


# River Flood Simulation

Catalog Tree

- data
  - terrain
  - floodsim.gdb
    - flood\_stages\_ras.img
- docs
  - floodsimbbc\_20140218\_2.sxc
  - pbccFloodSim\_20140218.mxd
- scratch
- tools
  - pbccFloodSim\_20130217.pbcc.t

Contents

Name	Preview	Describe
terrain		
floodsim.gdb		
flood_stages_ras.img		



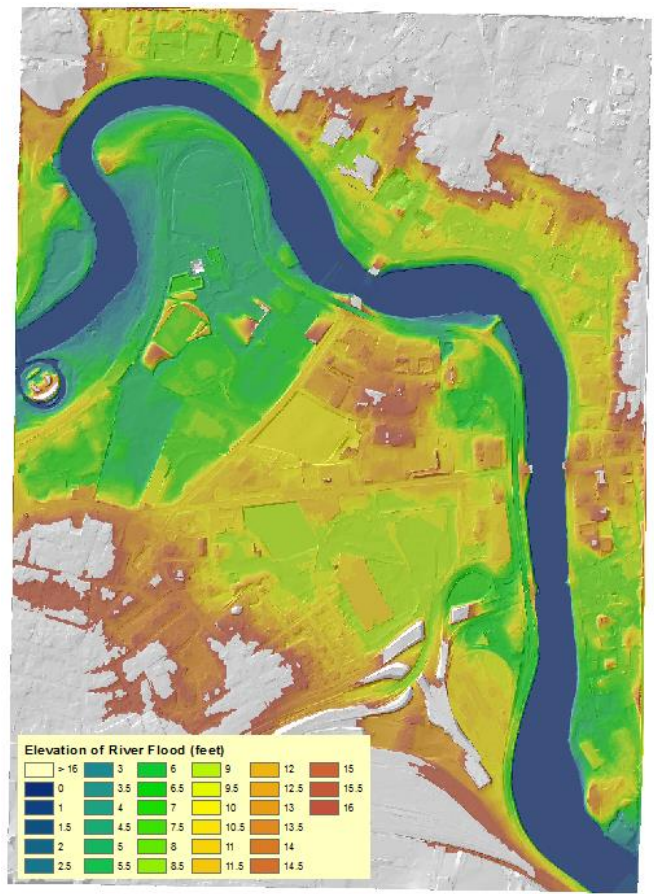
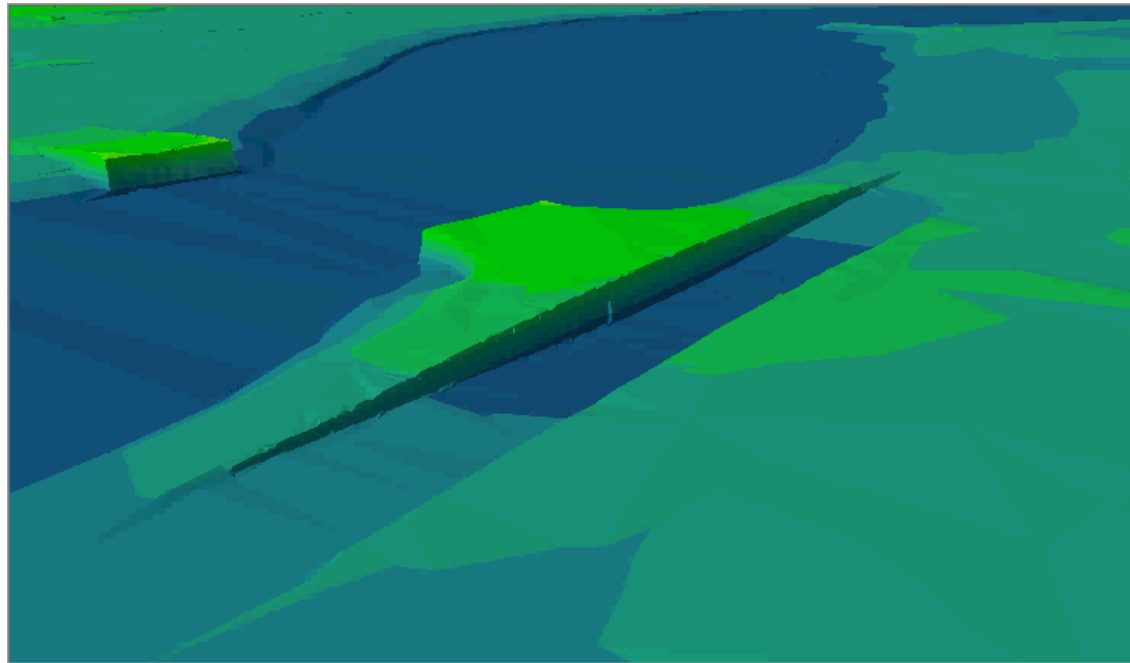
```

# Flood Sim
# Flow # in the flowline (1, 2)
## Flood area with elevation equal to or lower than current flood stage
print "Calculating elevation stage" + str(x) + ".tif"
ascopy_entrance = ""
elev_exp = "RasterToPolygon_conversion(entrance_dam, elev_exp)
floodline_area = RasterToPolygon_conversion(floodline_area, floodline_area)

## Spread flow lines into the floodline lower area
print "Flow lines" + str(x) + ".tif"
ascopy_entrance = elev_exp
floodline = RasterToPolygon_conversion(floodline_area, floodline_area)
floodline = RasterToPolygon_conversion(floodline_area, floodline_area)
ascopy_entrance = floodline
flood = RasterToPolygon_conversion(floodline_area, floodline_area)
ascopy_entrance = ""

## Save the current flood stage area to the shape
print "Current flood stage" + str(x) + ".tif"
stage_area = CellStatistics(flood_line, flood, "STATISTICS", "MEAN")

## Convert the flood area to a polygon and write it to our polygon feature class.
print "Export to Shape" + str(x) + ".tif"
flood_stage_shape = RasterToPolygon_conversion(flood_line, flood_stage_shape)
ascopy_RasterToPolygon_conversion(flood_line, flood_stage_shape)
ascopy_AddField_management(flood_stage_shape, "stage_area", "float", "LENGTH", "M")
M = "M / 2.5"
ascopy_CalculateField_management(flood_stage_shape, "stage_area", "M")
                
```



Click to play video.

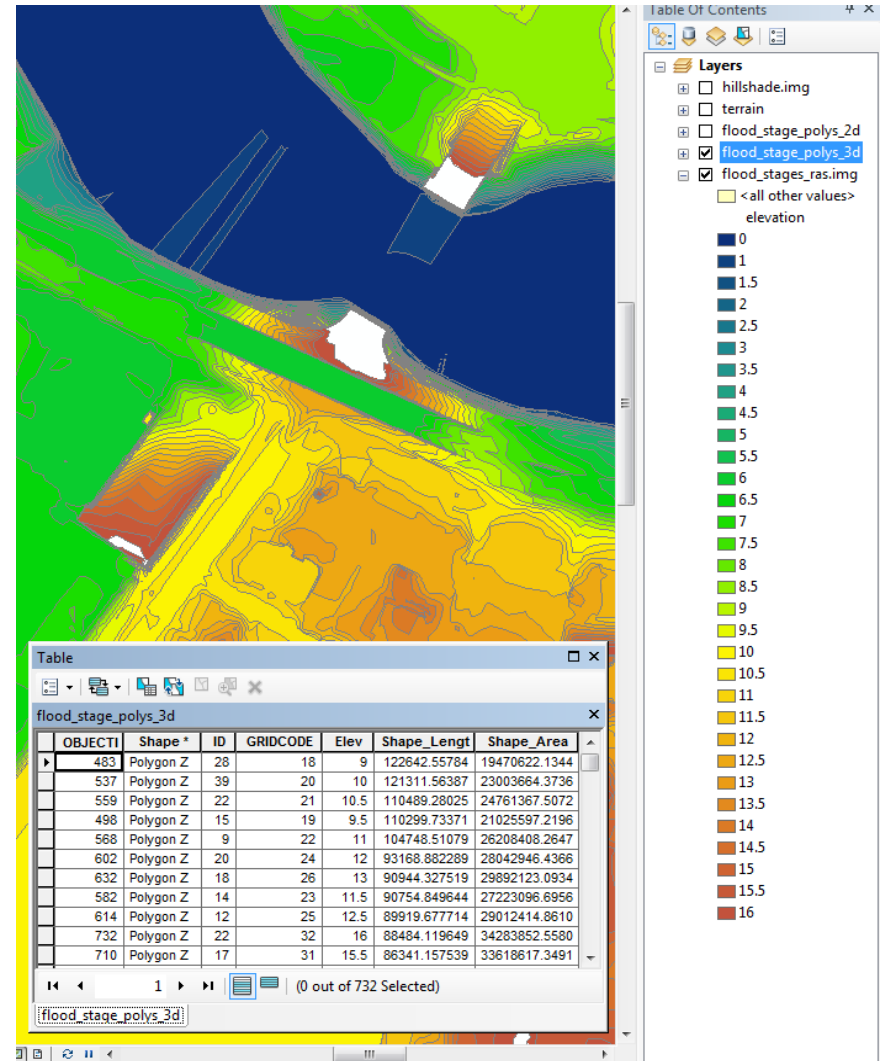
# Flood Simulation Objective

Develop estimated flood stage polygons that estimate the extent of inundated areas as the Charles River rises in half-foot increments.

The source of water for this flood is assumed to be the river. Land area remains dry until connected to the river by the flood. Depressions are not filled until water rises above intervening landforms.

The attribute table for the polygon layer is shown on the right. The Elevation field denotes the height of the water at each flood stage.

2D and 3D feature classes were produced.



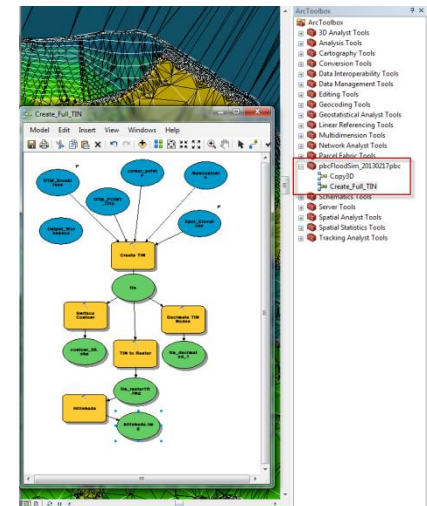
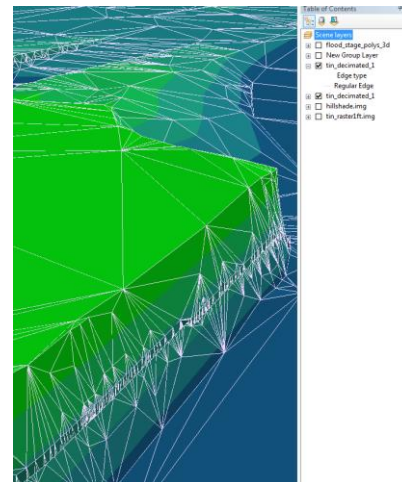
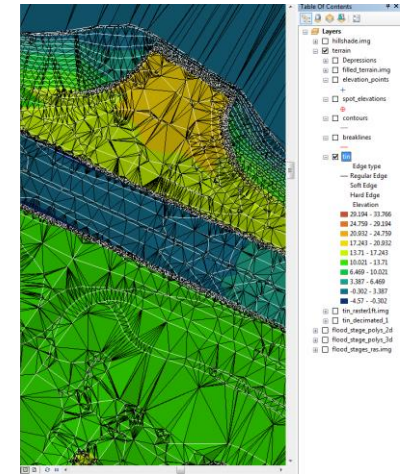
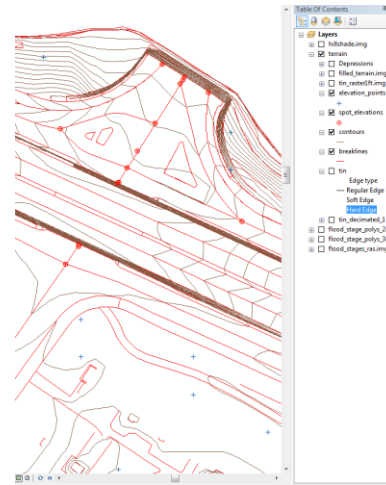
# Surface Development

All of the estimations are based on a survey provided by the client. The data are based on a 2010 photogrammetric survey conducted by InfoTech.

According to their documentation, their elevations are based on the North American Vertical Datum of 1988.

Masspoints, Spot Elevations Contours and Breaklines were interpolated using the ESRI CreateTIN tool. The TIN was converted to a raster with a 1 Foot cell resolution.

The procedure is recorded in the CreateFullTin model provided in the Tools Folder.

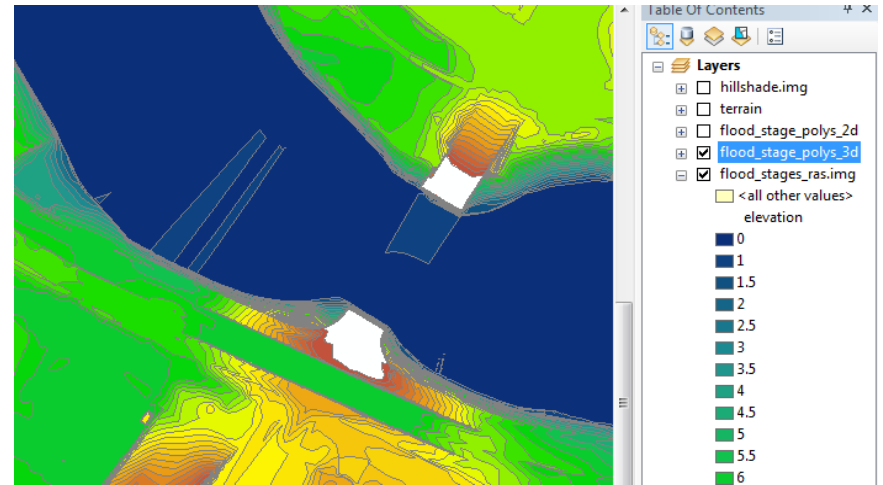


# Flood Model

The model begins by locating the areas of the terrain surface that do not drain into the river. These areas do not flood until they are connected to the river by the flood.

Masspoints, Spot Elevations Contours and Breaklines were interpolated using the ESRI CreateTIN tool. The TIN was converted to a raster with a 1 Foot cell resolution.

The procedure is recorded in the CreateFullTin model provided in the Tools Folder.



```
59
60 ##Increment Flood Stage
61 for x in range(2, 33):
62
63     ## Find area with elevations equal to or lower than current flood index
64     print "Extracting elevation stage" + str(x) + "\n"
65     arcpy.env.mask = ""
66     elev_Exp = "VALUE <= " + str(x)
67     elevMask = ExtractByAttributes(raster_dem, elev_Exp)
68     #elevMask.save("G:\current_projects\geo\boston_projects\harvard\floodsim_pbc\scratch\
69
70     ## Spread from River into the connected lower area
71     print "Cost Distance " + str(x) + "\n"
72     arcpy.env.mask = elevMask
73     cost = CreateConstantRaster(x, "INTEGER", 1,)
74     floodMask = CostDistance(river_ras, cost)
75     arcpy.env.mask = floodMask
76     flood = CreateConstantRaster(x, "INTEGER", 1,)
77     arcpy.env.mask = ""
78
79     ## Save the current flood stage area to the stages raster
80     print "Updated Stages Raster " + str(x) + "\n"
81     stages_ras = CellStatistics([stages_ras, flood], "MINIMUM", "DATA")
82
83     ## Convert the flood zone to a polygon and write it to our polygon feature class.
84     print "Raster to Poly " + str(x) + "\n"
85     flood_stage_shp = "G:\current_projects\geo\boston_projects\harvard\floodsim_pbc\scratch
86     arcpy.RasterToPolygon_conversion(flood, flood_stage_shp, "SIMPLIFY", "VALUE")
87     arcpy.AddField_management(flood_stage_shp, "Elev", "DOUBLE", "", "", "", "", "NULLABLE")
88     e1 = x / 2.0
89     arcpy.CalculateField_management(flood_stage_shp, "Elev", e1)
90
```

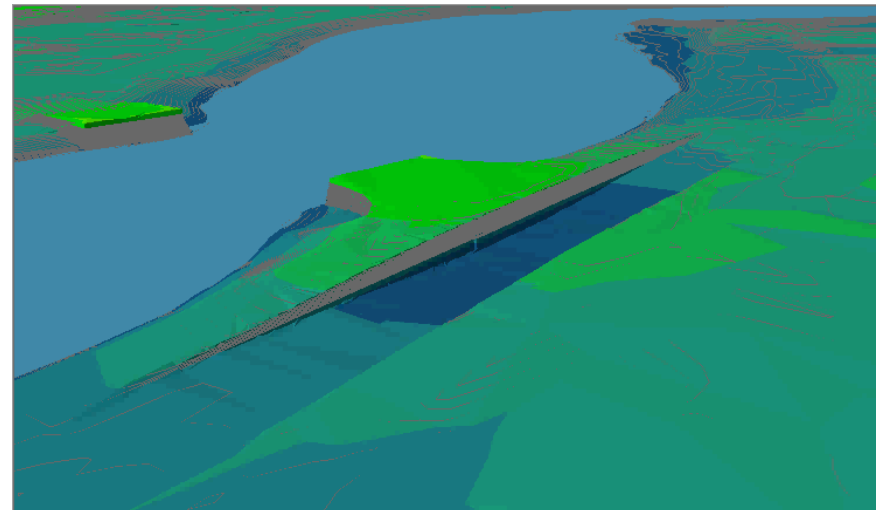
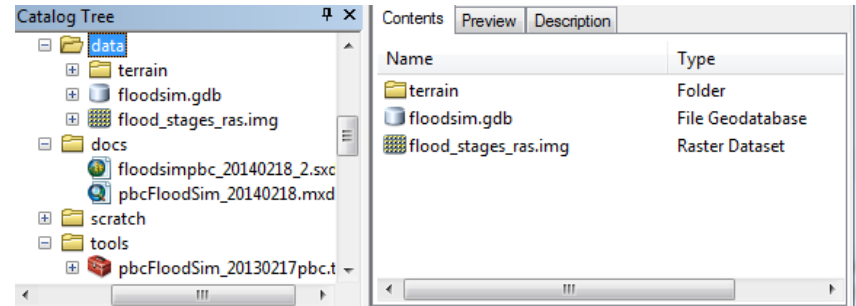
# Products

The data folder includes a geodatabase with two feature classes holding the sequence of flood stage polygons.

There are 32 polygons in this feature-class each of them covers the full extent of the flood at the given stage.

Because the river has an elevation of 1 foot, there is no flood stage for 0.5 feet. These polygons are saved as 2D and 3D features in the FloodSimpbc geodatabase. The geodatabase also includes each polygon as an individual feature class.

A raster that represents the vanguard of each flood stage in plan is also included since it is helpful for viewing in plan.



[Click to play video!](#)

