



FACTOR

1

# Earthquake Updates and Enhancements to Processing for 3.2

TROY SCHMIDT, SENIOR SOFTWARE DEVELOPER

FACTOR, INC.



# Outline

2

- Goals of Hazus 3.2 release
- Hazard Factors Processing
- Probabilistic Analysis
- ShakeMap Analysis
- Performance comparison
- Import Updates
- Questions



# Goals of Hazus 3.2 release

3

- Migrate ArcObjects calls to Geoprocessing Tools
- Eliminate pGDBs still needed for Earthquake
- Leverage Spatial SQL capabilities
- Improve Processing Speed
- Simplify processing where possible
- Make internal processing more accurate



# Hazard Factors Processing

Soil, Liquefaction, Landslide, WaterDepth

4

- New spatial tables created in study region database: eqSrSoil, eqSrLQF, eqSrLND and eqSrWaterDepth
- Values updated when default value updated, user datamap loaded, or as first step of analysis
- ArcObjects uses ICursor an iterative row by row loop processing method
- Spatial SQL utilizes set based approach updating all values in a single step



# Hazard Factors Processing

Soil, Liquefaction, Landslide, WaterDepth

5

## v3.1

- Create fields dynamically in pGDB (if not existing)
- Loop over all features and load the default value
- Loop over all features 1 by 1 and update values by performing spatial lookup
- Push updated values from pGDB to SQL via stored procedure

## v3.2

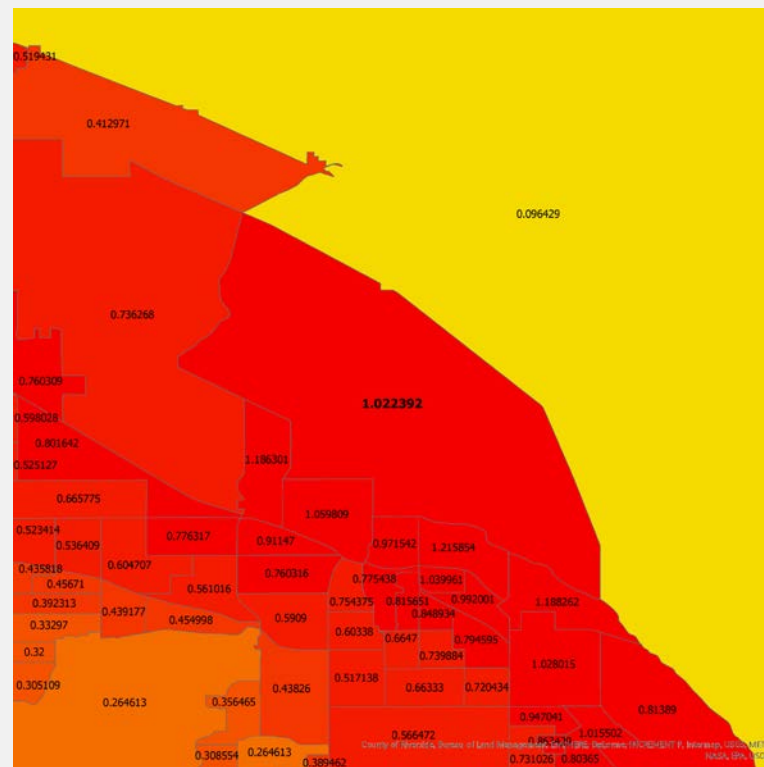
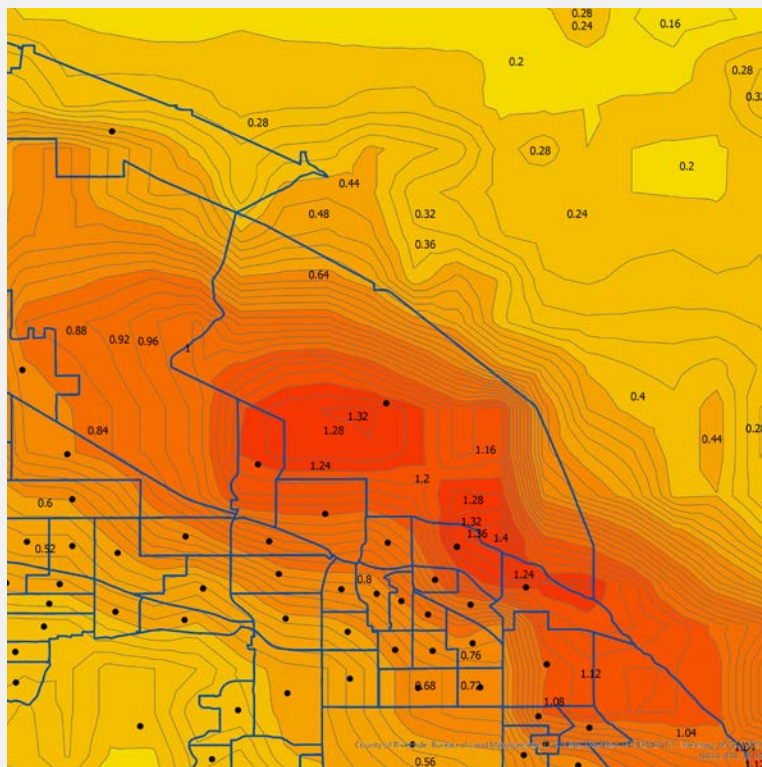
- Use Spatial SQL via stored procedure to update features value in a set based fashion



FACTOR



# Probability & ShakeMap Analysis





# Probability Analysis

7

## v3.1

- Processed central USGS grids to USGS grids just of the study region at aggregation
- Loop over all pGDB tables and add fields (PGA, PGV, SA03, SA10)
- Loop over all features 1 by 1 and update values by performing spatial lookup
- Loop over all features and compute their soil amplification value in code
- Push updated values from pGDB to SQL via stored procedure

## v3.2

- Stored procedure to update values directly using central USGS grids



# ShakeMap Analysis

8

## v3.1

- Loop over all pGDB tables and add fields (PGA, PGV, SA03, SA10)
- Loop over all features 1 by 1 and update values by performing spatial lookup
- Push updated values from pGDB to SQL via stored procedure

## v3.2

- Stored procedure to update values directly using spatial SQL against new spatial tables





# Performance Comparison

Study Region	Hazus 3.1	Hazus 3.2	Performance increase
Charleston, SC	1hr 43min	10 min (Win10 32GB)	10.3x faster
Greater Los Angeles area	1hr 58min	58 min (Win10 32GB)	2.03x faster
Kenai Alaska	Could not import 50K AEBM	1hr 2min (Win10 32GB)	

Aggregation times enhanced to less than 5 minutes\*



# Importing (UDF & AEBM)

## v3.1

- Tabular data was supplied as input with lat long values as fields
- Geometry was created for each row as it was imported
- Temporary tables were created and then joined and records add to SQL through a convoluted process
- For some processes, 3.x users needed elevated ESRI license level to complete

## v3.2

- Views for each feature type were created as templates of the data structure
- pGDB Feature classes are the input and their geometry is imported directly by projecting to Hazus wkid 4326
- Import done in a single step via parameterized SQL



# Impacts of Import Changes

11

- Spatial data is imported directly and not recreated
- Paves the way for importing line and shape data
- Process is greatly simplified for maintainability and future enhancement
- Speed is increased dramatically. Approx. 49,000 records in 5 minutes

# Questions?