Input/Output Modelling

Project Summary

The project will center around implementing a collection of classes and static methods meant to solve systems of equations, model a complex network, and document the cascading result of the failure of a given node.

Project Problem/Solution

Abstractly, the project will be essentially a calculator for solving systems of equations and a way to document the cascading effect of a node failure on a given network. However, due to this abstraction, the program will be open to subclassing to allow us to apply the systems of equations and complex networks to a variety of concrete problems.

The two applied problems will be primarily concerned with Input/Output economics and economic disaster simulation. With regards to Input/Output economics, the project should be able to correctly assess, given a sector by sector breakdown of the cost of an economic endeavor, whether a given economy will be able to handle the demand of their new project. Furthermore, since a common representation of a network graph is an incidence matrix, we will transform our system of equations into a network graph and simulate the potential for a given economic shock to create a cascade failure in the network, and identify other industries which may need assistance as a result of such a failure.

From the perspective of this course a program of this nature presents obvious difficulties when it comes to implementation. Granted, for small systems of equations and small graphs we are not presented with any significant problems. However, as we begin to perform a significant number of multiplications for rather large systems making sure that we know what is going on "under the hood becomes all the more important." There is a serious risk of integer overflow given the size and number of numerical operations to be performed. Furthermore, since many of the operations performed will involve be done via loops there are several opportunities presented which allow for applications of loop unrolling and code optimization.

Methodologies

Object-Oriented Design: The project will organize a matrix into a collection of various nested objects. For instance, a matrix will be treated as a collection of row and column vectors, and a vector is a collection of tuples. By implementing our classes in this abstract mathematical form we are able to subclass the problem into various concrete applications.

Overflow Safeguards: Since the numbers for the different sectors of a given economy are rather large there will need to be methods to detect and guard against integer overflow and representing our numbers as strings if needed and implementing string based operations. Ideally, if overflow is identified at any point during a computation the project will be restored to a prior state and perform calculations using strings.

Design Patterns: As part of protecting against overflow ideally the project should be able to reset itself to a prior state, and perform its calculations using strings. This would be using the Memento Design Pattern to reset the calculation in the event of overflow.

Compiler Based Code Optimization: Since there are several loops and nested loops within most matrix calculations this presents us with a great opportunity to perform loop optimizations learned in this class using a code profiler to assess the efficiency and design of our methods.

Development/Design

Test/Demo/Validation

Members