Simulation of a University as a Dynamical System

REU Site: Interdisciplinary Program in High Performance Computing

Annie Castner[1], Jenny Louthan[2], Edwardo Rivera[3], Christian Weigandt[4]
Graduate Assistant: Jonathan S. McHenry[5]
Faculty Mentors: Dr. Nagaraj K. Neerchal,[5] Dr. Matthias K. Gobbert[5]
Client: Dr. Michael Dillon[6]

Technical Report HPCF–2012–13, `www.umbc.edu/hpcf` > Publications

Abstract

Dynamical systems describe how complex physical and artificial systems change over time. One such system is a university. The state of a university is described by its endowment, enrollment, research funding, graduation rate, and other characteristics. The Office of Institutional Research at University of Maryland, Baltimore County has been collecting data on characteristics of UMBC in a data warehouse since 1995. We define these characteristics as state variables in our model and formulate deterministic and stochastic equations to describe the relationships among them while utilizing feedback loops in the model to generate cyclical effects of variable relationships. We build the code in MATLAB to simulate this system for UMBC and use a graphical user interface (GUI) to visualize each variable as the university evolves. This model is dependent upon both time and user interactions with initial values of state variables. Our final code models and produces visualizations of the state of the university as it changes over time as well as simulates "what-if" scenarios to aid in campus-wide discussion regarding growth.

Key words: Dynamical Systems, Discrete Simulation, MATLAB, GUI.

AMS subject classifications (2010): 05C21, 37F99, 90B10, 90B15, 90C35.

1 Introduction

A university simulation requires a complex network of equations for state variables to be iteratively evaluated. In order to visualize the model, we start with a simple dynamical system [2] with minimal interactions among variables as represented in Figure 1.1. This figure—although highly simplistic relative to our model—provides a clear representation of a university as a graph, in which nodes and directed edges represent state variables and dependency relationships, respectively.

---

[1] Department of Applied & Computational Mathematics and Statistics, University of Notre Dame
[2] Department of Mathematical Sciences, Lewis & Clark College
[3] Department of Computer Science and of Mathematics, University of Puerto Rico, Rio Piedras Campus
[4] Department of Mathematics and Computer Science, High Point University
[5] Department of Mathematics and Statistics, University of Maryland, Baltimore County
[6] Office of Institutional Research, University of Maryland, Baltimore County

For example, the value of studentsPerFaculty is an input variable in the function that determines the value of the instructionQuality variable for a given timestep of the model. The dotted line from the reputation node to the apply node is representative of a time delay within this dependency. In other words, the apply variable in timestep i is a function of the value of the reputation variable from timestep $i - 1$. However, this representation grossly oversimplifies variable relationships in our model. For a full graphical representation of our dynamical system model and a complete list of variable names and their descriptions, see Figure A.1 and Table A.1, respectively, in the appendix.

The time step of our university model is one year. The result is a discrete simulation that provides iterative snapshots of the university as opposed to a continuous portrayal.

In Section 2, we discuss the motivation behind our research, describe the data resources necessary to create such a model for a university, and explore how we access and utilize these resources. In Section 3 we outline the development of the dynamical systems model of a university, including equation formulation and implementation in code. Section 4 details our application of the concepts from Section 3 to UMBC. In Section 5, we speak to our model as a whole, state our final results, and examine possibilities for future work.

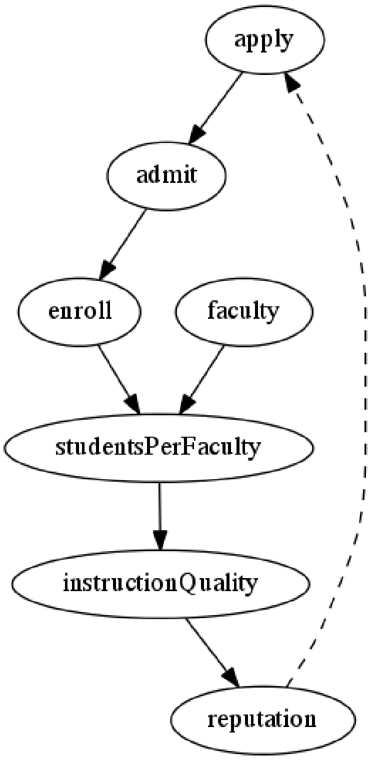


Figure 1.1: Simple Model.

2 Motivation for Model

The Office of Institutional Research (OIR) at UMBC collects data concerning the university's finances, admission, enrollment, student applicants, degrees, and faculty. For data since 1995, they are available in a Data Warehouse. One function of OIR is to analyze this data in order to observe trends over time and understand relationships between variables. The university is then able to use this information to make decisions regarding allocation of its budget, admittance policies, expenditures, and other elements of its operation. We propose the creation of a tool used to utilize UMBC's large quantity of data in order to show quickly and easily how various elements of the university interact with one another and change as the university evolves.

Certain aspects of a university are easy to quantify. For example, the acceptance rate is calculated directly from the number of applicants and the number admitted. However, some aspects of the university are very qualitative yet have a profound impact on how the university operates such that a working model of the university is only realistic and useful to OIR if it captures these complex and difficult to measure relationships. One such example of this is the university's reputation, which effects how many students apply to a given school, the quality of those applicants, and how selective the university must be of those applicants in determining admission. Since current reputation affects these aspects of the university in the future—e.g., the following year—we see a need for feedback loops in our dynamical system model.

Furthermore, the model would be especially useful if the user has the ability to change starting values of the input variables, such as how many students the school admits or how much funding it receives through state appropriations. This ability to change initial values and simulate possible consequences of these changes would aid OIR in determining possible outcomes of "what-if" scenarios. Above all, the purpose of this tool is to aid OIR and the UMBC administration in making positive decisions for the UMBC's future growth and improvement.

2.1 UMBC Data Warehouse and IPEDS

In order to realistically model a university as a dynamical system, we first determine which aspects of the university are fundamental for providing a comprehensive depiction of its state. In addition to defining which variables are important, we gather information about these variables from previous years. UMBC's primary source of campus data is OIR; this data is highly organized and accessible to the public via the office's website. Among the most valuable data that we use from OIR is that which can be found in the annual Common Data Sets online (`www.umbc.edu/oir`).

Since producing a model that is as realistic as possible is paramount, we work extensively with the past data to find significant trends and better understand correlations between variables before replicating these findings in our model. Most of the data that we use to create the model is retrieved from the Integrated Postsecondary Education Data System (IPEDS), online at `nces.ed.gov/ipeds`. By the amended Higher Education Act of 1965, every college, university, and technical institute that offers financial aid to students must publicly report data on enrollment, finances, graduation rates, financial aid, faculty and staff, etc. IPEDS contains all of this data. The IPEDS system has a series of tools, but we focus primarily on the uses of its Data Center. The Data Center is simple to navigate, allowing the user to select an institution or institutions followed by variables and years of the desired information which is then downloadable in the form of a table.

2.2 U.S. News and World Report

A primary example of the neceessity of feedback loops in our model is the effect that reputation has on the university. For our model of reputation, we use one of the country's most respected university ranking systems, U.S. News and World Report [1]. Information detailing the exact information used in their ranking formula and how heavily each item is weighted is publicly available. For the simulation of our model, we replicate this ranking as precisely as possible using the available data. Obtaining an accurate portrayal of a university's U.S. News and World Report ranking—one of the variables simulated by this project—allows us to examine possible steps required in order to maximize reputation as it is viewed on a national scale.

This U.S. News ranking list is organized such that the universities with their reputation evaluated to be highest are ranked 1, 2, 3, and so on, continuing up to ranking 194 in the 2012 list. Before a school is ranked, its reputation is represented with a score ranging from 1 to 100, with 100 being the best. Since the list number is relative to other colleges whereas the 1 to 100 score indicates the university's individual performance, we use the latter to quantify UMBC's reputation.

3 Dynamical Systems Model of a University

3.1 Formulating Equations

Our first step in building update functions for the model is determining the factors that contribute to the value of each variable listed in Table A.1 in the appendix. These include time, other variables, randomness, bounds dictated by institutional resources, or any combination of these. Given the availability of university data from previous years, one of our main approaches is to fit models to past data in order to extrapolate into the future. For much of the data, it is not immediately apparent where variable dependencies lie or which functions provide the best model for the data. Since a university's quantifiable variables are well-understood within that university, we work with different university offices and reference online sources, e.g., IPEDS, to produce realistic equations. Additionally, there exist common techniques for quantifying some of our unmeasurable state variables, such as the university's reputation or quality of its instruction, which we explore and utilize as described in Section 2.2.

3.2 MATLAB Implementation of a Dynamical System

After considering a number of system dynamics software packages, including STELLA and iThink (both at `www.iseesystems.com`) and OpenModelica (`www.openmodelica.org`), we decide to implement our model in MATLAB. Many programs designed specifically for dynamical systems modeling form the basis for their functionality with stock and flow diagrams in a "drag and drop" environment; from this environment, basic code may or may not be able to be generated, with the language of the code dictated by the software. MATLAB allows us to build our university model by writing source code, thus giving us complete control over the format and functionality of the model. Furthermore, the widespread popularity and knowledge of the MATLAB programming language gives our model a level of accessibility unattainable by more specialized system dynamics software packages.

The fundamental element of our model in MATLAB is a one-dimensional cell array data structure `u`, where `u{1}` contains initial values which describe the state of the university in the year 2011. The ith iteration of our model—i.e., the state of the university i years after 2010—is represented in this array by cell `u{i}` which contains a structure data type. The structure in cell `u{i}` has its own data containers called fields. Each state variable in our model is represented by an individual field. Fields can contain data of any type, but doubles are used for our purposes.

The intricate handling of variable interactions in our model necessitates writing and employing a system of equations in our MATLAB code. We formulate an *update function* for each state variable—or field—in order to realistically summarize the state of the university at each time step. The development of such functions is detailed in Sections 3.1 and 4.1. For each iteration of the model, the value of each field is evaluated by calling its respective update function.

As seen in Figure 1.1 and described in Section 1, each university structure in our model can be characterized as a graph where the nodes are its fields and edges the dependencies of its update functions. In order to calculate the value of each field properly for each iteration of the model, we topologically sort the graph such that for all fields i upon which field j is dependent, the calculation of i precedes that of j within a given iteration. In general terms, a topological sort of a graph G with a series of directed edges $[i, j]$ is given by:

$$\forall [i, j] \in E(G), \quad i \text{ precedes } j. \tag{3.1}$$

Despite its cyclical nature, our model is considered acyclic when sorting since it is only necessary to order variables based on interactions within the current time step. We achieve such a topological sort in our code by running a MATLAB function which parses through each variable's update function to compute its dependencies within the current time step. These dependencies give the state variables that must be evaluated before we can calculate the result. Based upon this information, we create a graph of variables to be topologically sorted via MATLAB's `graphtopoorder` function. Our variables need to be topologically sorted only after a change has been made to the dependencies among the variables in the model. After the sort, our system is ready to be evolved as dictated by the user.

As demonstrated in Section 4.2, the user's knowledge of the update functions and code which generate the model need not be in depth. Default initial values of state variables are set to real data from the year 2011 that we obtained through data collection and discussion with OIR. The user can change these initial values to his or her choosing, run the MATLAB simulation, and view the resulting values of the university's state variables for each of the desired number of iterations. All user interactions are accessible through the model's graphical user interface (GUI) and require no knowledge of the complexity of the code.

4 Application of Model to Simulate UMBC

4.1 Specifying Equations for UMBC

As seen in Section 3.1, the process of creating equations is complex and varies from one variable to another. The following sections detail three methods we use for formulating variables: examining past data's relationship with time and researched minimum and maximum values, plotting trends in dependencies among state variables, and incorporating stochasticity in relatively stable variables.

4.1.1 Example with Past Data and Capping Techniques

In this section we provide an example of how past data is used to select and estimate parameters of a stochastic model for one of our variables, the number of admitted freshmen. Figure 4.1 shows the actual number of freshmen admitted to UMBC from 2001 to 2011. Looking at Figure 4.1, we see that a linear model is an inherently good fit for the data; however, the number of freshmen admitted to UMBC each year cannot increase to infinity as time progresses, and thus a linear model is unsatisfactory. Discrepancies between past data and future predictions like this one necessitate exploring other options for realistic equations. In this case, we eliminate the linear growth of equation by modeling the data with a probability density function. MATLAB generates the mean and standard deviation of the past data for admitted freshmen. Given that UMBC expects the number of admitted freshmen to remain relatively stabilized rather than grow linearly, we set the update function for admitted freshmen to draw from a normal distribution

$$\text{admitFreshmen}_i \sim \text{Normal}(\mu, \sigma^2) \tag{4.1}$$

with mean μ and variance σ^2 set equal to values computed from past data. To constrain the number of applicants to between 3000 and 5000, we first sample from the normal distribution and then truncate at these upper and lower limits. This model for admitted freshmen encompasses both the reality of stochasticity within the data and the need for stability.

The MATLAB code of the update function for the number of admitted freshmen in year i reads

```
function out = update_fall_admitFreshmen(u,i)
  out = normrnd(3913.0,509.6);
  if out < 3000
    out = 3000;
  elseif out > 5000
    out = 5000;
  end
end
```

This code implements the capping and uses the MATLAB function `normrnd` that takes mean $\mu = 3913.0$ and standard deviation $\sigma = 509.6$ as input.

4.1.2 Example with Dependencies Among Variables

Other variables in the UMBC model are represented with complex interactions among variables. One example is the update function for the number of out-of-state students enrolled in the year of the ith iteration, modeled by

$$\text{studentsOutState}_i = 0.1155\, e^{0.01249\, \text{reputation}_{i-1}}\, \text{studentsUndergraduates}_i \tag{4.2}$$

As the reputation of the university grows, so does the number of students classified as out-of-state. In our model, reputation is bounded between 0 and 100, with the number

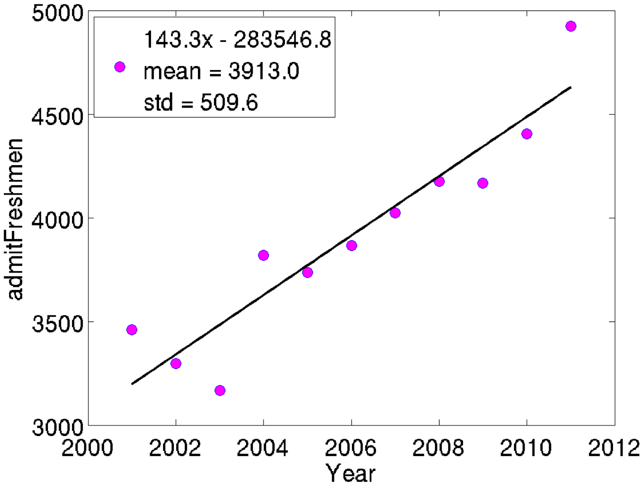


Figure 4.1: Plot of OIR data for number of admitted freshmen.

of out-of-state students similarly bounded from 11.55% to 40.27% of the total number of undergraduates.

To estimate the parameters of (4.2), data from various public schools in the United States are fitted to the exponential model. Since we are concerned with simulating the out-of-state enrollment at UMBC—a public university—and the percentage of these students grows more quickly in private universities, the data points in the left plot of Figure 4.2 represent public universities. In this plot, it seems apparent that a linear fitting would be better suited to the data than an exponential one; however, the right plot of Figure 4.2, which utilizes data from both public and private institutions, demonstrates the need for an exponential model. The rankings in this plot and equation are from U.S. News and World Report and the percentages of out-of-state students from values that the represented schools reported to another source, the Princeton Review (`www.princetonreview.com/home7b`).

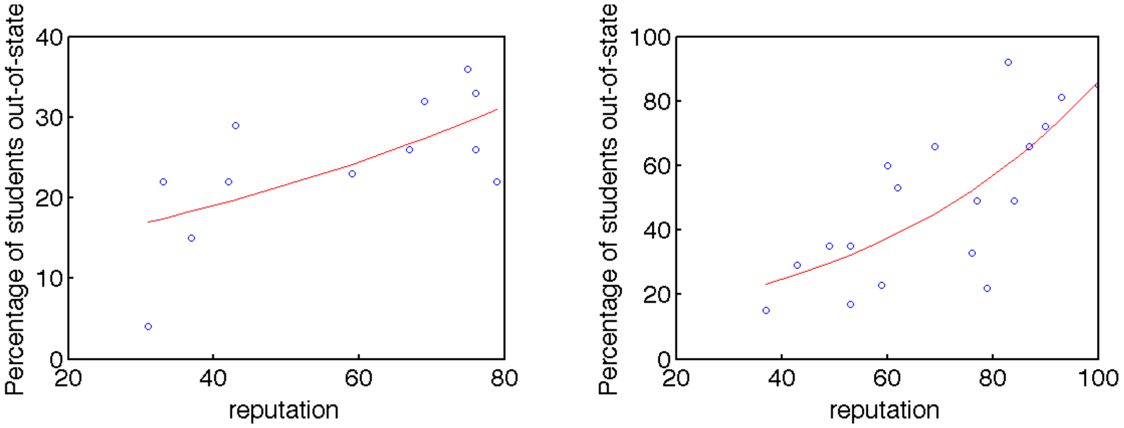
Our MATLAB code for this update function reads

```
function out = update_fall_studentsOutState(u,i)
  out = 0.1155 * exp(0.01249*u{i-1}.reputation) * u{i}.studentsUndergraduates;
end
```

Here, `u` is a cell array whose element `u{i}` is a struct whose fields include `reputation`, `studentsUndergraduates`, and the entries listed in Table A.1. The index $i = 1, 2, \ldots$ specifies the iteration count of `u{i}` as described in Section 3.2.

4.1.3 Example with Stochasticity for Relatively Constant Values

Another variable of exemplifying our methods for assigning update functions is the number of research faculty at UMBC in the ith iteration of the model. The value of this variable is used in calculating the reputation of the university and remains is relatively constant with



(a)

(b)

Figure 4.2: Data from (a) public institution, (b) all institutions.

some minute variation between years. To account this variation, we use a Poisson random variable with mean 3. A uniform random variable combined with an indicator function chooses the sign of our Poisson variable, and our function adds this result to the previous year's research faculty:

$$\text{facultyResearch}_i = \text{facultyResearch}_{i-1} + YZ, \tag{4.3}$$

where

$$\begin{aligned} Y &= 2\ \mathbb{1}(X < 0.5) - 1, \\ X &\sim \text{Uniform}[0,1], \\ Z &\sim \text{Poisson}(\lambda) \end{aligned}$$

with $\lambda = 3$. Our MATLAB code that implements (4.3) as an update function reads

```
function out = update_fall_facultyResearch(u,i)
  y = 2 * (unifrnd(0,1) < 0.5) - 1;
  out = u{i-1}.facultyResearch + y * poissrnd(3);
end
```

This function demonstrates the accuracy necessary in modeling the consistency of the variable and its small variation as seen in Figure 4.3.

4.2 Graphical User Interface Examples

In order for the user to operate our model of UMBC and to check the validity and realism of the equations, we create a graphical user interface (GUI) in MATLAB that provides visualizations of our simulation. With this GUI, the user can view and edit the initial values of state variables and observe the impact these changes have on the university model over time. After initial values are set—or left to their default settings—we are able to set the number of iterations for which our simulation runs, run the simulation itself, and view the

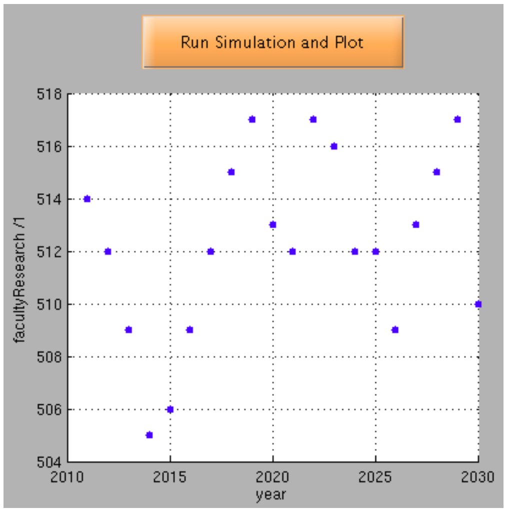
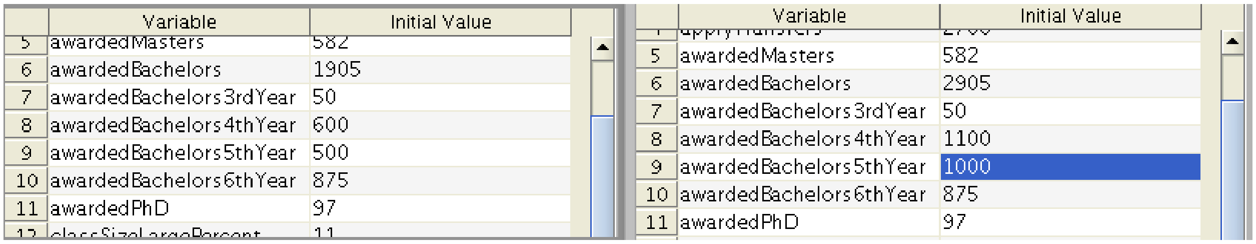


Figure 4.3: Simulated data for number of UMBC research faculty.



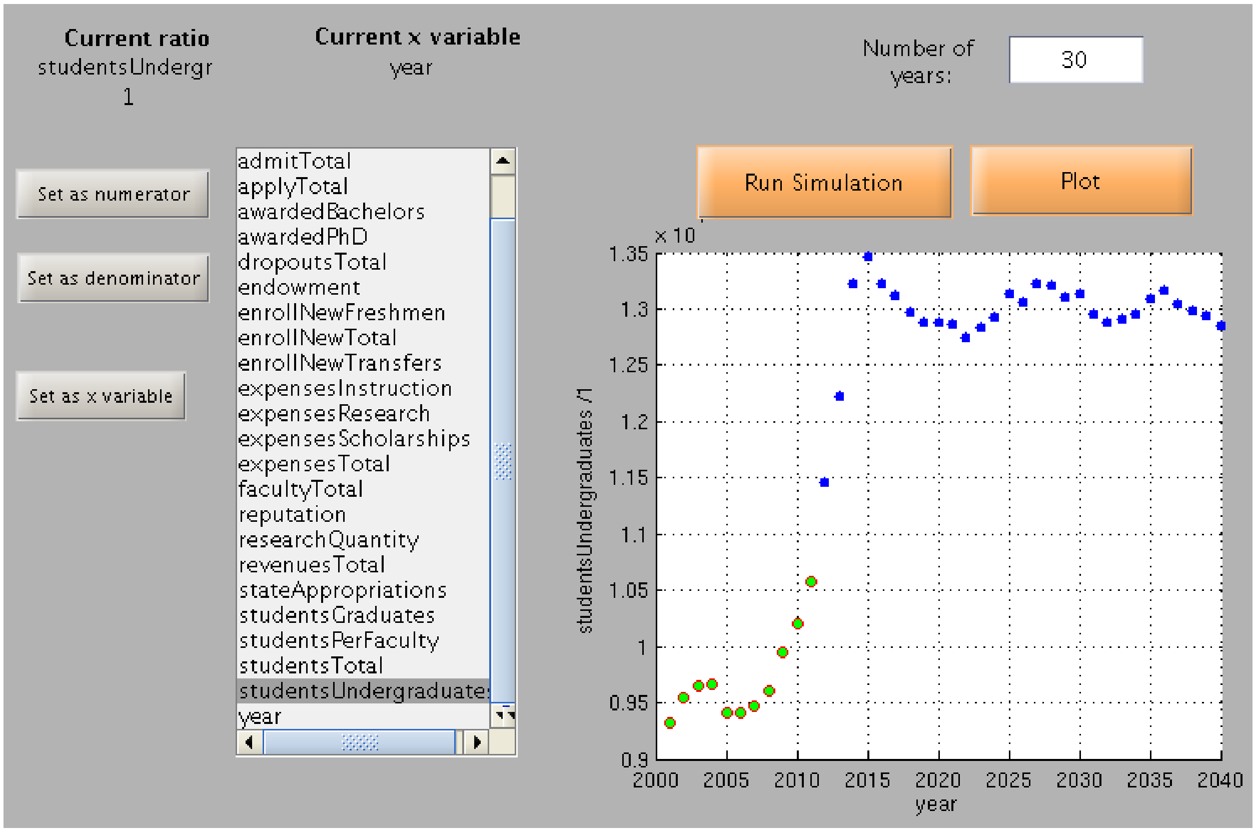
	Variable	Initial Value
5	awardedMasters	582
6	awardedBachelors	1905
7	awardedBachelors3rdYear	50
8	awardedBachelors4thYear	600
9	awardedBachelors5thYear	500
10	awardedBachelors6thYear	875
11	awardedPhD	97

	Variable	Initial Value
5	awardedMasters	582
6	awardedBachelors	2905
7	awardedBachelors3rdYear	50
8	awardedBachelors4thYear	1100
9	awardedBachelors5thYear	1000
10	awardedBachelors6thYear	875
11	awardedPhD	97

Figure 4.4: An example of user-edited initial values of fields indicating number of bachelor's degrees awarded.

evolution of state variables during this simulation in detailed plots of their values. A variable or ratio of variables is selected to be plotted on the y-axis against another chosen variable on the x-axis. Year is the default variable for the x-axis in order to easily understand the evolution of the university through time; other variables can be used if relationships and dependencies are to be studied. Figures 4.4, 4.5, and 4.6 display screenshots exemplifying the functionality of our GUI and its ability to reflect our model's accuracy and examine possible answers to "what-if" scenarios implemented by changing the model's initial values.

Figure 4.4 is a screenshot of an example of user interaction with the model. The left input GUI contains default initial values for our model while the right input GUI shows the fields awardedBachelors4thYear and awardedBachelors5thYear increased by 500 by the user. Note that the field awardedBachelors has also been increased accordingly. Figures 4.5 and 4.6 demonstrate how we can view the effects this change has on the simulation by examining the evolution of another field, studentsUndergraduates. The green data points represent past data for UMBC that was gathered from OIR's website while the blue data points correspond to simulated values. With our default initial values, the number of undergraduate students grows to about 13,000 as expected of a university with the resources and capacity of UMBC. However, when the change in initial conditions is instituted, the undergraduate student body



Current ratio
studentsUndergr
1

Current x variable
year

Number of years: 30

Figure 4.5: Example of results for number of undergraduate students when simulation is run for 30 iterations under default initial conditions.

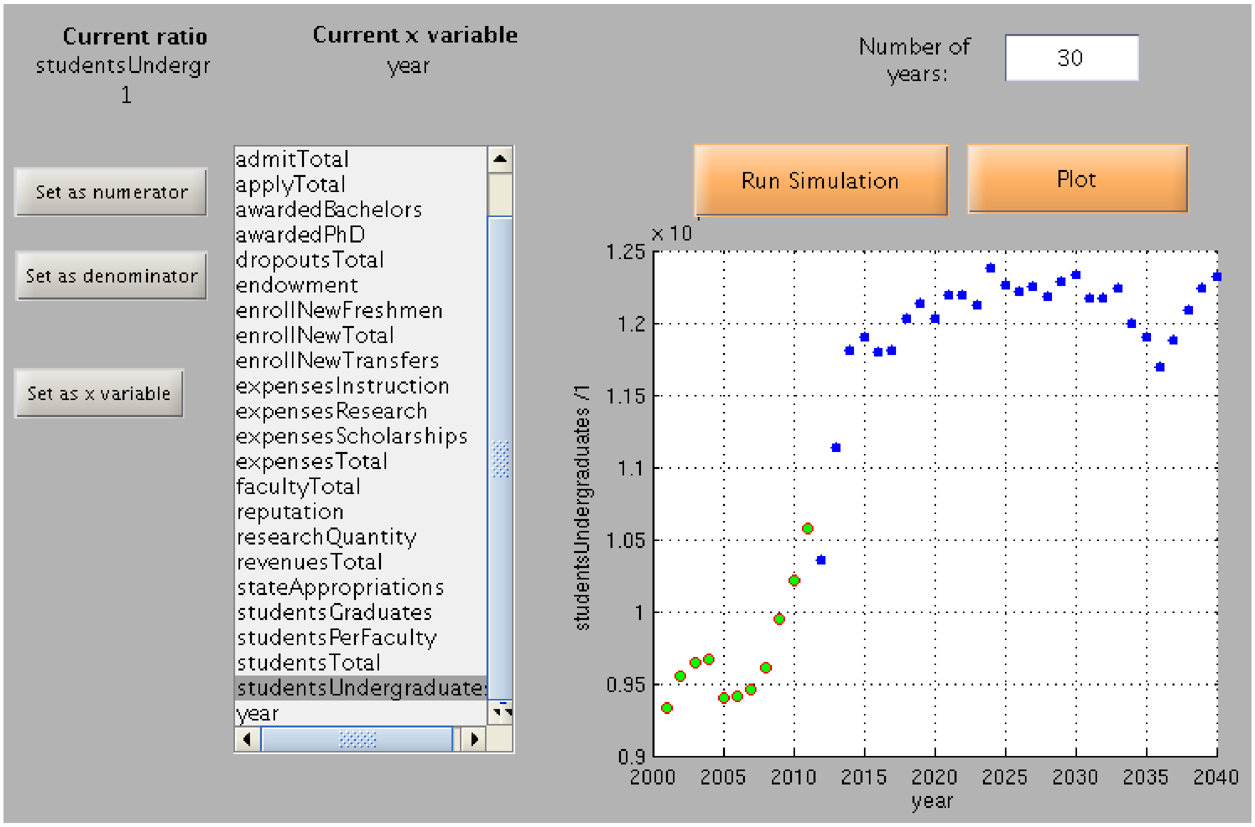
grows to only 12,000 and fails to recover within thirty iterations.

One can see from the GUI that the model accounts for a nearly equal number of student leaving the university as are newly enrolled each year, indicating that UMBC may choose to increase its admission rates temporarily in the case that an unusually high number of undergraduates leave in a given year.

5 Results and Conclusions

Our model is currently in a state capable of producing a realistic simulation of UMBC in terms of 78 state variables. Our package of MATLAB files includes organized and accessible documentation for each of these variables and their update functions, in addition to a README file detailing the structure and functionality of the MATLAB code. These features enable the user to fully understand the model and the steps required to make changes or additions in the future.

Extensions of our project may include adapting a more general model that can be specialized to simulate any university, a feature that could foster useful juxtapostions of interesting components of comparable universities. In order to increase realism in the simulation, split-



Current ratio
studentsUndergr
1

Current x variable
year

Number of years: 30

Figure 4.6: Example of results for number of undergraduate students when simulation is run for 30 iterations under edited initial conditions.

ting yearly iterations into semesters may prove useful. Our model of a university is highly simplified; the composition of a real university involves thousands of components interacting in highly complex and often subjective ways. Thus, we could also explore extending the complexity of our model by including more variables and interactions.

Acknowledgments

These results were obtained as part of the REU Site: Interdisciplinary Program in High Performance Computing (`www.umbc.edu/hpcreu`) in the Department of Mathematics and Statistics at the University of Maryland, Baltimore County (UMBC) in Summer 2012. This program is funded jointly by the National Science Foundation and the National Security Agency (NSF grant no. DMS–1156976), with additional support from UMBC, the Department of Mathematics and Statistics, the Center for Interdisciplinary Research and Consulting (CIRC), and the UMBC High Performance Computing Facility (HPCF).

References

[1] U.S. News and World Report. How U.S. News calculates the college rankings. `http://www.usnews.com/education/best-colleges/articles/2011/09/12/how-us-news-calculates-the-college-rankings-2012?page=4`, accessed August 10, 2012.

[2] Jeff Wasbes. Modeling dynamic systems: An introduction to system dynamics modeling. Slide presentation at `http://www.environmentalevaluators.net/wp-content/uploads/2011/02/Modeling-Dynamic-Systems.pdf`, accessed August 10, 2012.

A Graph of Model and List of State Variables

A convenient representation of the university's state variables is as a directed graphical model, as shown in Figure A.1. Table A.1 lists all 78 state variables of our model.

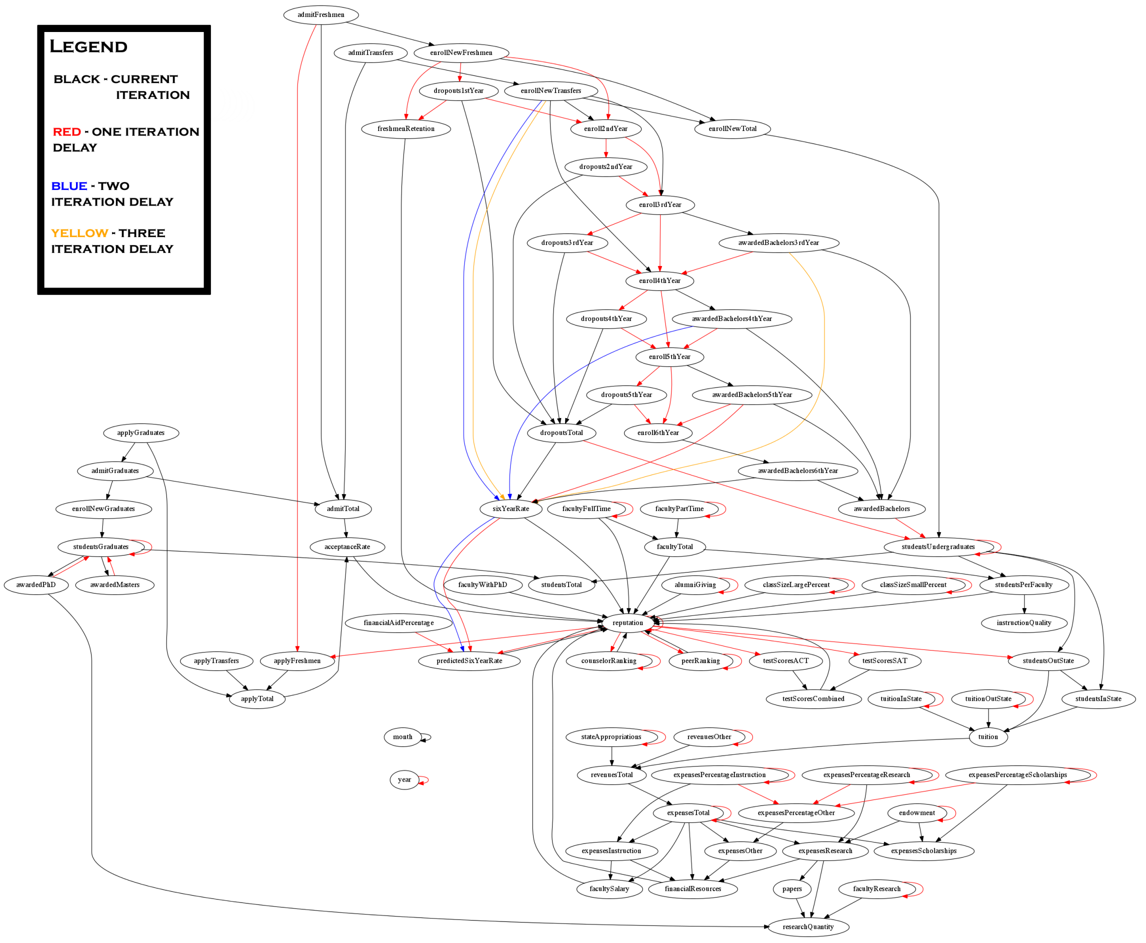


Figure A.1: Graph visualization of our final model.

Variable Name	**Description**
acceptanceRate	Percent of students admitted from those who applied
admitFreshmen	Number of admitted freshmen
admitGraduates	Fraction of applyGraduates
admitTotal	Summation of all admissions
admitTransfers	Number of transfer students admitted
alumniGiving	Percentage of living alumni with a Bachelor's degree who donate within the current year to the university.
applyFreshmen	Number of freshmen applicants
applyGraduates	Number of graduate applicants
applyTotal	Total number of aplicants
applyTransfers	Number of transfer applicants
awardedBachelors	Total number of Bachelors degrees awarded
awardedBachelors3rdYear	Number of Bachelors degrees awarded to juniors
awardedBachelors4thYear	Number of Bachelors degrees awarded to seniors
awardedBachelors5thYear	Number of Bachelors degrees awarded to 5th year students
awardedBachelors6thYear	Number of Bachelors degrees awarded to 6th year students
awardedMasters	Number of Masters degrees awarded
awardedPhD	Number of PhDs awarded
classSizeLargePercent	Percentage of classes classified as "large", i.e., 50 students or more
classSizeSmallPercent	Percentage of classes classified as "small", i.e., 20 students or less
counselorRanking	Ranking out of 5 points; from a survey of opinions of high school guidance counselors randomly selected from nation's best high schools
dropouts1stYear	Number of students who dropped out freshmen year
dropouts2ndYear	Number of students who dropped out sophomore year
dropouts3rdYear	Number of students who dropped out junior year
dropouts4thYear	Number of students who dropped out senior year
dropouts5thYear	Number of students who dropped out fifth year
dropoutsTotal	Number of total dropouts (undergraduate)
endowment	University endowment
enroll2ndYear	Number of students enrolled in their sophomore year
enroll3rdYear	Number of students enrolled in their junior year
enroll4thYear	Number of students enrolled in their senior year
enroll5thYear	Number of students enrolled in their fifth yeay
enroll6thYear	Number of students enrolled in their sixth year
enrollNewFreshmen	Number of newly enrolled freshmen
enrollNewGraduates	Number of newly enrolled graduate students
enrollNewTotal	Summation of all newly enrolled students
enrollNewTransfers	Number of newly enrolled transfers
expensesInstruction	Amount of money funding instruction
expensesOther	Amount of funding in "other" fields
expensesPercentageInstruction	Percentage of expenses that are spent on instruction (salaries, etc.)
expensesPercentageOther	Percentage of expenses on that are spent funding everything else
expensesPercentageResearch	Percentage of expenses that are spent on research
expensesPercentageScholarships	Percentage of expenses that are spent on funding scholarships
expensesResearch	Amount of money funding research
expensesScholarships	Amount of money funding scholarships
expensesTotal	Amount of money being used to fund scholarships
facultyFullTime	Number of full-time faculty
facultyPartTime	Number of part-time faculty

Variable Name	**Description**
facultyResearch	Total number of research faculty
facultySalary	Percentile of faculty salary, compared to how much other universities pay their faculty
facultyTotal	Total number of faculty, both full and part time
facultyWithPhD	Percentage of full-time faculty with doctorate or terminal degree for their field of study
financialAidPercentage	Percentage of undergraduates receiving financial aid in a given year
financialResources	Average spending per student on educational expenditures.
freshmenRetention	Percentage of students who return for their second year of study
instructionQuality	Quality of instruction
month	Current month
papers	Number of academic papers published by university faculty and students
peerRanking	Ranking out of 5 points; from a quality survey of opinions of college presidents, provosts, and deans of admissions nationwide
predictedSixYearRate	Predicted percentage of students who will graduate within six years of enrollment for given year
reputation	Quantifies reputation of university
researchQuantity	Measures research from papers, PHD's awarded, expenses Research, and faculty Research
revenuesOther	Revenue money coming from everything except tuition, state appropriations, and endowment spending
revenuesTotal	Total revenue
sixYearRate	Percentage of students who graduate within six years of enrollment
stateAppropriations	Amount of funding from state each year
studentsGraduates	Total number of graduate students enrolled
studentsInState	Number of in-state students
studentsOutState	Number of out-of-state students
studentsPerFaculty	Student/Faculty ratio
studentsTotal	Total number of students enrolled, both undergraduates and graduates
studentsUndergraduates	Total number of undergraduate students enrolled
testScoresACT	Average ACT test score for given year of newly enrolled students
testScoresCombined	Percentile of the average ACT and SAT test scores of the newly enrolled students
testScoresSAT	Average SAT test score for given year of newly enrolled students
tuition	Total amount of revenue coming from tuition
tuitionInState	Cost of in-state tuition for one student in one year
tuitionOutState	Cost of out-of-state tuition for one student in one year
year	Current year

Table A.1: Complete list of state variables in our model.