Simulation of an Economic Model for Earthquake Recovery

Chris Chamberlin

University of Washington – Department of Geography

ABSTRACT

This poster describes the design and implementation of a software simulation of an economic model of earthquake loss and recovery. The model calculates losses due to building and lifeline damage in the period after a simulated earthquake in the Los Angeles area. The simulation incorporates damage and economic data from several sources and produces loss predictions based on this model. The software is designed to efficiently execute multiple simulation runs to account for the random elements of the model. Its object-based design will allow for future extension or modification of aspects of the model.

BACKGROUND

The project described here is a component of the Direct Losses, Social Impacts, and Community Resilience – Los Angeles Lifeline Study. The study will include social and resilience models, which will be simulated in a manner similar to the economic model described here. The study region is the Los Angeles Department of Water & Power service area in Los Angeles County (see map).

The model calculates predicted economic losses due to business closure after an earthquake. The simulation is run on a hypothetical sample of businesses generated from actual business information for the County. Building, water, and power damage data is drawn from other simulations, including HAZUS, and serves as input to this loss model. Vulnerability to building and lifeline damage is based on existing surveys (University of Delaware Disaster Research Center), for seven industry sectors.

OBJECTIVE

This simulation is a supporting tool to enable experimentation with damage mitigation techniques, by efficiently simulating and predicting economic losses due to earthquake damage. The simulator design should be a faithful representation of the underlying loss model, so that future development of one component can lead to similar improvements in the other. This objective is met by clearly defining the relationship of each data and procedural component in the simulator to the corresponding element of the economic loss model.

SIMULATION PLATFORM

Initial prototyping of the model implementation was completed in a spreadsheet, which is inefficient for large datasets but helps to refine the model and will provide a check on the results from the final model implementation.

The full implementation is written in C++, using the programming language’s standard template library (STL). It is designed to be portable, so it will run on any platform with a C++ compiler. C++ helps to enable clear data structure design and clean extensibility so the simulator can be enhanced in the future.

SIMULATOR DESIGN

A simplified object model shows the structure of the data that serves as input to the simulator:

- **Resiliency Simulator**
  - Industry
    - Damage impacts
  - Business
    - Business size
  - Zone
    - Building damage rates
    - Lifeline status by time

After the input data is loaded, the simulation runs by examining every business at each time step. Lifeline and building damage is calculated based on a business’s location (by zone, same as a census tract for this implementation). Based on this damage, the average level of impact for the industry, and a random element, the open/closed status of a business is selected. The number and length of time steps are adjustable.

Because there are several points where outcomes are randomly selected, the simulator is not deterministic. Hence, we run many iterations of the same scenario to develop a sample of possible economic loss levels. In order to execute multiple simulation runs efficiently, the unchanging input data is kept separate from the simulation results, so the in-memory data structures can be reinitialized quickly. This means that for a large dataset such as that of the Los Angeles study area, static data does not need to be reloaded after each simulation run, and multiple runs are fast to execute.

After all of the simulation runs are complete, simple summary data is saved along with the more detailed results of each run in a simple text format. Further analysis of the results will likely be performed with a combination of simple scripts and statistical software.

At present, input data from the various sources (such as the HAZUS earthquake simulator output) is imported mostly by hand into the comma-delimited text format used by the simulator. In the future, supporting scripts may be valuable to speed conversion of datasets. Another improvement would be to enhance the input data storage with a relational database; the object-oriented nature of the design would make this possible without modifying the simulator core.

CONCLUSIONS

While the simulation software is currently unfinished, initial experience indicates that the choice of custom-written software allows valuable flexibility to make the software closely match the underlying model. The C++ library (STL) is essential to speed development by providing data structures that are integral to the software. In addition to coding the simulator, collecting input data and transforming it to a format usable by the simulator also takes substantial time.

ACKNOWLEDGEMENTS

Principal investigator Stephanie Chang is the primary designer of the model implemented by the software described here.