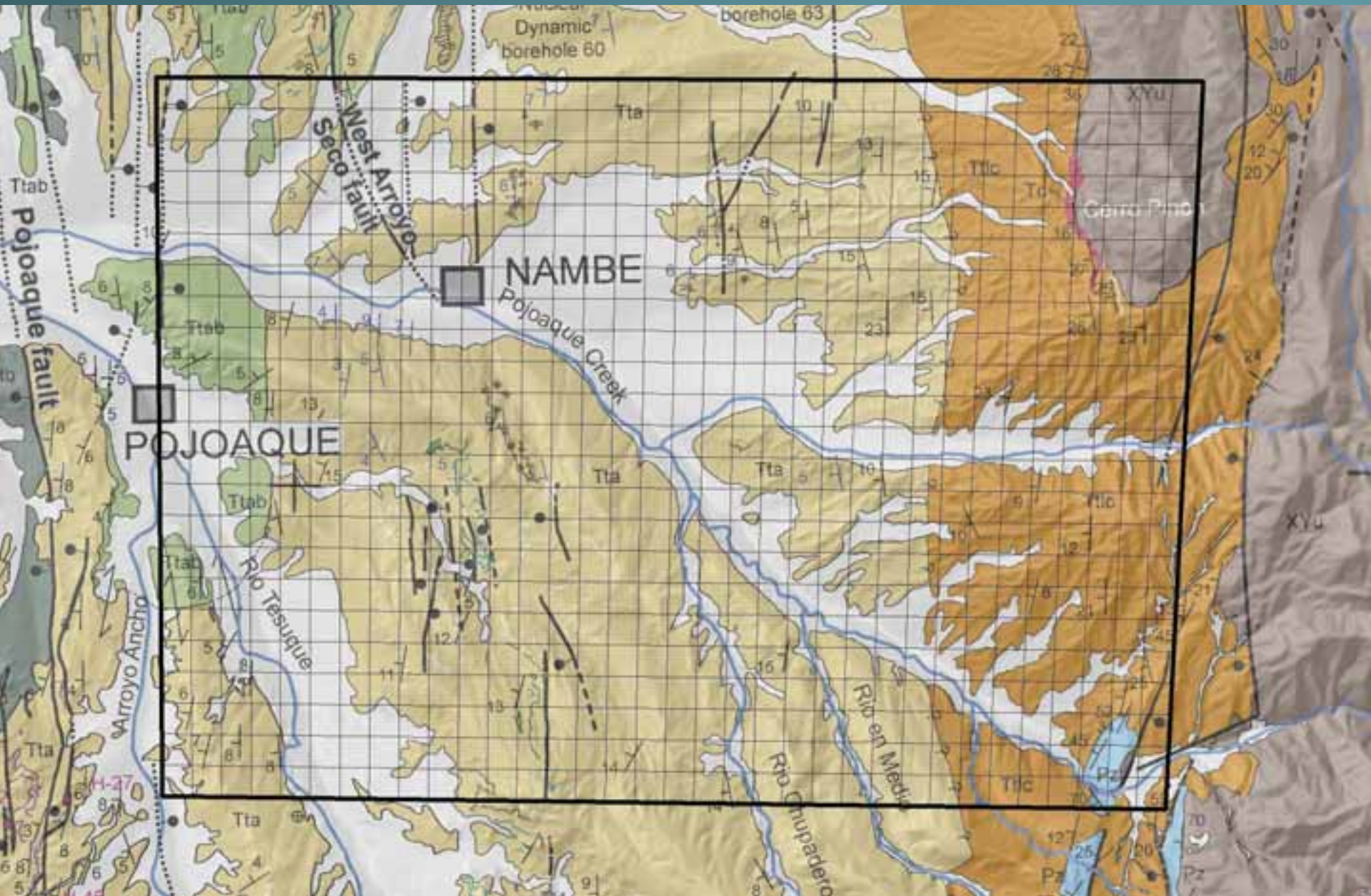
Grids and Meshes: Triangular Mesh With Embedded Polyline



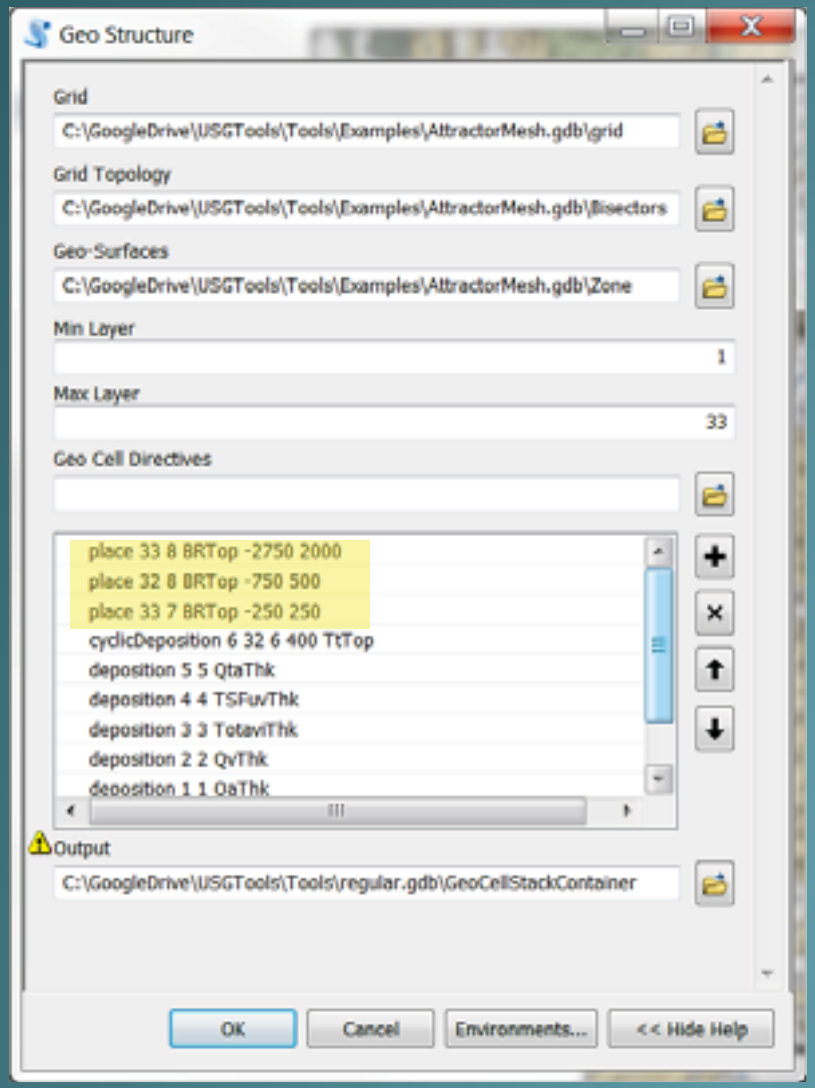
Grids and Meshes: Quadrilateral Mesh With Attractor



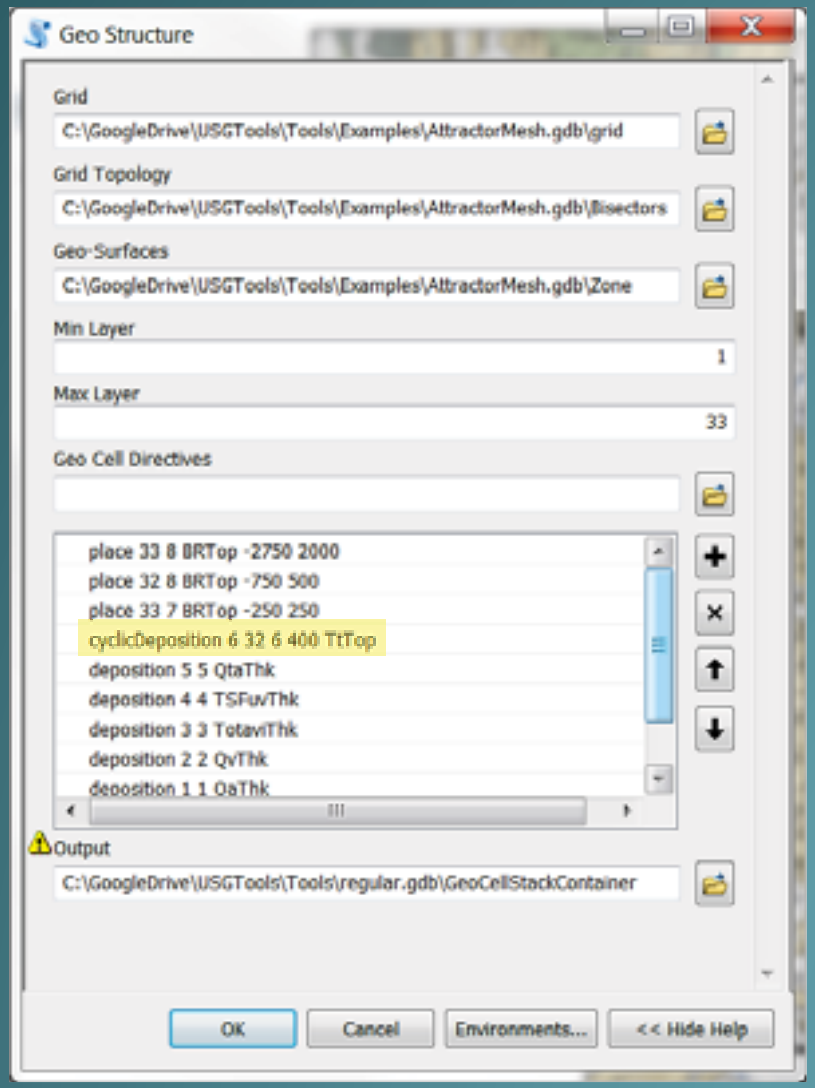
Grids and Meshes: Quadrilateral Transfinite Mesh



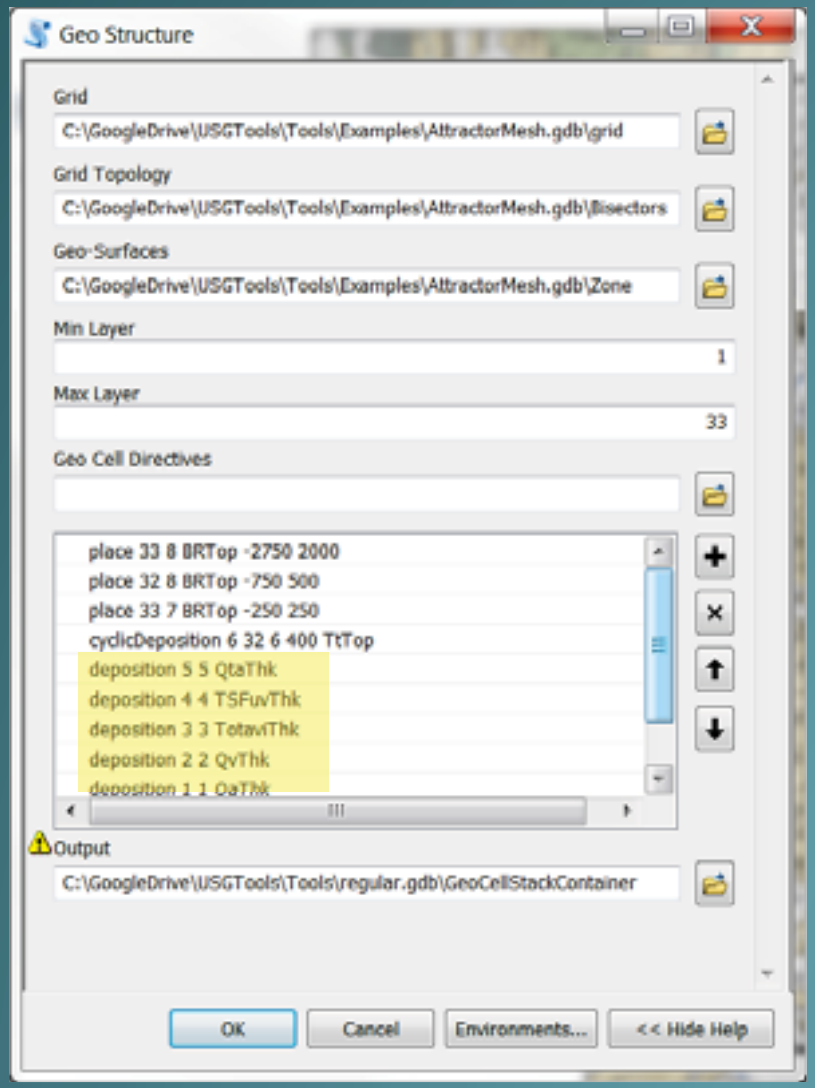
Geo-Cellular Model

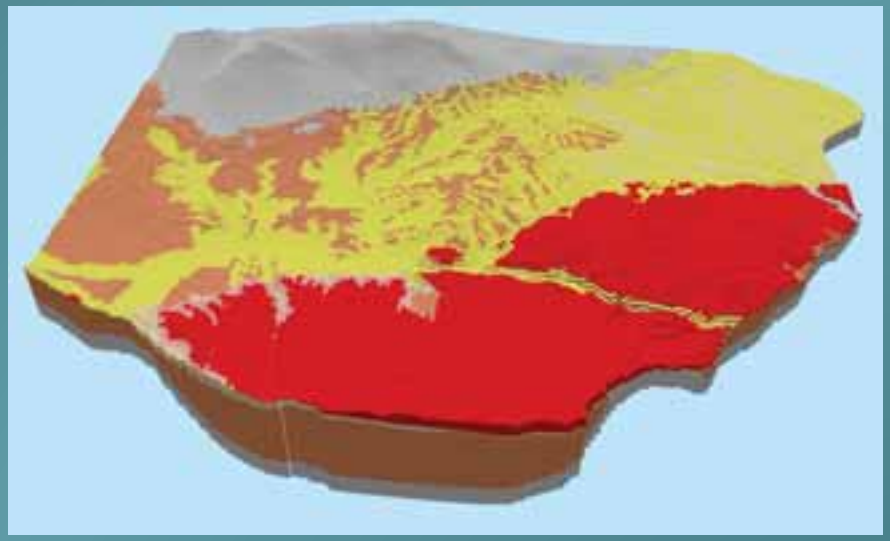


Geo-Cellular Model

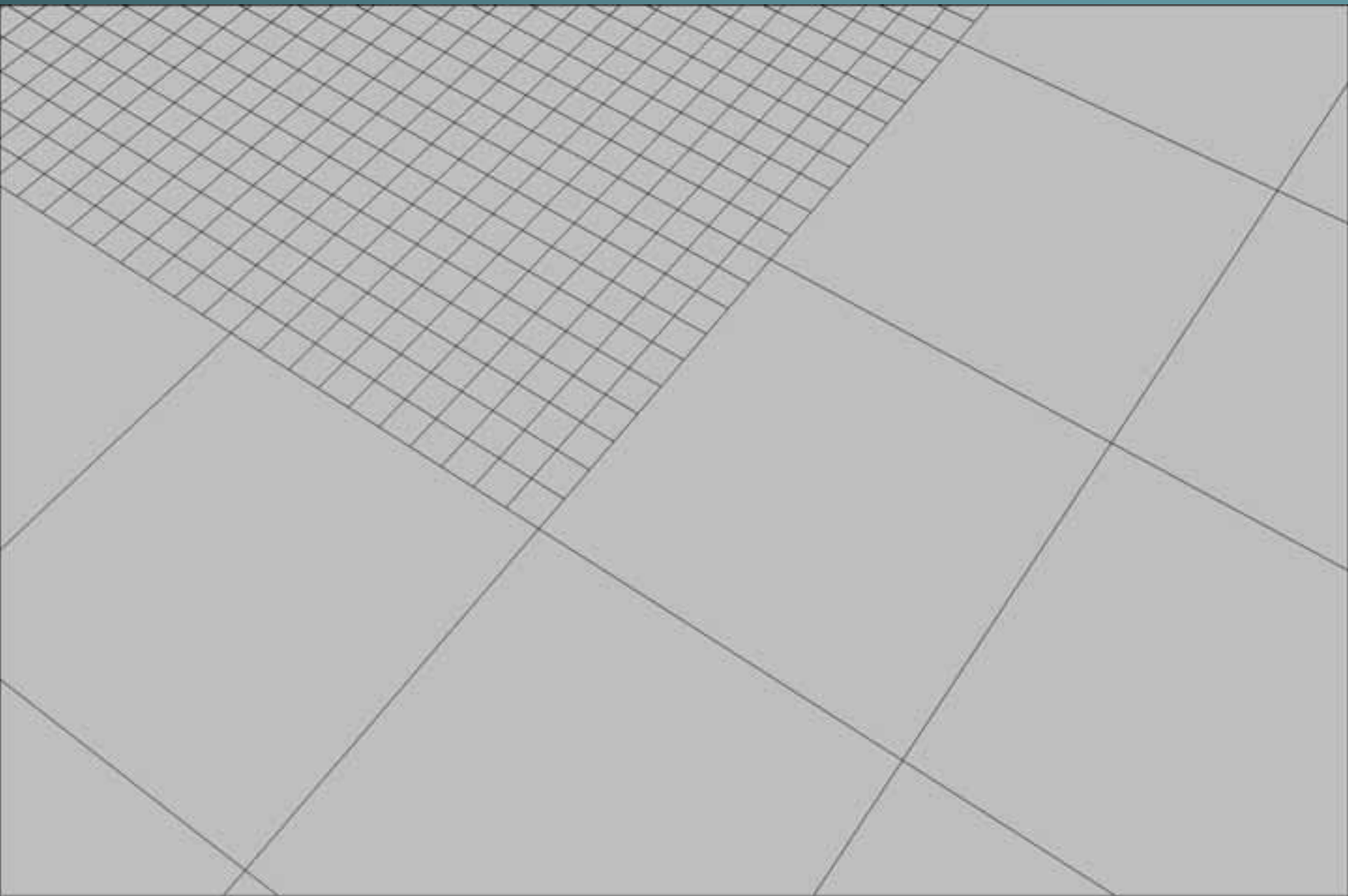


Geo-Cellular Model

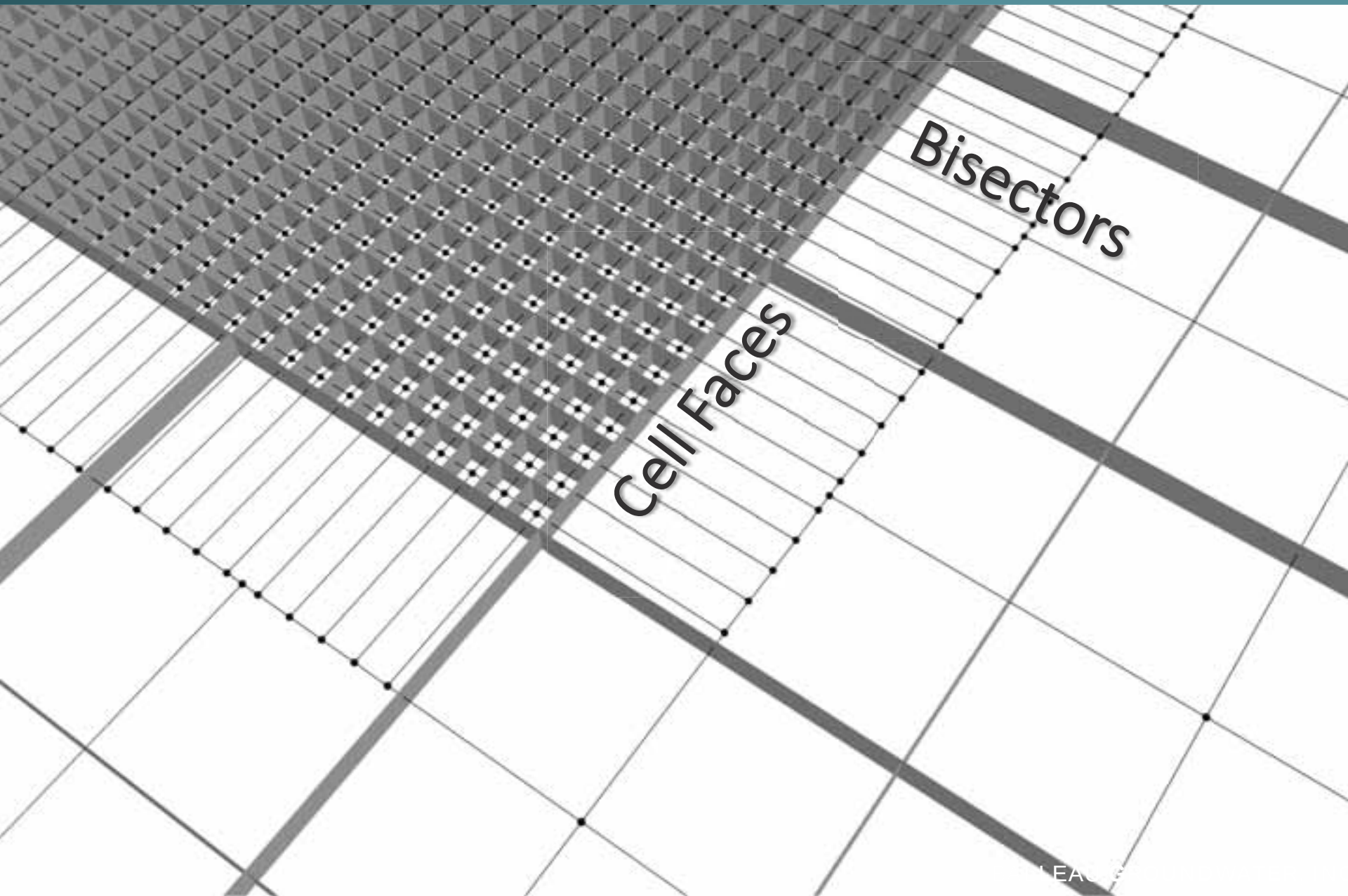




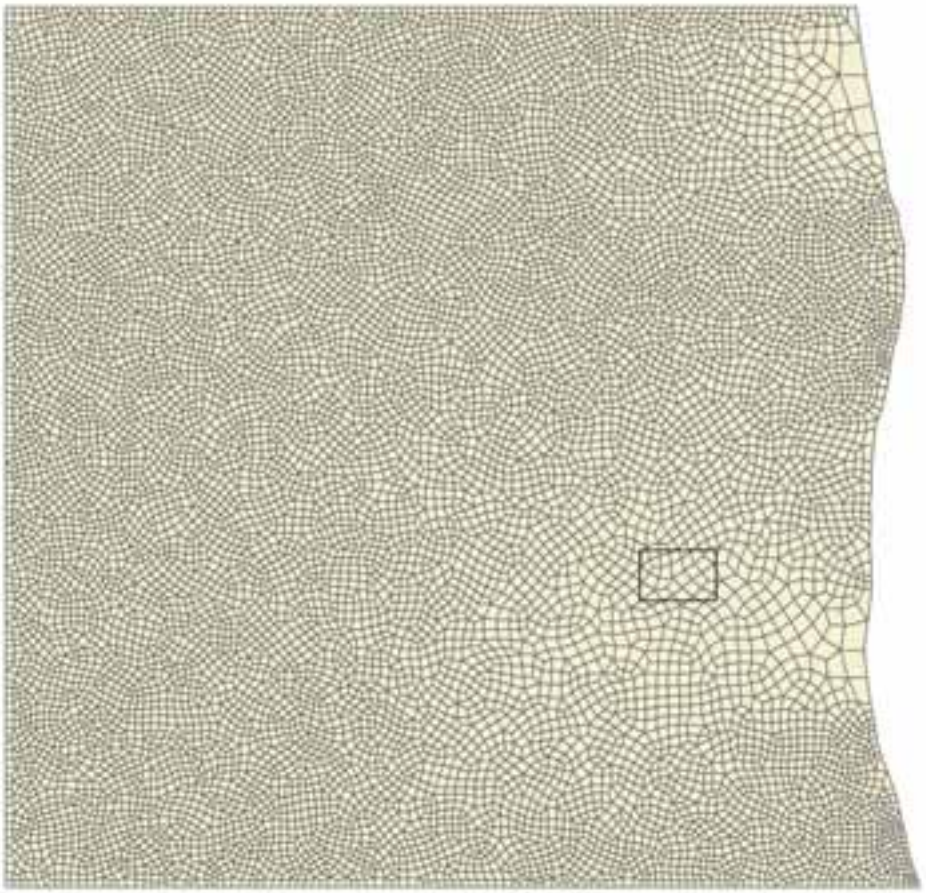
Building Grid Topology: Lateral Connections

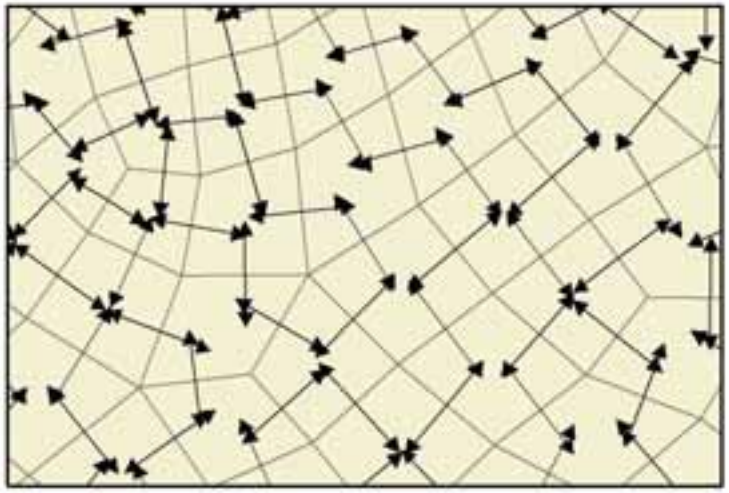


Building Grid Topology: Lateral Connections



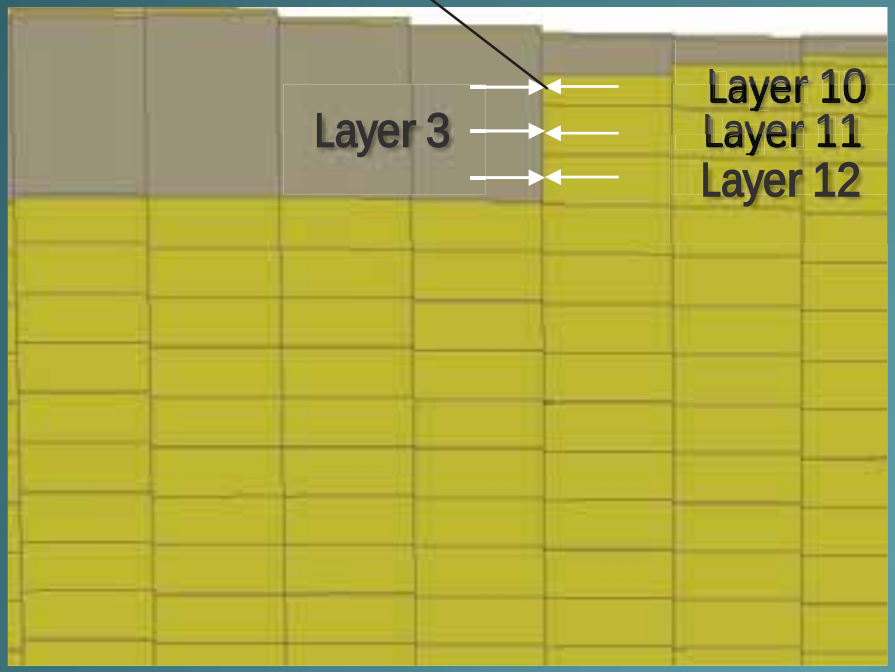
Building Grid Topology: Lateral Connections





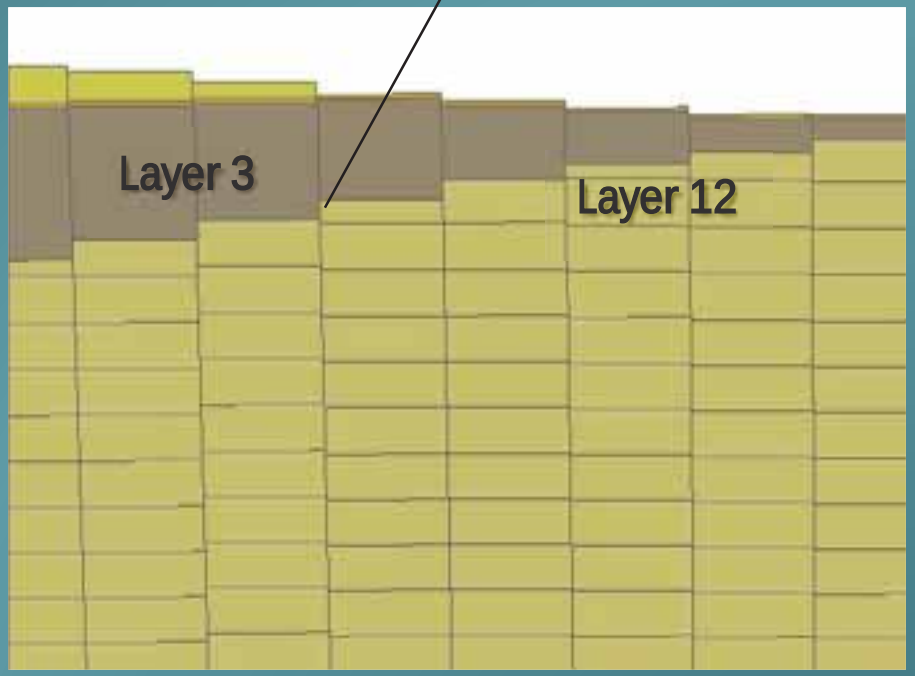
Building Grid Topology: Lateral Connections

Layers 10,11 and 12 connect to layer 3



1) Lateral Connections and flow area by geometric intersection

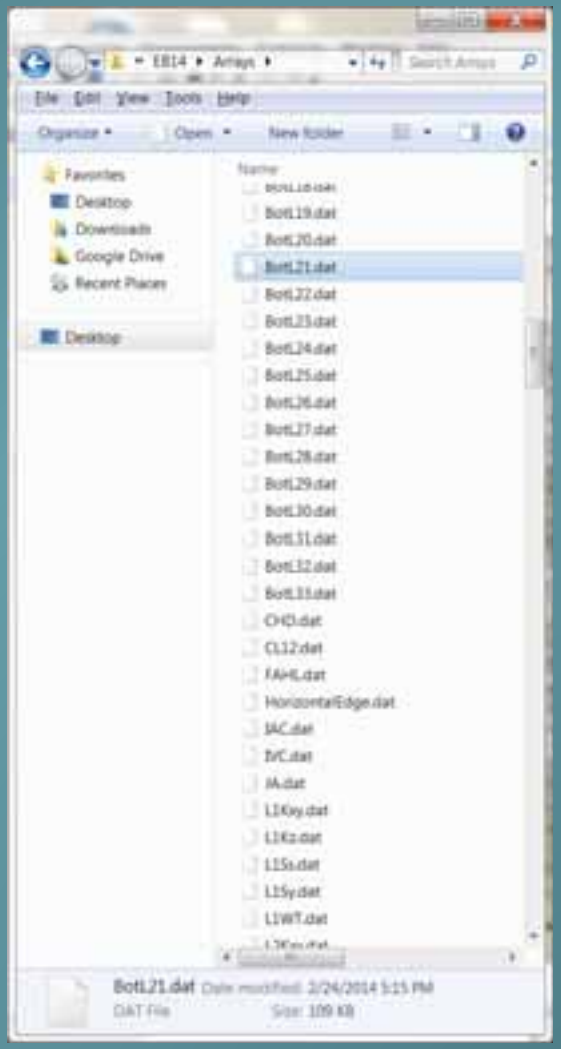
Layers 12 ends (pinches out), no connection to layer 3.



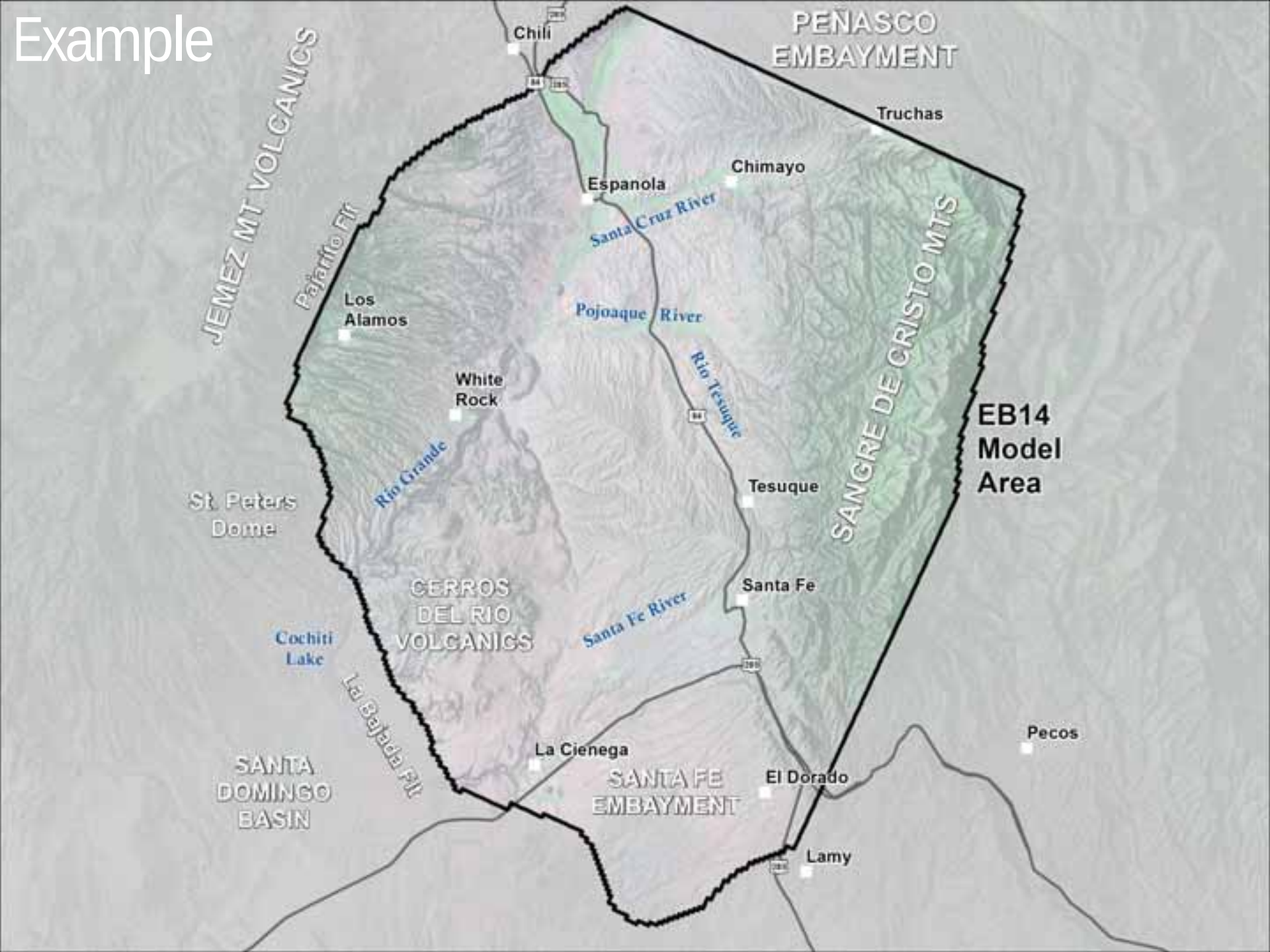
2) Lateral Connections by layer, flow area by layer geometry

Interaction with MODFLOW-USG

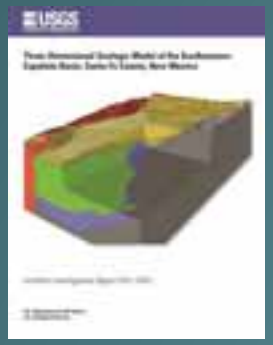
1) Write discretization package arrays
2) Write LPF/UPW arrays
3) Write starting head
4) Return number of cell in each layer
5) Return top most active cells
6) Return edge cells
7) Write constant head package
8) Write RIV1 and ET packages
9) Create polygon FC from geo-cells
10) Create cross section
11) Read MF-USG solution head file and create water-table grid and saturated outcrop.

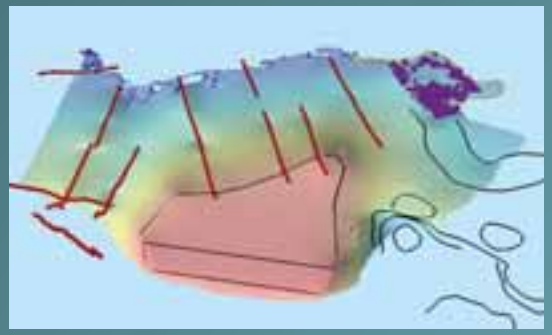


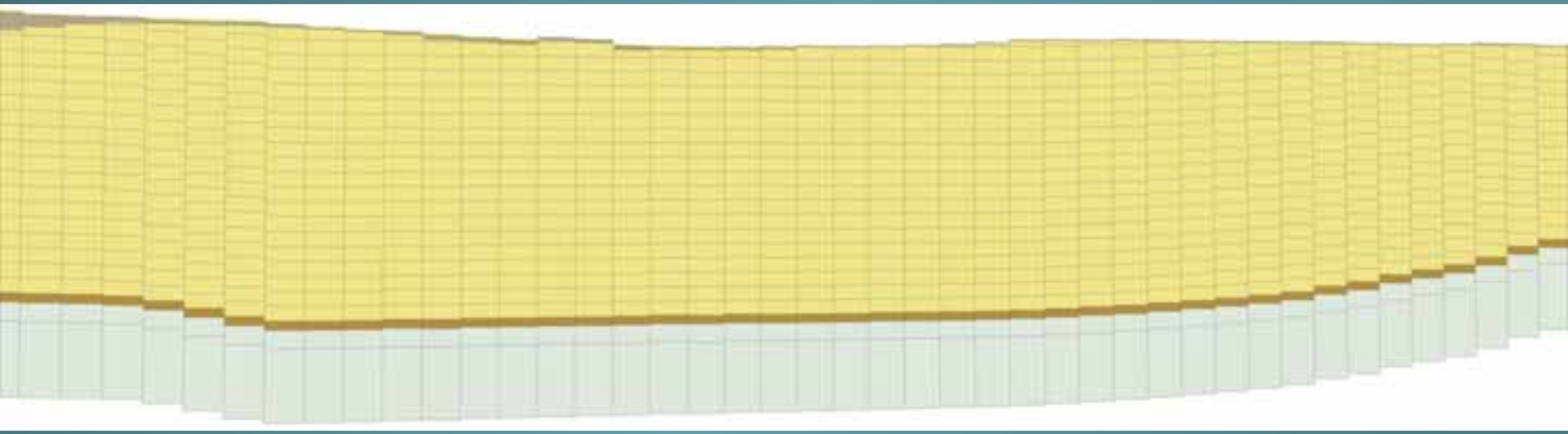
Example


Chili
PEÑASCO EMBAYMENT
Truchas
Chimayo
Espanola
Santa Cruz River
JEMEZ MT VOLCANICS
Pajarito Flt
Los Alamos
Pojoaque River
Rio Tesuque
SANGRE DE CRISTO MTS
White Rock
EB14 Model Area
Rio Grande
Tesuque
St. Peters Dome
Santa Fe
CERROS DEL RIO VOLCANICS
Santa Fe River
Cochiti Lake
La Bajada Flt
Pecos
La Cienega
SANTA DOMINGO BASIN
SANTA FE EMBAYMENT
El Dorado
Lamy

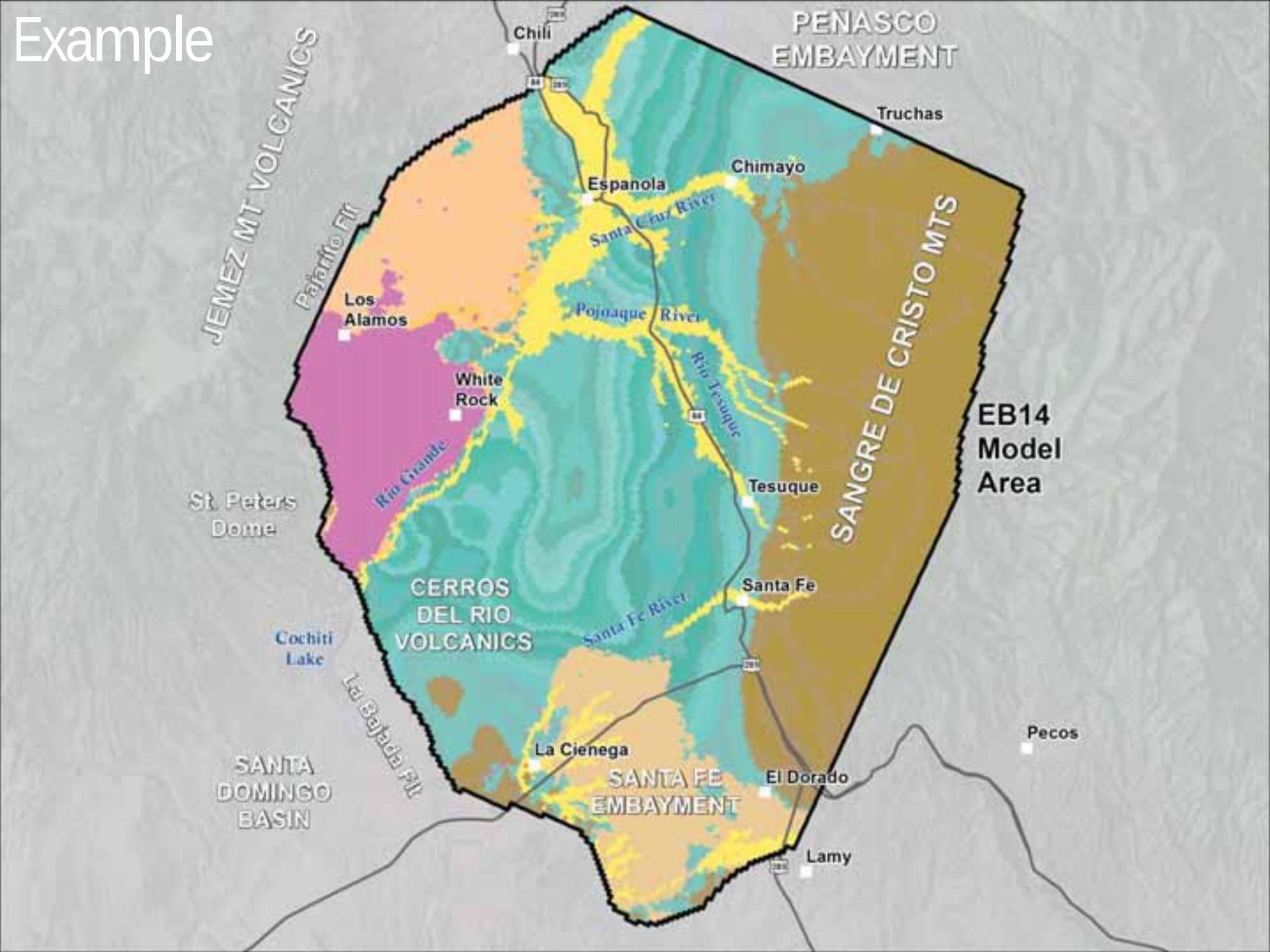
Example



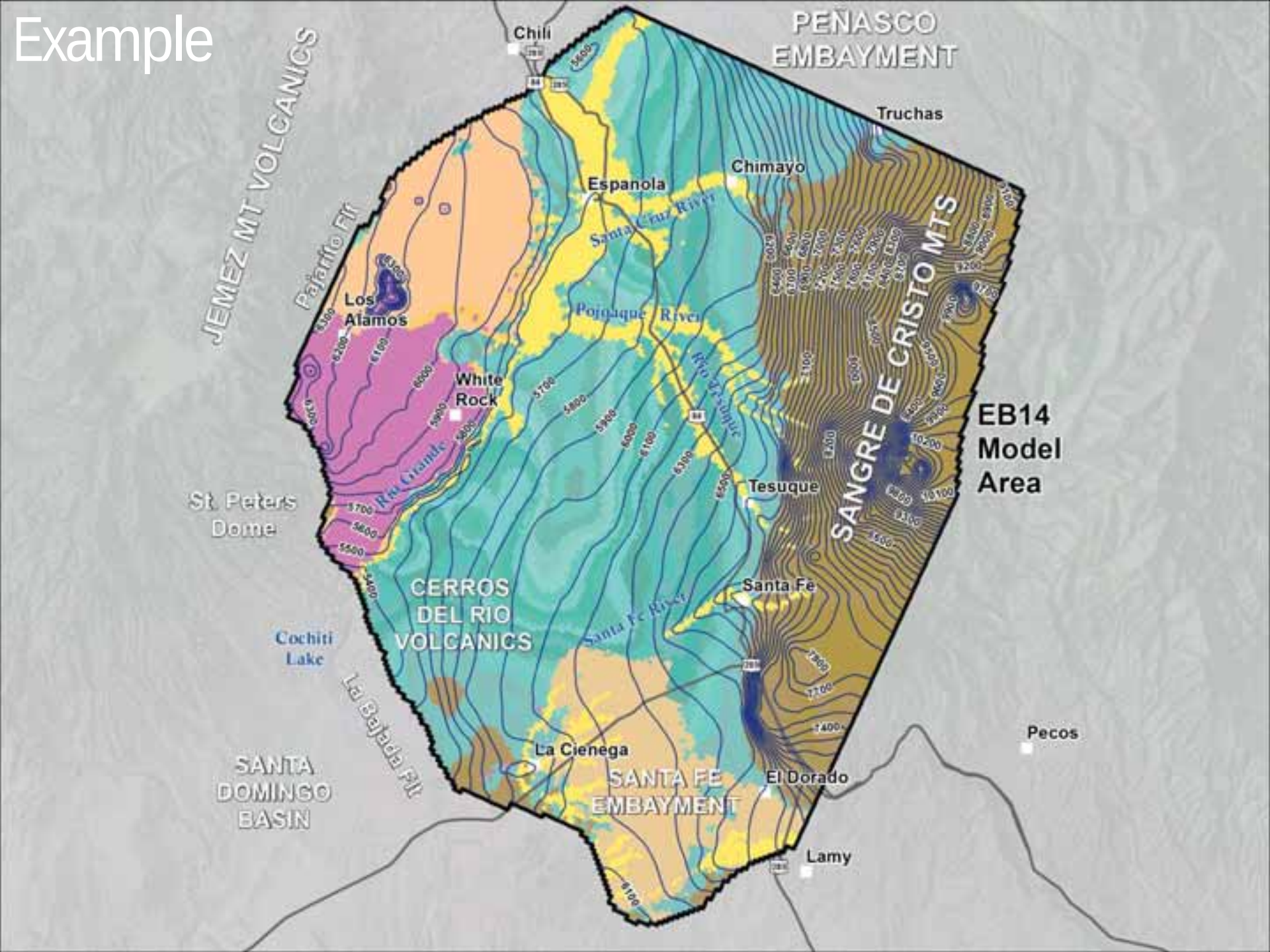




Example


PEÑASCO EMBAYMENT
Chili
Truchas
Chimayo
Espanola
Santa Cruz River
JEMEZ MT VOLCANICS
Pajarito Flt
Los Alamos
Pojoaque River
Rio Tesuque
SANGRE DE CRISTO MTS
White Rock
EB14 Model Area
Rio Grande
Tesuque
St. Peters Dome
Santa Fe
CERROS DEL RIO VOLCANICS
Santa Fe River
Cochiti Lake
La Bajada Flt
Pecos
La Cienega
SANTA DOMINGO BASIN
SANTA FE EMBAYMENT
El Dorado
Lamy

Example


PEÑASCO EMBAYMENT
Chili
Truchas
Chimayo
Espanola
Santa Cruz River
Pojoaque River
Rio Nambe
JEMEZ MT VOLCANICS
Pajarito Flt
Los Alamos
White Rock
Rio Grande
SANGRE DE CRISTO MTS
EB14 Model Area
Tesuque
St. Peters Dome
CERROS DEL RIO VOLCANICS
Santa Fe
Santa Fe River
Cochiti Lake
La Bajada Flt
Pecos
La Cienega
SANTA DOMINGO BASIN
SANTA FE EMBAYMENT
El Dorado
Lamy

Next Steps?

1) Ghost Node Placement and Calculation

2) Connected Linear Network Support

3) Transient stress packages