PROGRAMS FOR ATTRACTING UNDER-REPRESENTED MINORITY STUDENTS TO GRADUATE SCHOOL AND RESEARCH CAREERS IN COMPUTATIONAL SCIENCE

FINAL REPORT

for Period October 1, 1995 - September 30, 1997

James C. Turner Jr.
Thomas Mason
Bruno Guerrieri

Florida A&M University
Tallahassee, FL 32307

October 1997

Prepared for

THE U.S. DEPARTMENT OF ENERGY
AWARD NO. DE-FG02-~~96ER25302~~ 95ER25280

DISCLAIMER

PROGRAMS FOR ATTRACTING UNDER-REPRESENTED MINORITY STUDENTS TO GRADUATE SCHOOL AND RESEARCH CAREERS IN COMPUTATIONAL SCIENCE

ABSTRACT

Programs have been established at Florida A & M University to attract minority students to research careers in mathematics and computational science. The primary goal of the program was to increase the number of such students studying computational science via an interactive multimedia learning environment. One mechanism used for meeting this goal was the development of educational modules. This academic year program established within the mathematics department at Florida A&M University, introduced students to computational science projects using high-performance computers. Additional activities were conducted during the summer; these included workshops, meetings, and lectures. Through the exposure provided by this program to scientific ideas and research in computational science, it is likely that their successful applications of tools from this interdisciplinary field will be high.

NOTICE

Recently, the department of mathematics at Florida A & M University became engaged in a great deal of self-study. Of the new perspectives that emerged from this self-examination, one is perhaps the most significant: The mathematics faculty must embrace the use of technology in the curriculum. Thus, an early product of this self-examination was the implementation of several novel projects, one of which will be delineated in this report.

During the academic year 1996-97, a small group of mathematics faculty became dedicated to providing FAMU students with an interactive multimedia learning environment. This new environment has the potential of dramatically increasing the effectiveness and efficiency of instruction and learning at FAMU. At the heart of this new project was the development of educational modules on topics from computational science. These modules were meant to be more than just multimedia courseware. They were to play a central role in a comprehensive instructional solution designed to meet the practical requirements of faculty and students.

As a participant in the Undergraduate Computational Engineering and Science (UCES) Project, the department and the university made a commitment to promoting the emerging field of computational science. This commitment is manifested by identifying and supporting excellence in computational science education at the undergraduate level. During the period of this award, the emphasis was on collecting, developing, and distributing to FAMU students a set of computational science educational materials. These materials are problem driven, modular in format, and interactive. Future plans include working with other UCES members to produce a set of full "electronic classes", which could become available on-line from the UCES Web Server.

The following topics have been developed as a preliminary state of educational modules:

- **Computerized Tomography** -This module is designed to give students an introduction to computerized tomography. A brief historical introduction is given. A mathematical model of a CT scan is described. Next, using tools from computational science, a discretized problem is formulated. This formulation used the backprojection method and discrete Fourier transforms. The module ends by solving the discrete problem via an iterative reconstruction algorithm.

- **Design of a Computer-Based Presentation on Mathematical Modeling using Differential Equations: Linear Dynamical Systems** - This module will be developed for students who are moderately literate in undergraduate mathematics, but almost certainly have not taken a course in differential equations. The underlying thesis of this effort is that mathematical modeling can be introduced, understood, and mastered by such students if difference equations are used rather than differential equations. Such an approach does not, of course, preclude the eventual, or even simultaneous, use of differential equations. However, the relative simplicity of difference equations allows a student to concentrate on the modeling process, which is not the case with the typical differential equation approach.

Topics

1. Introduction to the Card Format.
2. Modeling a Savings Account
3. Theory of First Order Affine Systems
4. Financial Applications
   a.) Certificates of Deposit
   b.) Annuities
   c.) How the Lottery Works
   d.) Loan Amortization
5. Political Science Applications
   a.) Partisanship
   b.) Political Attention
   c.) Birth and Death of State Agencies
   d.) Unemployment and Incumbency

- **Design of a Computer-Based Presentation on Mathematical Modeling using Differential Equations: Nonlinear Dynamical Systems** - This module extends the approach of the above module to nonlinear dynamical systems focusing on models of population.

Topics

1. Logistic Equation Models
   a.) Limitations of Exponential Growth
   b.) Logistic Model
   c.) Logistic Model of Population Growth
   d.) Logistic Model of Belief Systems
2. Equilibria and Cobwebbing
3. Analysis of Logistic Models
   a.) Analysis of the Model of Population Growth
   b.) Analysis of the Model of Belief Systems

- **Traffic Flow** - This module is a spin-off from the FAMU High School Supercomputing Challenge. The traffic flow considered consists of cars moving on one side of a divided highway. Rules are developed to describe the behavior of a vehicle. These rules are then placed into a model using von Neumann and Moore neighborhoods. A simulation of the model is then given using cellular automation and *Mathematica*. The last two sections of the code model traffic flow in one direction on a two lane highway. This code can also be used to solve a maze, which is created to model a section of a city .

Design of a Computer-Based Presentation on Mathematical Modeling using Difference Equations

Volume I. Linear Dynamical Systems

Dr. Thomas W. Mason
Department of Mathematics
Florida A&M University
Tallahassee, FL 32307

Volume I. Linear Dynamical Systems

Table of Contents

Introductory Remarks	2
General Layout of a Card	3
Dynamical Modeling	4
Home	5
Modeling a Savings Account	6
Theory	10
Practice	15
Theory of First Order Affine Systems	22
Financial Applications	25
Certificates of Deposit	26
Annuities	28
How the Lottery Works	30
Loan Amortization	32
Political Science Applications	34
Partisanship	35
Political Attention	38
Birth and Death of State Agencies	40
Unemployment and Incumbency	42

Introductory Remarks

This volume — Linear Dynamical Systems — and its companion volume, Nonlinear Dynamical Systems, are written for the audience of a student who is moderately literate in undergraduate mathematics, but almost certainly has not taken a course in difference equations. The underlying thesis of this effort is that mathematical modeling can be introduced, understood, and mastered by such students if difference equations are used rather than differential equations. Such an approach does not, of course, preclude the eventual, or even simultaneous, use of differential equations. However, the relative simplicity of difference equations allows a student to concentrate on — and participate in — the modeling process, which is not the case with the typical differential equation approach.

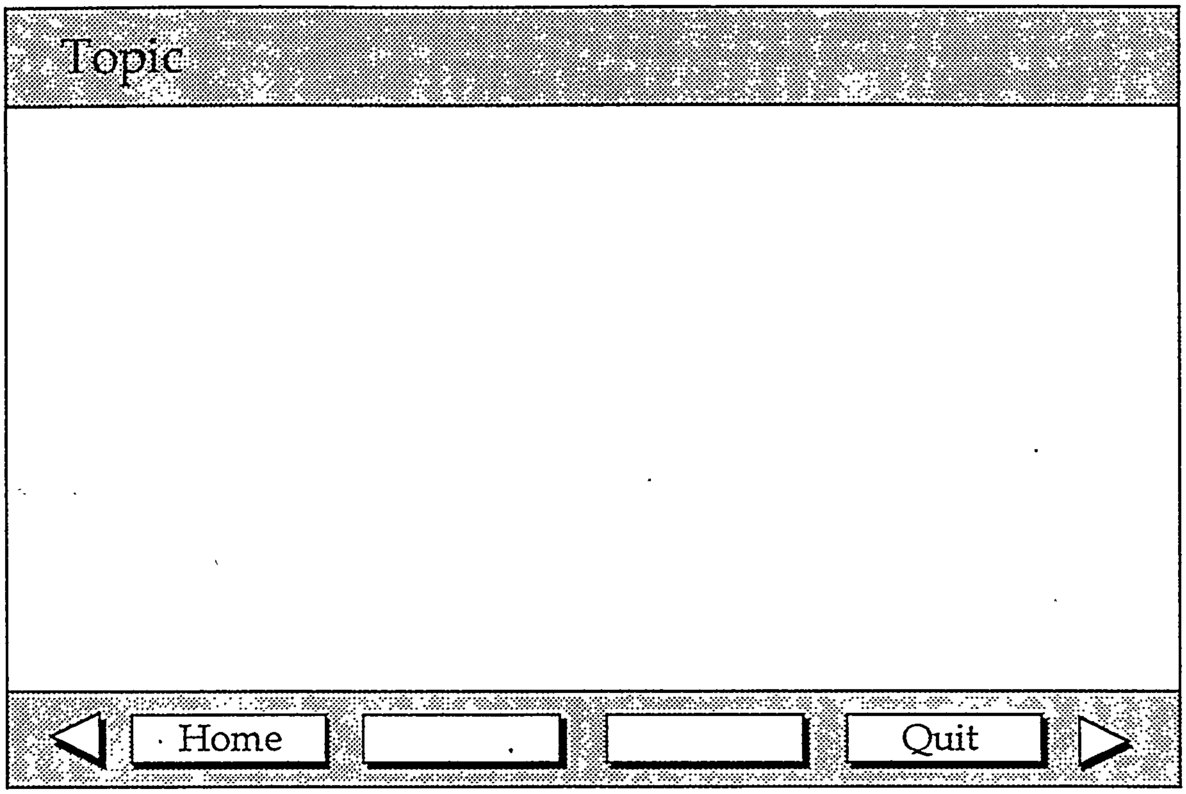
The subject matter for the presentation has been chosen with the adolescent student in mind. It is the experience of the author that such students are primarily absorbed with matters focusing on either money or sex. Hence, Volume 1 presents a difference equation approach to linear dynamical systems concentrating on models of personal finance with a short venture into politics. Volume 2 extends the approach to nonlinear dynamical systems focusing on models of population.

The presentation shows how the material could be presented to students using a hypertext system of the genre of *HyperCard*©. However, the material does not currently exist in a hypertext medium. Therefore, the computer screen displays are simply representative of the sorts of options that students could be given as they study, explore, and master Linear Dynamical Systems.

The material presented in this volume was excerpted from the following references:

HyperCard Reference Manual, Apple Computer, Inc., 1993
Discrete Dynamical Systems: Theory and Applications, James T. Sandefur, 1990, Clarendon Press
Mathematical Thinking About Politics: An Introduction to Discrete Time Sytems, G. R. Boynton, 1980, Longman, Inc.

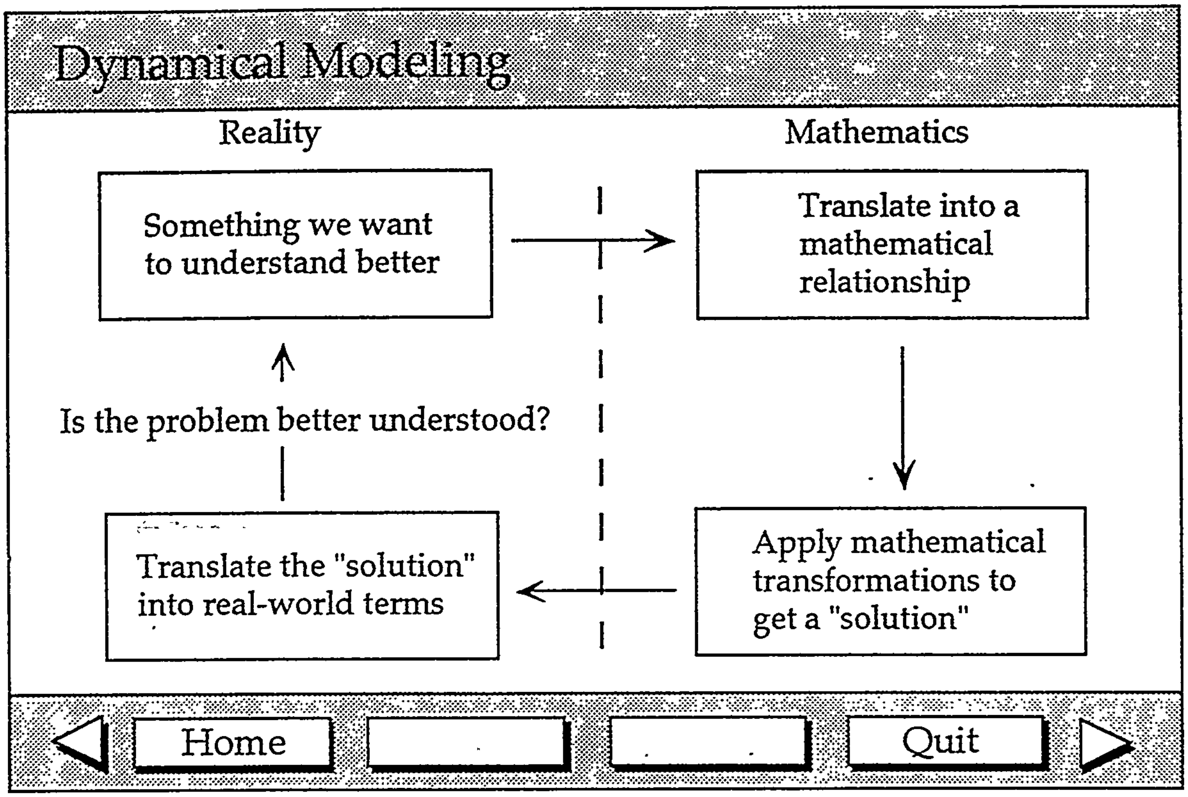
General Layout of a Card



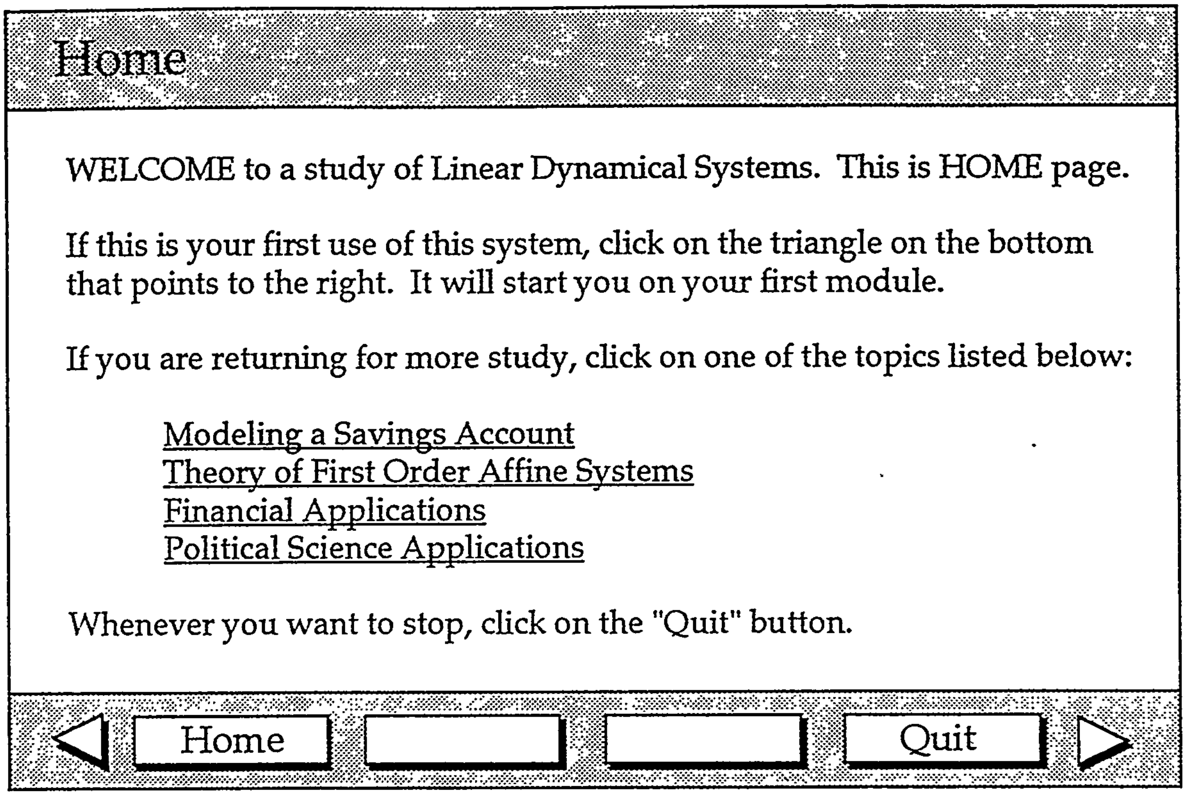
The presentation is given on a series of cards. The general layout of a typical card is shown above. At the top is a statement of the Topic under discussion. The discussion material for the card will be given in the blank space. This material may consist of sentences, diagrams, equations, pictures, or combinations of them all. For this report, pictures are excluded because the technology is not available.

The triangles and rectangles in the bottom strip furnish the hypertext capabilities of the presentation. The triangular buttons represent the ability to move to the immediately-preceding or immediately-following card. The button labeled "Home" represents the ability for the student to transfer to the beginning of the entire presentation. The button labeled "Quit" represents the ability for the student to terminate the presentation. The unlabeled buttons represent optional card locations that are dependent upon the context of a given card.

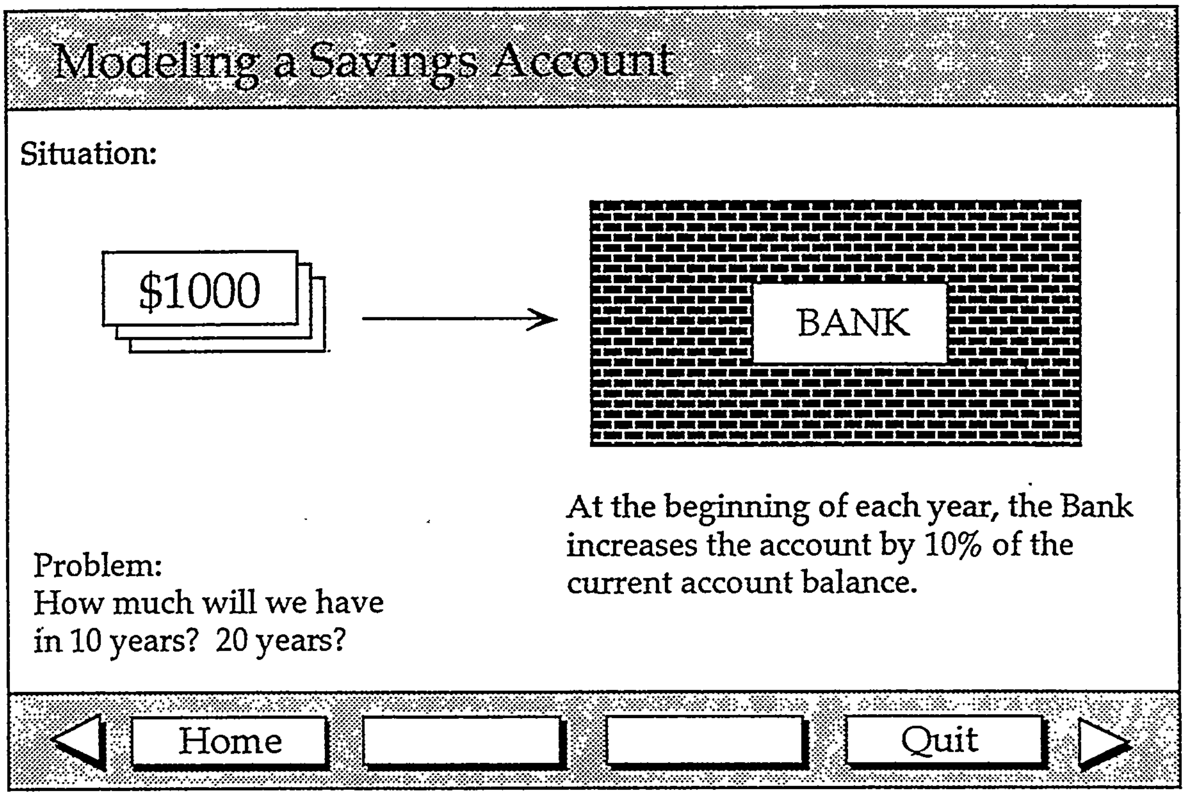
In addition, words in the discussion material which are underlined have the same function as buttons in the bottom strip. Those words allow students to transfer to other cards while in the middle of the presentation to get "refresher" information relevant to the concepts being presented. After reading the "refresher" information, the student can return to his original location in the presentation.



We begin our discussion with a graphic presentation of the essential steps in dynamical modeling.



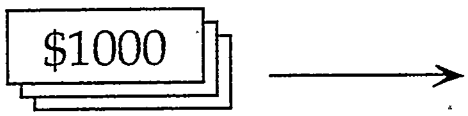
Although the "Home" page is not exactly the first page in our presentation, it serves as a traffic director. When the student begins, he will pass through this page and begin the first module — Modeling a Savings Account. On subsequent uses of the system, the student can jump immediately to the last module he was studying. At any time, the student can visit this page by clicking on the "Home" button at the bottom of the screen and be re-directed to any part of the presentation.



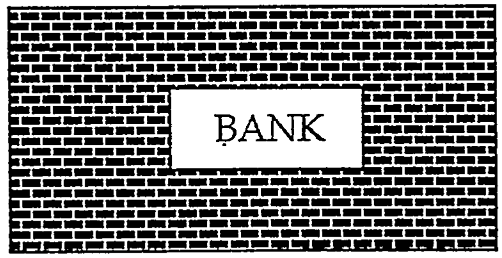
We have identified the problem. Next, we will translate into a mathematical relationship.

Modeling a Savings Account

Translation: *Let $A(n)$ be the amount in our account at the beginning of year n.*



$A(0)$ is the amount at time 0,
$A(0) = 1000$

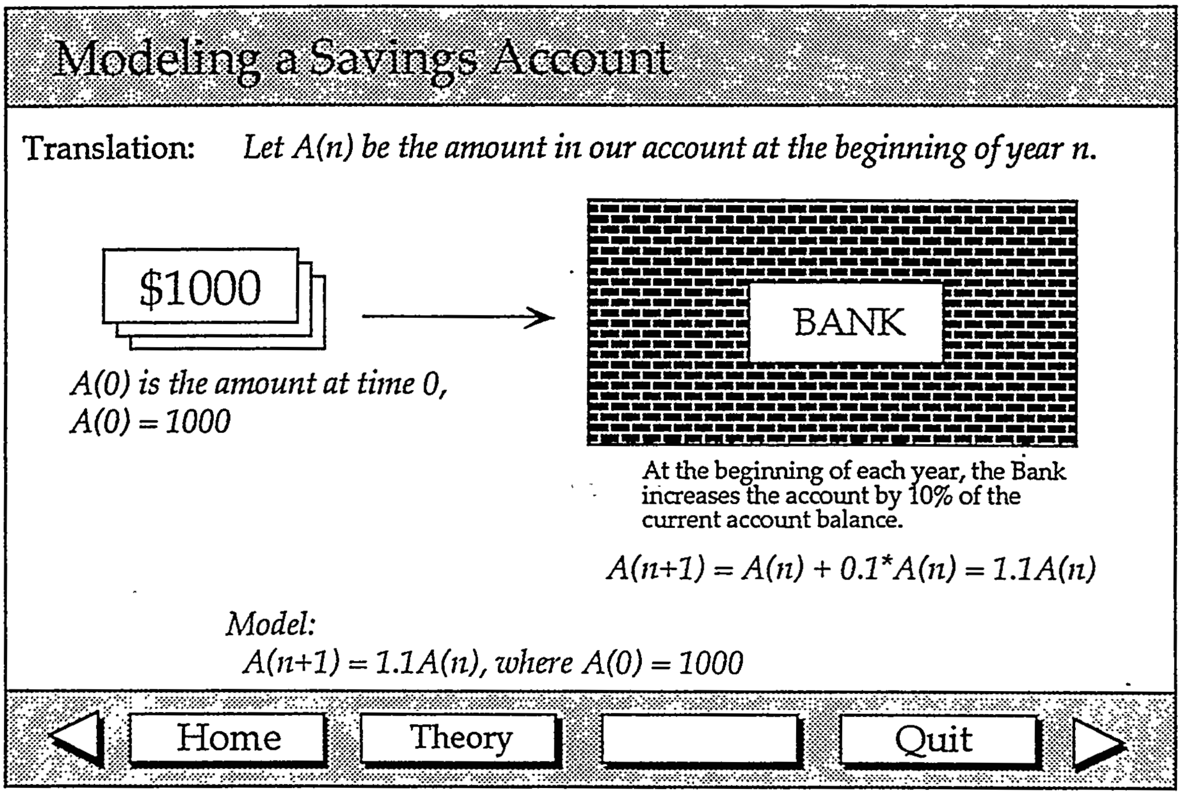


Problem:
How much will we have in 10 years? 20 years?
$A(10) = ?$ $A(20) = ?$

At the beginning of each year, the Bank increases the account by 10% of the current account balance.

$A(1) = 1000 + 0.1*1000 = 1000 + 100 = 1100$
$A(2) = 1100 + 0.1*1100 = 1100 + 110 = 1210$

Home | | | Quit



The model is complete. Note that a new button, "Theory," has been added. The student now has the option of clicking this button to begin to learn the terminology and theory associated with (linear) difference equations. If the student wishes, he may defer studying this information until specific terminology has been used. This "just in time" approach to imparting information is one of the important strengths of hypertext presentations.

Modeling a Savings Account

Translation: *Let A(n) be the amount in our account at the beginning of year n.*

Model: $A(n+1) = 1.1A(n)$, *where* $A(0) = 1000$

Mathematical Solution:

To get a general solution, let's replace the factor of 0.1 (10%) by I.
Thus, the bank adds 100I% of the account balance each year.

We know that $A(n+1) = A(n) + I^*A(n) = (1 + I)A(n)$, so let's take a few terms:

$A(1) = A(0) + I^*A(0) = (1 + I)A(0)$

$A(2) = A(1) + I^*A(1) = (1 + I)A(1) = (1 + I)[(1 + I)A(0)] = (1 + I)^2A(0)$

$A(3) = A(2) + I^*A(2) = (1 + I)A(2) = (1 + I)[(1 + I)^2A(0)] = (1 + I)^3A(0)$

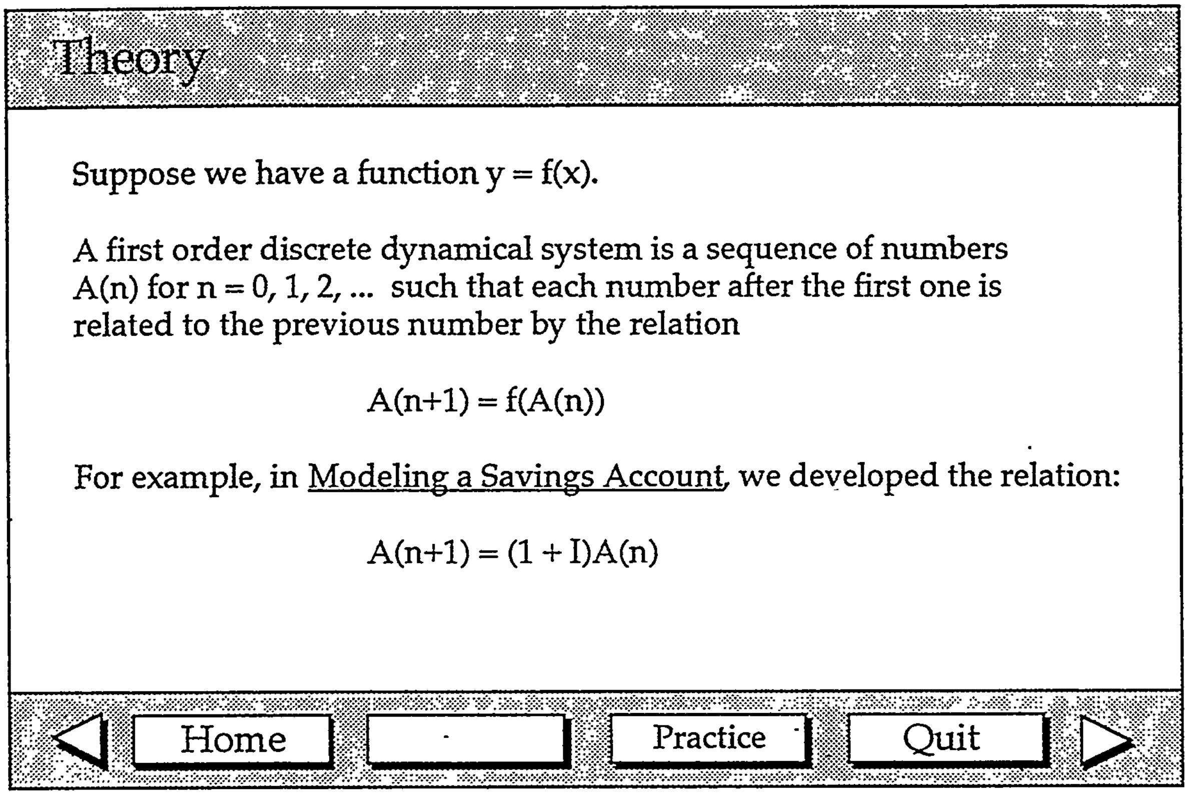
...

$A(k) = (1 + I)^kA(0)$

◁ Home | Theory | Practice | Quit ▷

Another button has been added, the "Practice" button. When the student exercises this option, he will be given modeling exercises with both hints and answers.

All stacks subsidiary to a given main topic are listed after topic. Thus, on the following pages, you will find the card(s) for "Theory" and "Practice," both of which are tailored to the topic: Modeling a Savings Account. In future topics, the terms "Theory" and "Practice" could also occur, but those cards will be tailored to that particular main topic.



This is the first example of a subsidiary stack. The next several cards discuss the terminology and theory appropriate to the model developed in "Modeling a Savings Account." In other modules, there will also be Theory stacks which will discuss the terminology appropriate to the models developed in those modules. The hypertext development system will keep the various Theory stacks separate even though I am using the same term on the button.

Theory

The order of a system is the number of initial values of A(n) that are needed for calculations.

For example, for $A(n+1) = (1 + I)A(n)$, A(1) is calculated as

$$A(1) = (1 + I)A(0)$$

At this point we can now calculate A(2), and then A(3), etc. We only needed to know A(0) to get going. There is no way to calculate A(0) because our index n cannot be negative.

Home | | Practice | Quit

Theory

How many initial values are needed for this dynamical system?

$$A(n+2) = 2A(n)$$

To get A(2), we need A(0): $A(2) = 2A(0)$
To get A(3), we need A(1): $A(3) = 2A(1)$

Once we have A(2) and A(3), however, we can calculate all remaining terms:

$$A(4) = 2A(2)$$
$$A(5) = 2A(3)$$
$$A(6) = 2A(4)$$

Two initial values are needed, so this is a SECOND order dynamical system.

◁ Home | | Practice | Quit ▷

Theory

The most general form of a first order linear dynamical system is

$$A(n+1) = rA(n), \text{ for } n = 0, 1, \ldots$$

A solution to a dynamical system is a function A(k) defined for all integers $k \geq 0$ that satisfies the dynamical system.

A general solution satisfies the dynamical system and involves a constant c which can be determined once an initial value is given.

A particular solution satisfies the dynamical system and, when $k = 0$, satisfies the equation $A(0) = a_0$.

Thus, the particular solution for the dynamical system given above is

$$A(k) = a_0 r^k$$

Home | Practice | Quit

Theory

Summary:
The most general form of a first order linear dynamical system is

$$A(n+1) = rA(n), \text{ for } n = 0, 1, \ldots$$

The particular solution for the first order linear dynamical system is

$$A(k) = a_0 r^k$$

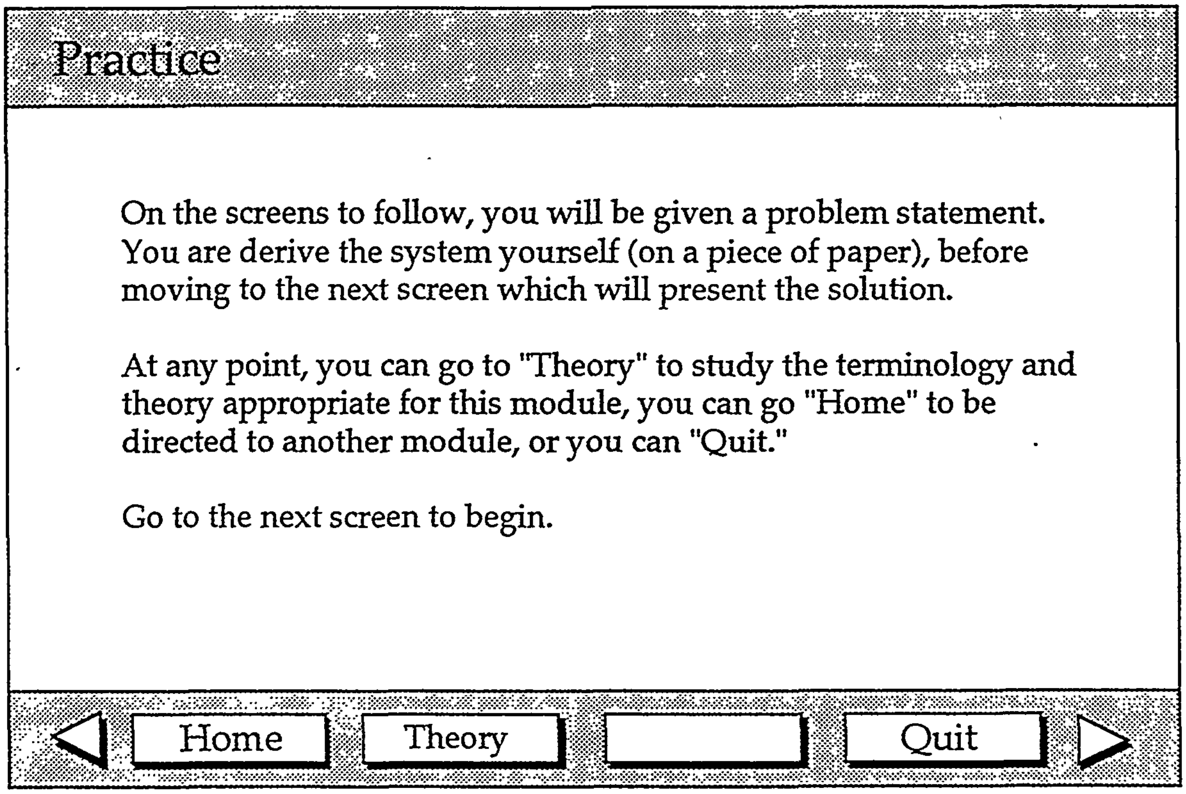
Note that the behavior of the solution of first order linear dynamical systems is determined by the size or sign of the constant r.

If $|r| < 1$, then the solution A(k) goes to zero, either exponentially or in an oscillatory fashion.

If $|r| > 1$, the solution goes exponentially to either positive or negative infinity.

Home | Practice | Quit

This is the end of the "Theory" stack for first order linear dynamical systems. At this point the student would either click "Practice," "Home" (to be directed to the next module), or "Quit."



This is the second example of a subsidiary stack, a stack for giving practice exercises to the student. The first card gives the directions, and subsequent cards present the practice exercises with illuminating commentary.

Practice

Suppose a bank pays 5 per cent interest on its savings accounts, compounded annually.

Let A(n) be the amount of money in the account at the beginning of year n.

(a) Formulate a dynamical system for the amount in the account in year n+1 using the amount in account in year n.

(b) Given that the initial deposit is A(0) = 200, find the amount in the account after 1, 2, 3, and 4 years.

(c) Find the general solution to the dynamical system.

Home | Theory | | Quit

And the solution:

Practice

Let A(n) be the amount of money in the account at the beginning of year n.

(a) amount in year $n+1$ = amount in year n + interest from year n

$$A(n+1) = A(n) + 0.05A(n)$$
$$A(n+1) = 1.05A(n)$$

(b)
$$A(1) = 1.05A(0) = 1.05(200) = 210$$
$$A(2) = 1.05A(1) = 1.05(210) = 220.5$$
$$A(3) = 1.05A(2) = 1.05(220.5) = 231.525$$
$$A(4) = 1.05A(3) = 1.05(231.525) = 243.10125$$

(c)
$$A(k) = (1.05)^k A(0)$$

Home | Theory | | Quit

Practice

Suppose a broker charges a 2 per cent service charge on the money in your savings account each year. Further, this broker makes bad investments each year and you do not earn any interest on your account.

Let A(n) be the amount of money in the account at the beginning of year n.

(a) Formulate a dynamical system for the amount in the account in year n+1 using the amount in account in year n.

(b) Given that the initial deposit is A(0) = 500, find the amount in the account after 1, 2, 3, and 4 years.

(c) Find the general solution to the dynamical system.

Home | Theory | | Quit

Practice

Let A(n) be the amount of money in the account at the beginning of year n.

(a) amount in year $n+1$ = amount in year n - amount lost from year n

$A(n+1) = A(n) - 0.02A(n)$

$A(n+1) = 0.98A(n)$

(b) $A(1) = 0.98A(0) = 0.98(500) = 490$

$A(2) = 0.98A(1) = 0.98(490) = 480.2$

$A(3) = 0.98A(2) = 0.98(480.2) = 470.596$

$A(4) = 0.98A(3) = 0.98(470.496) = 461.18408$

(c) $A(k) = (0.98)^k A(0)$

Home | Theory | | Quit

Practice

Suppose a bank pays 8 per cent each year on its checking accounts, but it also deducts 40 dollars per year as a service charge (after first adding on the interest).

Let A(n) be the amount of money in the account at the beginning of year n.

(a) Given that the initial deposit is A(0) = 1000 dollars, find the amount in the account after 1, 2, and 3 years. (Assume no checks were written)

(b) Formulate a dynamical system for the amount in the account in year n+1 using the amount in the account in year n.

Home | Theory | | Quit

Practice

Let A(n) be the amount of money in the account at the beginning of year n.

(a) $A(1) = 1000 + 0.08(1000) - 40 = 1000 + 80 - 40 = 1040$
$A(2) = 1040 + 0.08(1040) - 40 = 1040 + 83.2 - 40 = 1083.2$
$A(3) = 1083.2 + 0.08(1083.2) - 40 = 1083.2 + 86.656 - 40 = 1129.856$

(b) amount in year $n+1$ = amount in year n + interest for year n - service charge

$$A(n+1) = A(n) + 0.08A(n) - 40$$
$$A(n+1) = 1.08A(n) - 40$$

Note: This is a first order system (how many initial values are needed?), but we don't yet know how to find the general solution for this type of system.

Home | Theory | | Quit

Practice

Suppose that you borrow 2000 dollars from a friend. You agree to add 1 per cent interest each month to the amount of the load that is still outstanding and also to pay your friend 150 dollars each month. Your friend insists that the interest is first added on to what you owe and then your 150 dollar payment is subtracted.

Let A(n) be the amount of money owed at the beginning of month n.

(a) Formulate a dynamical system for the amount owed in month n+1 using the amount owed in month n.

Home | Theory | | Quit

Practice

Let A(n) be the amount of money owed at the beginning of month n.

amount owed in month *n+1* = amount owed in month *n*
+ interest on amount owed in month *n*
- payment to friend

$$A(n+1) = A(n) + 0.01A(n) - 150$$

$$A(n+1) = 1.01A(n) - 150$$

Home | Theory | | Quit

Practice

Situation #1:
Suppose it costs 120 dollars plus 20 cents a mile to rent a car for a week. Let A(n) represent the total cost for renting the car if you drive for a total of n miles.

Situation #2:
To make a telephone call to New York City costs 45 cents for the first minute and 33 cents for each additional minute. Let A(n) represent the cost of a call lasting for n minutes.

For each Situation, you are to write a dynamical system relating A(n+1) in terms of A(n). Before you write the general system, write expressions, for A(0), A(1), A(2), and A(3).

◁ Home | Theory | | Quit ▷

Practice

Situation #1: A(n) is the total cost after driving n miles.

$A(0) = 120$
$A(1) = A(0) + 20$
$A(2) = A(1) + 20$
$A(3) = A(2) + 20$

$A(n+1) = A(n) + 20 \quad (n = 0, \ldots)$

Situation #2: A(n) is the cost of a call lasting n minutes.

$A(0) = 0$
$A(1) = 45$
$A(2) = A(1) + 33$
$A(3) = A(2) + 33$

$A(n+1) = A(n) + 33 \quad (n = 1, \ldots)$

◁ Home | Theory | | Quit ▷

Practice

Situation #1:
Let A(n) be the number of gallons of gas left in a car after driving n miles. The car originally had A(0) = 12 gallons, and it goes 20 miles per gallon of gas [Note: That is 0.05 gallons per mile driven.] Write a dynamical system describing the amount of gas left after driving n+1 miles in terms of the amount of gas left after driving n miles.

Situation #2:
Suppose that a person takes a pill containing 200 milligrams of a drug every 4 hours, and assume that the drug goes into the bloodstream immediately. Also assume that every 4 hours the body eliminates 20 per cent of the drug that is in the bloodstream. Develop a dynamical system describing the amount A(n) of the drug in the bloodstream after taking the *n*th pill.

◁ Home | Theory | | Quit ▷

Practice

Situation #1: A(n) is the number of gallons left in a car after driving n miles when the car consumes 0.05 gallons per mile driven

$A(0) = 12$
$A(1) = A(0) - 0.05$
$A(2) = A(1) - 0.05$
$A(n+1) = A(n) - 0.05 \qquad (n = 0, ...)$

Situation #2: A(n) is the amount of drug in the bloodstream after taking the nth pill

$A(0) = 0$
$A(1) = 200$
$A(2) = 200 + 0.8A(1)$
$A(3) = 200 + 0.8A(2)$
$A(n+1) = 200 + 0.8A(n) \qquad (n = 1, ...)$

◁ Home | Theory | | Quit ▷

Theory of First Order Affine Systems

In the section Modeling a Savings Account you were introduced to the plain vanilla first order linear dynamical system

$$A(n+1) = rA(n),$$

which has the solution $A(k) = r^k A(0)$.

That system didn't have many realistic applications. But now with a minor modification, we have the first order AFFINE dynamical system:

$$A(n+1) = rA(n) + b \qquad \text{(b is a constant).}$$

For example, you have a savings account with annual interest and you add a constant amount to the account at the beginning of each year. That situation can be modeled with a first order affine dynamical system.

But models are no good if we can't get a solution. So, first we need the solution of: $A(n+1) = rA(n) + b$.

Home | | | Quit

This section has no interaction, so two cards are displayed on each page.

Theory of First Order Affine Systems

A real advantage of working with difference equations is that the mathematics is so accessible and the approach is so straight-forward. Now, how did we solve $A(n+1) = rA(n)$? We wrote some terms and saw a general pattern. Let's do the same thing for

$$A(n+1) = rA(n) + b.$$

$$A(1) = rA(0) + b$$
$$A(2) = rA(1) + b = r[rA(0) + b] + b = r^2A(0) + rb + b$$
$$= r^2A(0) + b(1 + r)$$
$$A(3) = rA(2) + b = r[r^2A(0) + rb + b] + b$$
$$= r^3A(0) + r^2b + rb + b$$
$$= r^3A(0) + b(1 + r + r^2)$$

Home | | | Quit

Theory of First Order Affine Systems

Since $A(3) = r^3A(0) + b(1 + r + r^2)$, we can see the form of A(k):

$$A(k) = r^kA(0) + b(1 + r + r^2 + ... + r^{k-1}).$$

But, what can be done with that god-awful term in parentheses? Fortunately, a clever person noticed that:

$$1 + r + r^2 + ... + r^{k-1} = (1 - r^k)/(1 - r)$$

Thus, $A(k) = r^kA(0) + b[(1 - r^k)/(1 - r)]$, which can be simplified to

$$A(k) = r^k[A(0) - (b/(1-r))] + [b/(1-r)]$$

This form is not so bad, and it contains the repeating term, b/(1-r). Does that have a significance?

Home | | | Quit

Theory of First Order Affine Systems

To see the significance of the repeating term, b/(1-r), we must take a slight detour from our derivation of the solution of first order affine systems.

A first order affine system, in fact any difference equation, is said to be in EQUILIBRIUM if every term has the same value — $A(0) = A(1) = A(2) = ...$

Let the equilibrium value be a. Then, for $A(n+1) = rA(n) + b$, both A(n+1) and A(n) will equal a. Thus

$$a = ra + b$$
$$(1-r)a = b$$
$$a = b/(1-r)$$

Our repeating term is the EQUILIBRIUM value for the first order affine system. And note that the system has no equilibrium if r = 1.

Home | | | Quit

Theory of First Order Affine Systems

So we also need to consider the special case of the first order affine system with $r = 1$: $A(n+1) = A(n) + b$.

$$A(1) = A(0) + b$$
$$A(2) = A(1) + b = A(0) + b + b = A(0) + 2b$$
$$A(3) = A(2) + b = A(0) + 2b + b = A(0) + 3b$$

So, $A(k) = A(0) + kb$

In other words, all we're doing in this system is adding the constant b at each step.

Home | | | Quit

Theory of First Order Affine Systems

Thus, the first order affine system:

$$A(n+1) = rA(n) + b$$

has one of the following solutions:

when $r \neq 1$:

$$A(k) = r^k[A(0) - a] + a$$

where $a = b/(1-r)$

when $r = 1$: $A(k) = A(0) + kb$

Home | | | Quit

Financial Applications

In this section, we will use the first order affine dynamical system to model the following applications:

- Certificates of Deposit
- Annuities
- How the Lottery Works
- Loan Amortization

The topics are independent of one another. Just click on the one you want. To return here, click on "Financial" at the bottom of the page.

Home | Financial | | Quit

This page serves as the "home" page for all financial applications

Certificates of Deposit

A Certificate of Deposit (CD) is a financial instrument where you give the bank a sum of money and they promise to give you a guaranteed percentage of interest in return. But, you can't get your money back in the interim. It's tied up for the lifetime of the CD.

Relatively short term CDs are now paying 6%, so let's say we could get long term CDs paying 7%. Further, you have a long worklife ahead of you.

So, if you deposited the same amount each compounding period (say, every quarter), and the CD is paying 7% interest, compounded quarterly, how much would you need to deposit every 3 months to have a million dollars in 20 years? in 30 years? in 40 years?

Home | Financial | | Quit

Certificates of Deposit

Model:

the amount in quarter n+1 = amount from quarter n
+ quarterly interest on amount from quarter n
+ deposit for quarter n+1

$$A(n+1) = A(n) + (0.07/4)A(n) + b$$
$$A(n+1) = 1.0175A(n) + b$$

Solution:

$$a = b/(1-r) = b/(1 - 1.0175) = b/(-0.0175) = -57.14\,b$$
$$A(0) = b$$
$$A(k) = r^k[A(0) - a] + a = (1.0175)^k[b - (-57.14\,b)] - 57.14\,b$$
$$A(k) = (1.0175)^k[58.14\,b] - 57.14\,b$$

Home | Financial | | Quit

Certificates of Deposit

If we want \$1,000,000 [this will be A(k)] in 20 years (k is 20 x 4 quarters = 80 quarters):

$$1{,}000{,}000 = (1.0175)^{80}[58.14\,b] - 57.14\,b$$
$$1{,}000{,}000 = 175.79\,b$$
$$b = \$5{,}688.55 \quad \text{(or \$22,754.20 annually)}$$

For 30 years:

$$1{,}000{,}000 = (1.0175)^{120}[58.14\,b] - 57.14\,b$$
$$1{,}000{,}000 = 409.1\,b$$
$$b = \$2{,}444.42 \quad \text{(or \$9,777.68 annually)}$$

For 40 years:

$$1{,}000{,}000 = (1.0175)^{160}[58.14\,b] - 57.14\,b$$
$$1{,}000{,}000 = 876.08\,b$$
$$b = \$1141.45 \quad \text{(or \$4,565.80 annually)}$$

Home | Financial | Quit

Annuities

Let's say you now have the $1,000,000 that you saved for for 40 years and you're now 65 years old. Further, let's say that you can get a guaranteed 5% at the bank (remember, you can't touch the money if it's in CDs).

You could live off the annual interest of $50,000 or you could take $50,000 out of the million and put the rest into 1-year CDs at say 6% and get $57,000. The nice thing here is that when you die, you would be leaving the whole million to your kids.

On the other hand, SCREW THE KIDS! In fact, you want $75,000 per year because YOU HAVE NEEDS that have been deferred by saving $5,000 per year for 40 years.

Don't go to the next page until you have written and solved a first order affine dynamical system model to determine how long the million will last.

Home | Financial | | Quit

Annuities

Model:

We assume that we start with $A(0) = 1,000,000$ and receive an annual interest of 5%. Further, we are taking a lump sum of $75,000 each year. (The calculations are nicer if we forego the first year.)

If $A(n)$ is the amount of money in our account at the beginning of year n, we want to know the value of k such that $A(k) = 0$.

amount in year n + 1 = amount in year n
+ interest for year n
- withdrawal of $75,000

$$A(n+1) = A(n) + 0.05A(n) - 75000 = 1.05A(n) - 75000$$

Home | Financial | | Quit

Annuities

Solution: $A(k) = r^k[A(0) - a] + a$

$r = 1.05$
$A(0) = 1{,}000{,}000$
$b = -75{,}000$
$a = b/(1-r) = (-75000)/(1 - 1.05) = 1{,}500{,}000$

$A(k) = (1.05)^k[1{,}000{,}000 - 1{,}500{,}000] + 1{,}500{,}000 = 0$
$500{,}000(1.05)^k = 1{,}500{,}000$
$(1.05)^k = 3$

At this point you can use logarithms, calculators, spreadsheets, whatever to find that k is between 22 and 23. In other words, you better kick the bucket about 87 or your kids will do the job themselves!

Home | Financial | | Quit

How the Lottery Works

Let's now look at another annuity — the State Lottery. This again is an annuity. A lump sum of money is put in an interest-bearing account and the interest (and part of the principal) is disbursed over a period of years. In this case, it's a fixed number of years — 20.

Our interest is in finding out how much the State "makes" on the Lottery money. That is, does all of the money go to the winners?

So, now your task is to write a first order affine dynamical system to determine how much money should be put in an account yielding 8% annual interest (the State can get a good deal!) and be depleted after 20 annual payments of $50,000 each (that's a million-dollar lottery win).

Home | Financial | | Quit

How the Lottery Works

Model:

We are looking for the starting amount, A(0). That amount will yield an annual interest of 8%. Further, we are taking a lump sum of $50,000 each year. (The calculations are nicer if we forego the first year.)

If A(n) is the amount of money in our account at the beginning of year n, we want to know the value of k such that A(k) = 0.

amount in year n + 1 = amount in year n
+ interest for year n
- withdrawal of $50,000

$$A(n+1) = A(n) + 0.08A(n) - 50000 = 1.08A(n) - 50000$$

Home | Financial | | Quit

How the Lottery Works

Solution: $A(k) = r^k[A(0) - a] + a$

$r = 1.08$
$k = 20$
$A(0) = ?$
$b = -50{,}000$
$a = b/(1-r) = (-50000)/(1 - 1.08) = 625{,}000$

$A(k) = (1.08)^{20}[A(0) - 625{,}000] + 625{,}000 = 0$
$4.66A(0) - 2{,}912{,}500 + 625{,}000 = 0$
$4.66A(0) = 2{,}287{,}500$
$A(0) = 490{,}879$

In other words, the state gets about half of the money for itself! Does this begin to explain why state lotteries are so attractive to politicians?

Home | Financial | | Quit

Loan Amortization

You got a student loan at 10%. At the time you graduate, you owe $15,000. You're given 3 options: 5 years, 10 years, or 15 years. Which should you take?

You should be getting pretty good at this. Write a linear affine dynamical system model to determine the monthly payment. Use a monthly interest rate of 0.1/12.

Home | Financial | | Quit

Loan Amortization

Model:

A(n) is the amount of money that we owe at the beginning of each month, and we want to know the value of b such that A(k) = 0 for k = 60, 120, or 180.

amount owed in month n + 1 = amount owed in month n
+ interest on amount owed in month n
- monthly payment

$$A(n+1) = A(n) + 0.0083A(n) - b = 1.0083A(n) - b$$

Home | Financial | | Quit

Loan Amortization

Solution: $A(k) = r^k[A(0) - a] + a$

$r = 1.0083$
$k = 60, 120$, or 180 (I'll show the details for $k = 60$)
$A(0) = 15{,}000$
$b = ?$
$a = b/(1-r) = b/(1 - 1.0083) = -120.5\,b$

$$A(k) = (1.0083)^{60}[15{,}000 + 120.5\,b] - 120.5\,b = 0$$

$k = 60$:	$b = \$318.38$	total amount paid = \$19,102
$k = 120$:	$b = \$197.88$	total amount paid = \$23,746
$k = 180$:	$b = \$160.80$	total amount paid = \$28,944

Pay the loans, but make the sacrifice, if you can, and pay it quickly.

Political Science Applications

As one might imagine, the applications of dynamical systems to be found in political science are much wordier than those of financial applications. Therefore, these applications tend to drone on from page to page. Please be patient, though, because the material may seem old familiar but the treatment is quite novel.

I have extracted applications on:

- Partisanship
- Political Attention
- Birth and Death of State Agencies
- Unemployment and Encumbency

At any point the "PoliSci" key will return you to this page, from which you can click onto another application. "Home" and "Quit" are always available.

Home | Quit

This is the "home" page for the Political Science Applications. The first application, "Partisanship," is carried through to a model and its solution. The other applications are modeling opportunities for the students as data is not available to create an actual model. One of the intriguing elements of Boynton's work in the application of dynamical systems to political science is his use of models for qualitative analysis.

Partisanship

A fact about American politics that is part of the general lore is that individuals divide themselves into Democrats and Republicans. We call this partisanship.

The strength of partisanship varies; some people think of themselves as very strong Democrats or Republicans, and some as not so strong Democrats or Republicans.

One might think that strength of partisanship is related to age. That is, younger adults (18 - 35) are less likely to think of themselves as strong partisans than are older adults (55 - 70).

Home | PoliSci | Quit

Partisanship

It is also known that partisans of one party are more likely to vote for candidates of that party than they are to vote for candidates of the other party. After all, that's what "partisan" means.

But all studies of partisanship among children show that partisanship has its beginnings well before the first vote, and that the partisanship of children is directly linked to the partisanship of their parents.

In the 1950s research was conducted in the United States and France on partisanship in the respective adult populations. The level of partisanship was vastly different in the two countries. In the United States, 75% of the adult population thought of themselves as partisans of one or the other party. In France, only 24% of the population felt similar partisan attachments.

Home | PoliSci | Quit

Partisanship

The individuals interviewed were also asked about the partisan attachments of their parents. In the U. S., 76% of those interviewed could remember the partisan attachment of their parents; in France, only 25%.

But when the data was analyzed, an unexpectedly similar pattern of political learning in the two countries is revealed:

	Know Father's Party		Don't Know Father's Party	
	France	US	France	US
Partisan	79%	82%	48%	51%
Not partisan	21%	18%	52%	49%

Of those who could remember their father's party, 80% (regardless of country) were themselves partisan; of those who could not remember, 50% were partisan.

Home | PoliSci | | Quit

Partisanship

What is needed is a formal representation of this learning process which will provide insight into the way levels of partisanship will change over time.

Let $A(n)$ represent the proportion of partisans in the current generation. Then, $A(n-1)$ is the proportion of partisans in the previous generation, the generation of their parents. Further, $[1 - A(n-1)]$ is the proportion of the previous generation who were not partisans.

Then, from the data given on the previous card, we know that 80% of the parents who were partisan will have children who are partisan, and 50% of the parents who were not partisan will have children who are partisan.

Thus, $$A(n) = 0.8A(n-1) + 0.5[1 - A(n-1)] = 0.3A(n-1) + 0.5$$

Home | PoliSci | | Quit

Partisanship

From the presentation given in Theory of First Order Affine Systems, we know how to solve this system:

$$A(n) = 0.3A(n-1) + 0.5$$

We are, however, used to seeing such systems advanced one period:

$$A(n+1) = 0.3A(n) + 0.5$$

Solution:

$$r = 0.3$$
$$a = b/(1-r) = 0.5/(1-0.3) = 0.714$$
$$A(k) = r^k[A(0) - a] + a$$
$$A(k) = (0.3)^k[A(0) - 0.714] + 0.714$$

As k gets large, $(0.3)^k$ will go to zero, leaving A(k) to apprach 0.714. Thus, our model predicts partisanship will increase in France and drop in the US.

Home | PoliSci | | Quit

Political Attention

There was a time when most of the food consumed was produced either by the individual family unit or by one's neighbor. Bread was baked each day by the local bakery. Meat was grown and slaughtered locally. In general, food production was a small-scale local operation.

As that changed, farms became larger, food was transported over longer and longer distances, and buyers and sellers became more concerned about the appearance of the food product. In order to make farming more productive, chemical fertilizers were used and poisons developed and used to kill bugs, diseases, etc. Chemicals were added to foodstuffs to preserve them and make them more attractive.

Then it was discovered that many additives, chemicals, and poisons are potentially harmful to those who eat the food.

Home | PoliSci | | Quit

Political Attention

Over the past thirty years there has been increasing political attention devoted to this problem.

If political attention to chemicals in the food is the output of the system, then the amount of chemicals in food can be thought of as the input.

When political officials compete for office through elections, they are concerned about the problems of interest to their constituents. This can be represented by a constant multiplied by the input; the constant represents citizen concern about health.

There is, however, a limit on the amount of time that government can devote to any given problem. There are many problems; attention has to be spread around. The press of other business can be represented by a constant multiplied by attention to chemicals in food in the past.

Home | PoliSci | | Quit

Political Attention

For purposes of model development, the preceding discussion can be summarized as follows:

The rule that transforms the amount of chemicals in food (the input) into the political attention to chemical in food (the output) can be summarized as:

current political attention is produced by the concern of citizens about health times the current amount of chemicals in food plus the press of other business times the attention given to this problem at the immediately past time period.

Use the following definitions to formulate this model:

$A(n)$	amount of political attention to chemicals in food in period n
$u(n)$	amount of chemicals in food in period n
α	press of other business
β	public concern about health

Home | PoliSci | | Quit

Political Attention

$A(n)$	amount of political attention to chemicals in food in period n
$u(n)$	amount of chemicals in food in period n
α	press of other business
β	public concern about health

current political attention [A(n)] is produced by [=]
the concern of citizens about health [β]
times the current amount of chemicals in food [u(n)] plus [+]
the press of other business [α] times the attention given to this problem at the immediately past time period [A(n-1)].

$$A(n) = \beta u(n) + \alpha A(n-1)$$

Note that unless we are willing to make u(n) a constant, we cannot solve this model. This model is a type known as nonhomogeneous.

Home | PoliSci | | Quit

Birth and Death of State Agencies

The sunset laws are one answer to big government and red tape. Sunset laws are laws that set up executive agencies for a specified period of time. When that time has elapsed, the governor and the legislature review the work of the agency to determine if there continues to be a need for the agency. It is assumed by the proponents of these laws that at least some agencies will, through this process, go out of existence.

The input is the number of new agencies set up in a given year. The number of new agencies is added to those agencies already in existence, that is, all agencies that existed in the previous year. From this is subtracted those agencies for which the death knell sounds after their review (say 7 years), but since this is not likely to be all agencies, this term must be multiplied by a constant representing the proportion that are not continued.

◁ Home | PoliSci | | Quit ▷

Birth and Death of State Agencies

Model this situation given the following definitions:

$A(n)$	number of executive agencies in year n
$u(n)$	number of agencies established in year n
α	proportion of reviewed agencies which go out of existence

The input is the number of new agencies set up in a given year. The number of new agencies is added to those agencies already in existence, that is, all agencies that existed in the previous year. From this is subtracted those agencies for which the death knell sounds after their review (say 7 years), but since this is not likely to be all agencies, this term must be multiplied by a constant representing the proportion that are not continued.

◁ Home | PoliSci | | Quit ▷

Birth and Death of State Agencies

Model this situation given the following definitions:

A(n)	number of executive agencies in year n
u(n)	number of agencies established in year n
α	proportion of reviewed agencies which go out of existence

[A(n+1)] = The input is the number of new agencies set up in a given year. The number of new agencies is added to those agencies already in existence, that is, all agencies that existed in the previous year. [u(n+1) + A(n)]
From this is subtracted those agencies for which the death knell sounds after their review (say 7 years), but since this is not likely to be all agencies, this term must be multiplied by a constant representing the proportion that are not continued. [- αA(n-7)]

$$A(n+1) = u(n+1) + A(n) - \alpha A(n-7)$$

◁ Home | PoliSci | | Quit ▷

Unemployment and Incumbency

In the past 60 years the public has come to assign principal responsibility for managing the performance of the economy to government. As a result, fluctuations in unemployment have consequences for election outcomes.

The Democrat party is believed by much of the public to be more effective in producing "good times" than is the Republican party. Thus, when unemployment increases, the public is likely to vote in Democrats and vote out Republicans. When unemployment decreases, the electoral prospect of Democrats is diminished and the prospects for Republicans brightens.

Home | PoliSci | | Quit

Unemployment and Incumbency

This argument can be formalized by treating the proportion of Democrats elected to Congress as the output of the system.

The input is change in unemployment — not the level of unemployment, but the change in the level of unemployment. Change in unemployment can be represented as the current level of unemployment from which is subtracted unemployment at the immediately pas period.

Once a Congressman is in office, his or her chance of being reelected is rather good. There is advantage in incumbency. The advantages that accrue to incumbents can be represented by a constant, which is multiplied by the proportion of Congressmen who were elected at the last election.

Finally, there must be a constant which represents citizen concern about change in unemployment.

Home | PoliSci | | Quit

Unemployment and Incumbency

Thus, the current proportion of Congressment who are Democrats is produced by the advantage of incumbency multiplied by the proportion of Democrats elected in the last election plus the concern of citizens about change in unemployment multiplied by unemployment now, from which is subtracted unemployment at the last time period.

Model this description of the situation using the following definitions:

A(n)	proportion of Congressmen who are Democrats in session n
u(n)	unemployment at the time of the election for session n
α	advantage of incumbency
β	citizen concern about change in unemployment

◁ Home | PoliSci | | Quit ▷

Unemployment and Incumbency

A(n)	proportion of Congressmen who are Democrats in session n
u(n)	unemployment at the time of the election for session n
α	advantage of incumbency
β	citizen concern about change in unemployment

Thus, the current proportion of Congressment who are Democrats is produced by [A(n) =] the advantage of incumbency multiplied by the proportion of Democrats elected in the last election [αA(n-1)] plus [+] the concern of citizens about change in unemployment [β] multiplied by unemployment now, from which is subtractedunemployment at the last time period. [u(n) - u(n-1)]

$$A(n) = \alpha A(n-1) + \beta[u(n) - u(n-1)]$$

◁ Home | PoliSci | | Quit ▷

Design of a Computer-Based Presentation on Mathematical Modeling using Difference Equations

Volume II. Nonlinear Dynamical Systems

Dr. Thomas W. Mason
Department of Mathematics
Florida A&M University
Tallahassee, FL 32307

Volume II. Nonlinear Dynamical Systems

Table of Contents

Introductory Remarks	45
General Layout of a Card	46
Dynamical Modeling	47
Home	48
Logistic Equation Models	49
Limitations of Exponential Growth	50
Logistic Model	53
Logistic Model of Population Growth	55
Logistic Model of Belief Systems	56
Equilibria and Cobwebbing	59
Analysis of the Logistics Models	67
Analysis of the Model of Population Growth	68
Analysis of the Model of Belief Systems	78

Introductory Remarks

This volume — Nonlinear Dynamical Systems — and its companion volume, Linear Dynamical Systems, are written for the audience of a student who is moderately literate in undergraduate mathematics, but almost certainly has not taken a course in difference equations. The underlying thesis of this effort is that mathematical modeling can be introduced, understood, and mastered by such students if difference equations are used rather than differential equations. Such an approach does not, of course, preclude the eventual, or even simultaneous, use of differential equations. However, the relative simplicity of difference equations allows a student to concentrate on — and participate in — the modeling process, which is not the case with the typical differential equation approach.

The subject matter for the presentation has been chosen with the adolescent student in mind. It is the experience of the author that such students are primarily absorbed with matters focusing on either money or sex. Hence, Volume 1 presents a difference equation approach to linear dynamical systems concentrating on models of personal finance with a short venture into politics. Volume 2 extends the approach to nonlinear dynamical systems focusing on models of population.

The presentation shows how the material could be presented to students using a hypertext system of the genre of *HyperCard™*. However, the material does not currently exist in a hypertext medium. Therefore, the computer screen displays are simply representative of the sorts of options that students could be given as they study, explore, and master Linear Dynamical Systems.

The material presented in this volume was excerpted from the following references:

HyperCard Reference Manual, Apple Computer, Inc., 1993

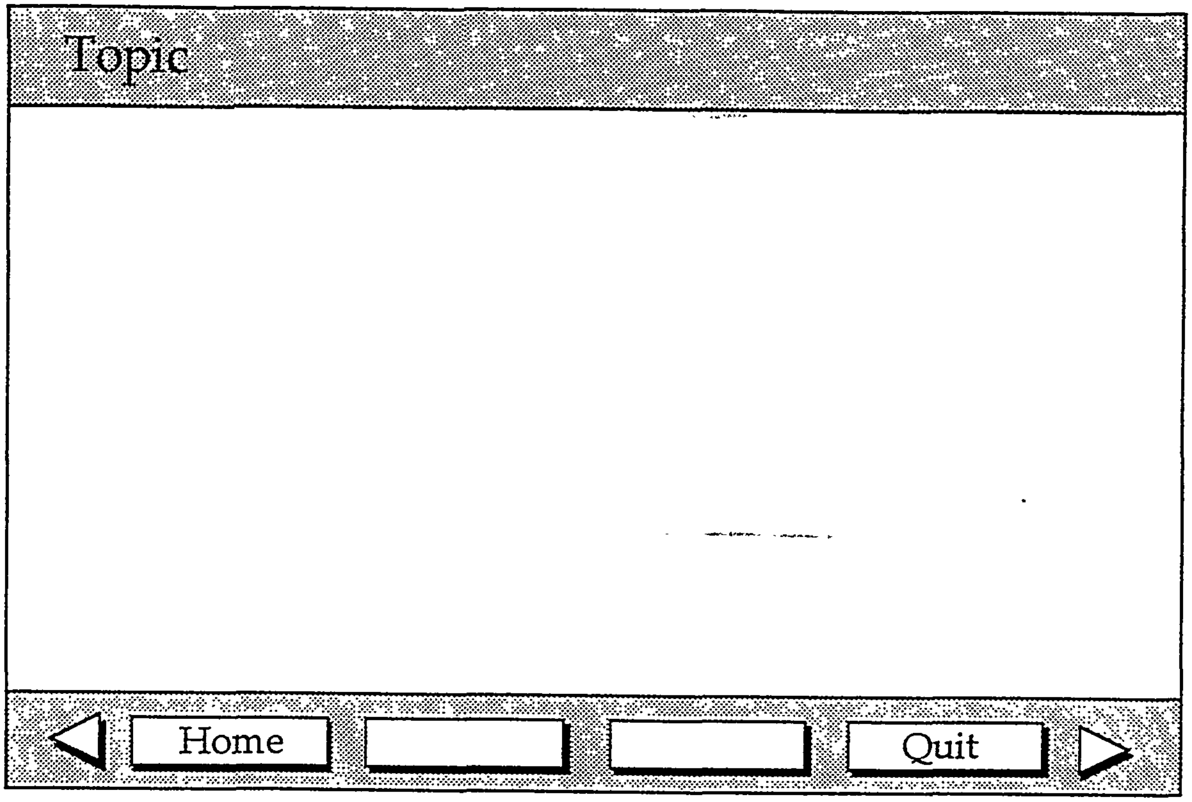
MacMath 9.2: A Dynamical Systems Software Package for the Macintosh™ , John Hubbard and Beverly West, 1993, Springer-Verlag

Discrete Dynamical Systems: Theory and Applications, James T. Sandefur, 1990, Clarendon Press

Differential Equations and Their Applications, Martin Braun, 1993, Springer-Verlag

Growth and diffusion Phenomena: Mathematical Frameworks and Applications, Robert Banks, 1994, Springer-Verlag

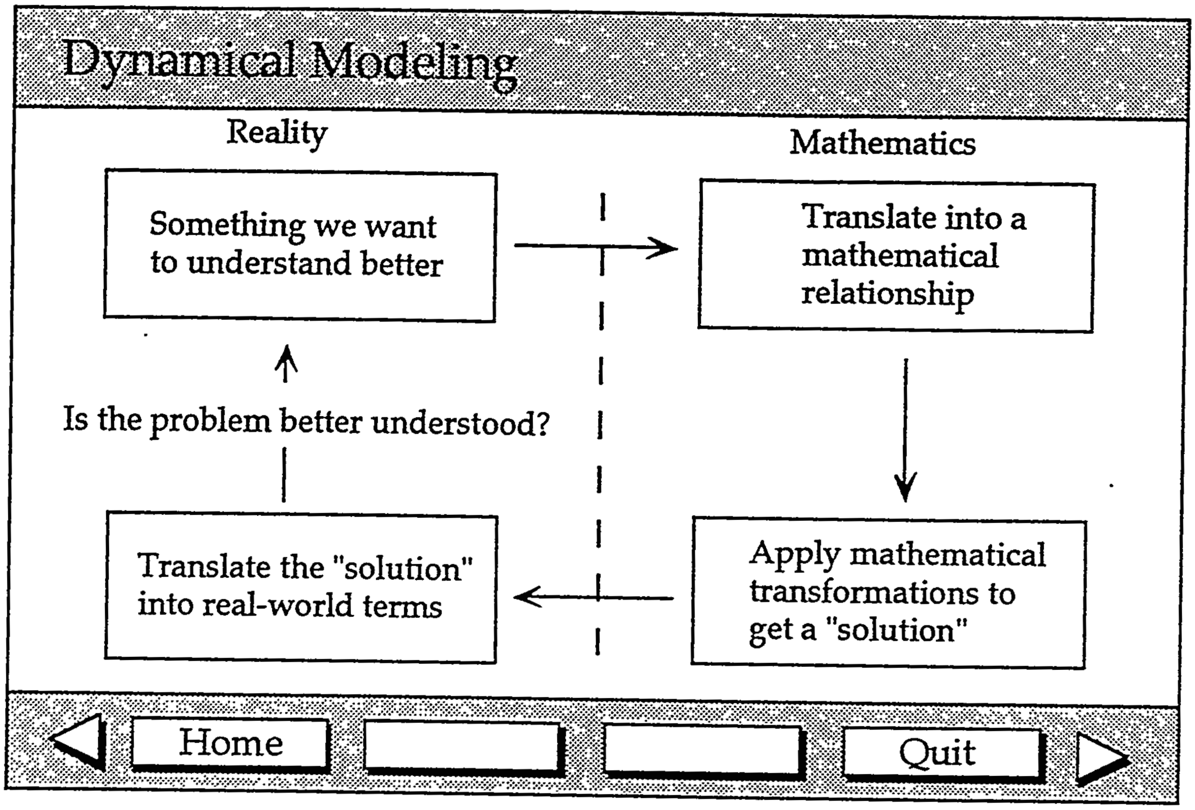
General Layout of a Card



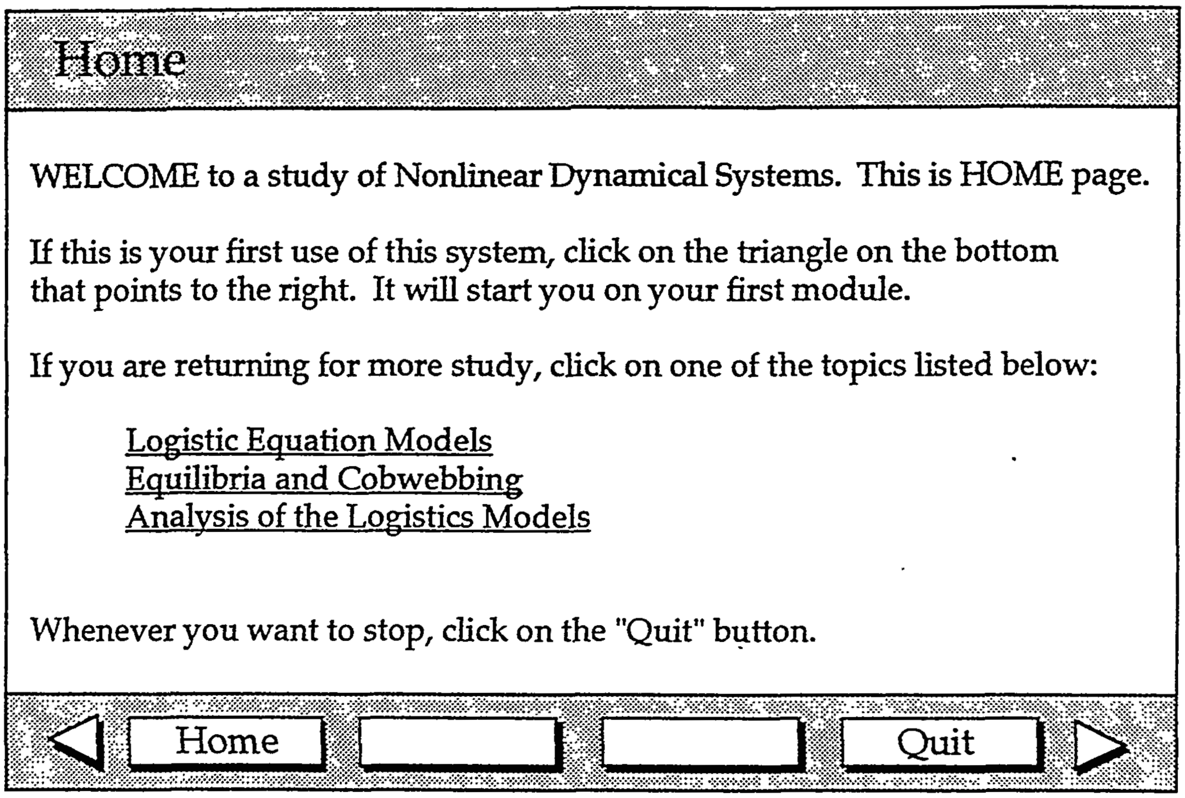
The presentation is given on a series of cards. The general layout of a typical card is shown above. At the top is a statement of the Topic under discussion. The discussion material for the card will be given in the blank space. This material may consist of sentences, diagrams, equations, pictures, or combinations of them all. For this report, pictures are excluded because the technology is not available.

The triangles and rectangles in the bottom strip furnish the hypertext capabilities of the presentation. The triangular buttons represent the ability to move to the immediately-preceding or immediately-following card. The button labeled "Home" represents the ability for the student to transfer to the beginning of the entire presentation. The button labeled "Quit" represents the ability for the student to terminate the presentation. The unlabeled buttons represent optional card locations that are dependent upon the context of a given card.

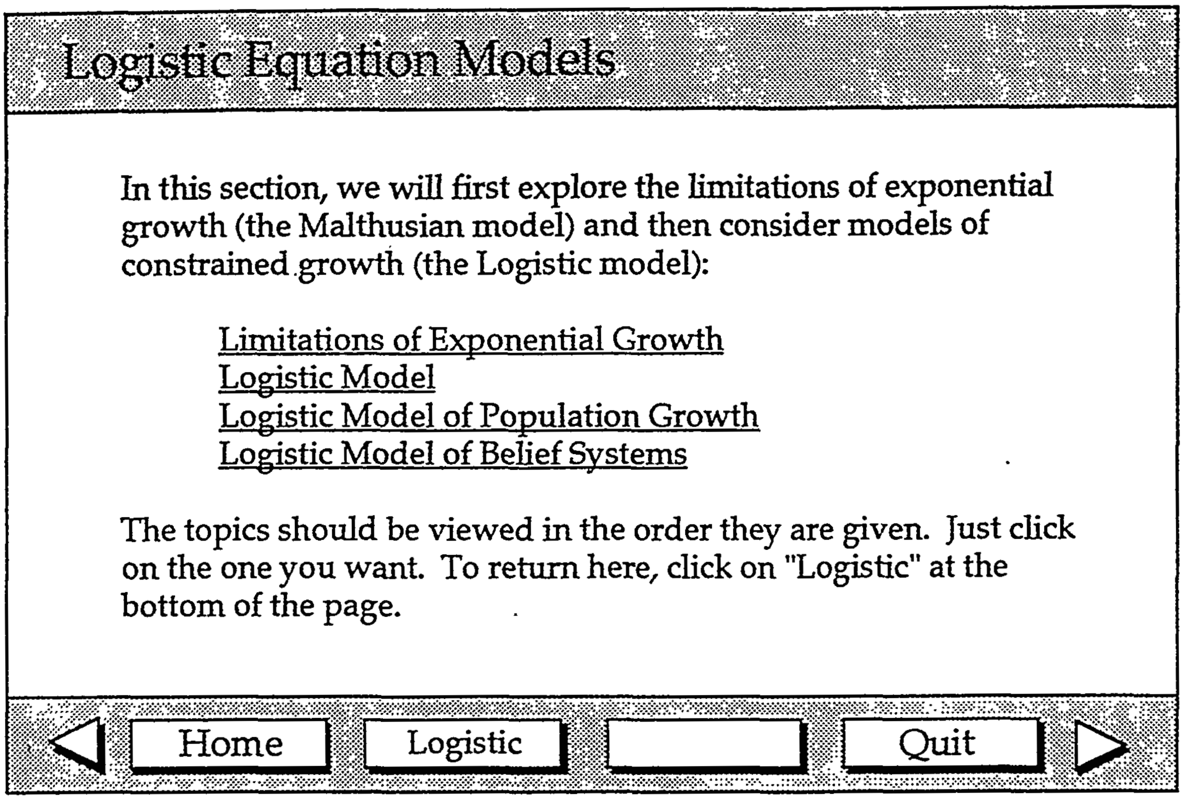
In addition, words in the discussion material which are underlined have the same function as buttons in the bottom strip. Those words allow students to transfer to other cards while in the middle of the presentation to get "refresher" information relevant to the concepts being presented. After reading the "refresher" information, the student can return to his original location in the presentation.



We begin our discussion with a graphic presentation of the essential steps in dynamical modeling.



Although the "Home" page is not exactly the first page in our presentation, it serves as a traffic director. When the student begins, he will pass through this page and begin the first module — Logistic Equation Models. On subsequent uses of the system, the student can jump immediately to the last module he was studying. At any time, the student can visit this page by clicking on the "Home" button at the bottom of the screen and be re-directed to any part of the presentation.



This is the "home" page for the section developing logistic models. The student is first shown the need for such models by considering limitations of the predictive power of models based solely on exponential growth. Although logistic models are usually associated with population models, I have also included discussion of models in which logistic modeling was used to show how rumors or "beliefs" spread within a population.

Limitations of Exponential Growth

Let A(n) be the population size at time period n. Assume that the number of births in a given time period is proportional to the size of the population in that period, and the proportionality factor is b — the birth rate.

$$\text{births in period } n = bA(n)$$

Further, assume in a like manner that the number of deaths in a given time period is proportional to the size of the population in that period, and the proportionality factor is d — the death rate.

$$\text{deaths in period } n = dA(n)$$

◁ Home | Logistic | | Quit ▷

Limitations of Exponential Growth

Then,

$$A(n+1) = A(n) + bA(n) - dA(n)$$
$$A(n+1) = (1 + b - d)\,A(n)$$

$$A(n+1) = (1 + r)\,A(n),$$
$$\text{where } r = b - d = \text{net growth rate}$$

From the material in Linear Dynamical Systems, we know that the solution to this system is

$$A(k) = (1 + r)^k A(0)$$

This is the Malthusian model of exponential growth, from which he predicted a world-wide catastrophe.

◁ Home | Logistic | | Quit ▷

Limitations of Exponential Growth

Then,

$$A(n+1) = A(n) + bA(n) - dA(n)$$
$$A(n+1) = (1 + b - d)\,A(n)$$

$$A(n+1) = (1 + r)\,A(n),$$
$$\text{where } r = b - d = \text{net growth rate}$$

From the material in Linear Dynamical Systems, we know that the solution to this system is

$$A(k) = (1 + r)^k A(0)$$

This is the Malthusian model of exponential growth, from which he predicted a world-wide catastrophe.

Home | Logistic | | Quit

Limitations of Exponential Growth

An analysis of the data for the beginning years of the country suggests an annual growth rate of about 3% ($r = 0.03$). Therefore, if we calculate

$$A(n + 1) = (1.03)^k A(0), \text{ where } A(0) = 4:$$

t	N	Pred	t	N	Pred	t	N	Pred
0	4		70	31	32	140	123	251
10	5	5	80	39	43	150	132	337
20	7	7	90	50	57	160	151	453
30	10	10	100	63	77	170	179	609
40	13	13	110	76	103	180	203	818
50	17	18	120	92	139	190	226	1099
60	23	24	130	106	187			

We see that the predicted values (Pred) are initially accurate but eventually grossly inflated.

Home | Logistic | | Quit

Limitations of Exponential Growth

The predicted values for the late 1800's were 10% or more in error, and the 1930 prediction is double the actual value.

Clearly, the exponential growth model, which seemed so promising for the first 100 years or so, is not a model adequate to predict the population of the United States.

Furthermore, data from animal populations also suggest that the model is satisfactory only as long as the population is not too large.

Home | Logistic | | Quit

Limitations of Exponential Growth

Malthus made his prediction of a population catastrophe in 1798:

"Population when unchecked increases in a geometrical ratio. Subsistence increases only in an arithmetic ratio. A slight acquaintance with numbers will show the immensity of the first power in comparison of the second."

In fact, Malthus had the ingredients of a correct interpretation. He recognized that there were limits to the growth of a population. He did not recognize, however, that those limits could have a moderating influence on the population long before catastrophe is on the horizon.

That insight is the germ of the Logistic Equation and a Dutch biologist named Verhulst (1837).

Home | Logistic | | Quit

Logistic Model

Our aim is to change the exponential model, which had a fixed growth rate, to one in which the growth rate is a function of the population size for a given time period.

Further, we saw that the problem with the exponential model was that the predictions were too large. So clearly, the growth rate must DECREASE with increasing population size.

Thus, something affects big populations that doesn't affect small ones and that something is competition! The population numbers are confronting the stark reality of finite resources that are insufficient for everyone to survive.

Home | Logistic | | Quit

Logistic Model

Thus, our new model, called the Logistic Model, starts with the assumption that the environment of the population can only support a certain number, say L, of the species.

That number L is called the CARRYING CAPACITY of the environment.

Home | Logistic | | Quit

Logistic Model

The (restrained) growth rate should relate to the carrying capacity as follows:

When the population is MUCH LESS than the carrying capacity, there is plenty of food for the population and the growth rate should be close to the unrestricted growth rate, r.

When the population is LESS than the carrying capacity, there is sufficient food for the population and the growth rate should be positive but not as large as the unrestricted growth rate, r.

When the population EXCEEDS the carrying capacity, there is not enough food and the growth rate should be negative.

Satisfy yourself that these conditions are met by:

restrained growth rate = r[1 - (A(n)/L)]

◁ Home | Logistic | | Quit ▷

Logistic Model

Our model now becomes:

$$A(n+1) = A(n) + r[1 - (A(n)/L)]A(n)$$

which simplifies to:

$$A(n+1) = (1+r)\,A(n) - b\,A^2(n),$$

where $b = r/L$.

This is the Logistic Model and the "$- b\,A^2(n)$" term is called a damping term because its effect is to dampen or suppress the growth of the population.

◁ Home | Logistic | | Quit ▷

Logistic Model of Population Growth

In 1920 R. Pearl and L. J. Reed derived a logistic model and applied it to the United States using a carrying capacity of 200,000,000:

t	N	Pred	t	N	Pred	t	N	Pred
0	4		70	31	29	140	123	125
10	5	5	80	39	37	150	132	141
20	7	7	90	50	48	160	151	155
30	10	10	100	63	60	170	179	167
40	13	13	110	76	74	180	203	177
50	17	17	120	92	91	190	226	184
60	23	22	130	106	108			

As you see, the predictions are excellent until we get near the carrying capacity. The Achilles heel of the Logistic Model is knowing the carrying capacity.

Home | Logistic | | Quit

Logistic Model of Population Growth

In *Differential Equations and Their Applications,* Martin Braun makes the following observation as a consequence of a logistic model application:

"In 1845 Verhulst prophesied a maximum population for Belgium of 6,600,000. Now, the population of Belgium in 1930 was already 8,092,000. This large discrepancy would seem to indicate that the logistic law of population growth is very inaccurate, at least as far as the population of Belgium is concerned.

However, this discrepancy can be explained by the astonishing rise of industry in Belgium, and BY THE ACQUISITION OF THE CONGO WHICH SECURED FOR THE COUNTRY SUFFICIENT ADDITIONAL WEALTH TO SUPPORT THE EXTRA POPULATION."

Mathematical models can yield unexpected insights.

Home | Logistic | | Quit

Logistic Model of Belief Systems

The following model is due to J. Sandefur as published in *Discrete Dynamical Systems: Theory and Applications:*

In the popular *Star Trek* series, the Federation's Prime Directive was that the *Enterprise* crew could not interfere in any world that had not had contact with other worlds. The fear was that any contact from another world could alter the course of history on a developing planet.

Suppose, then, that Earth is actually being watched by intelligent creatures from another planet. These aliens wish to study us without having us know they are there or at least they want most of us not to believe they exist.

◁ Home | Logistic | | Quit ▷

Logistic Model of Belief Systems

The aliens might reason as follows:

Let A(n) be the fraction of people on Earth that believes in flying saucers at time n. Then A(n) is some number between 0 and 1.

Thus, 1 - A(n) is the fraction of people on Earth who, at time n, do not believe in flying saucers.

The aliens assume that in each time period the believers convince a certain proportion of the nonbelievers that flying saucers do exist. That proportion depends on the interaction of believers and nonbelievers, which can be modeled by the product A(n)[1 - A(n)].

When a population is broken into two parts, A(n) and 1 - A(n), their product is called the contact ratio and is used extensively in the study of epidemics.

◁ Home | Logistic | | Quit ▷

Logistic Model of Belief Systems

So, in the absence of flying saucers, A(n) would satisfy the dynamical system

$$A(n+1) = A(n) + kA(n)[1 - A(n)]$$

If $k > 0$, then there is a tendency to believe, while if $k < 0$, there is a tendency to not believe in flying saucers.

Let's assume the aliens have determined through their studies that $k = -0.01$.

Home | Logistic | | Quit

Logistic Model of Belief Systems

Let's also assume that flying saucers do land in certain areas and b per cent (as a fraction) of the people see them each time period.

Then we have (up to) 100b per cent of new believers and our dynamical system model becomes:

$$A(n+1) = 0.99A(n) + 0.01A^2(n) + b$$

Our aliens must ask themselves:

How large can b become without having everyone believe in flying saucers? In other words, they are willing to have some people believe since, if most people do not believe, Earth's behavior will not change.

Home | Logistic | | Quit

Logistic Model of Belief Systems

This is as far as we can currently pursue this model because we have no provisions for analysis. That is, what are the fixed points or equilibria of these models?

That subject is taken up in the next section,

Equilibria and Cobwebbing.

Home | Logistic | | Quit

Equilibria and Cobwebbing

In Theory of First Order Affine Systems, we said that any difference equation is said to be in EQUILIBRIUM if every term has the same value — $A(0) = A(1) = A(2) = ...$

We let that equilibrium value be a, we substituted a for all "A(?)" terms in the difference relation, and then solved for a.

Find the equilibrium value(s) for the logistic equation:

$$A(n+1) = (1+r)A(n) - bA^2(n)$$

Home | | | Quit

Equilibria and Cobwebbing

Given the logistic equation:

$$A(n+1) = (1+r)A(n) - bA^2(n)$$

and substituting a as the equilibrium value yields:

$$a = (1+r)\,a - ba^2$$
$$ba^2 - ra = 0$$
$$(r/L)a^2 - ra = 0$$
$$ra[(a/L) - 1] = 0$$

Solution: $a = 0$ or $a = L$

This says that the population becomes extinct or it "maxes out" at the carrying capacity limit. Which is it going to be?

Home | | | Quit

Equilibria and Cobwebbing

Although the analysis can be done algebraically, it is far more interesting to use the analysis technique known as

Cobwebbing

Home | | | Quit

Equilibria and Cobwebbing

Cobwebbing is a graphical procedure which allows you to see the progression of A(n)s from one period to the next.

What we will be seeing is that some values seem to attract successions of A(n)s and other values seem to repel these successions.

Those attracting points are called attracting equilibria, and the points which repel are called repelling equilibria. Note that repelling equilibria are still equilibria because if that any A(n) every assumes that exact value, all successive A(n) will also remain there. But the "hit" must be the exact value. Just the smallest deviation will lead to repulsion.

Home | | | Quit

Equilibria and Cobwebbing

Let's take the logistic dynamical system:

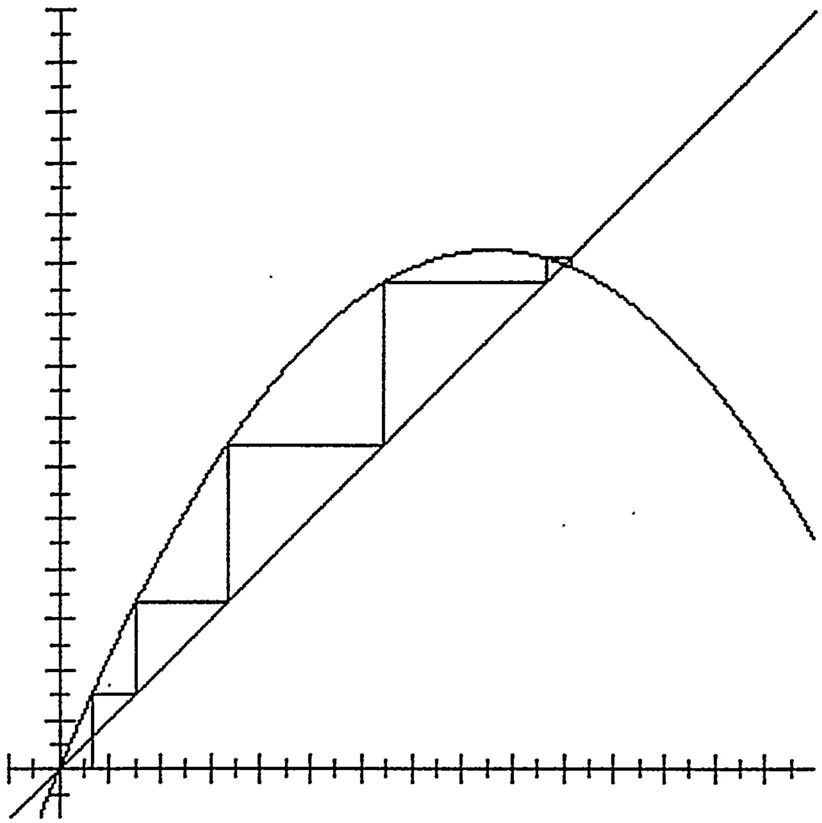
$$A(n+1) = (1+r)A(n) - bA^2(n)$$

and let r = 1.4 and L = 10.

$$A(n+1) = 2.4A(n) - 0.14A^2(n)$$

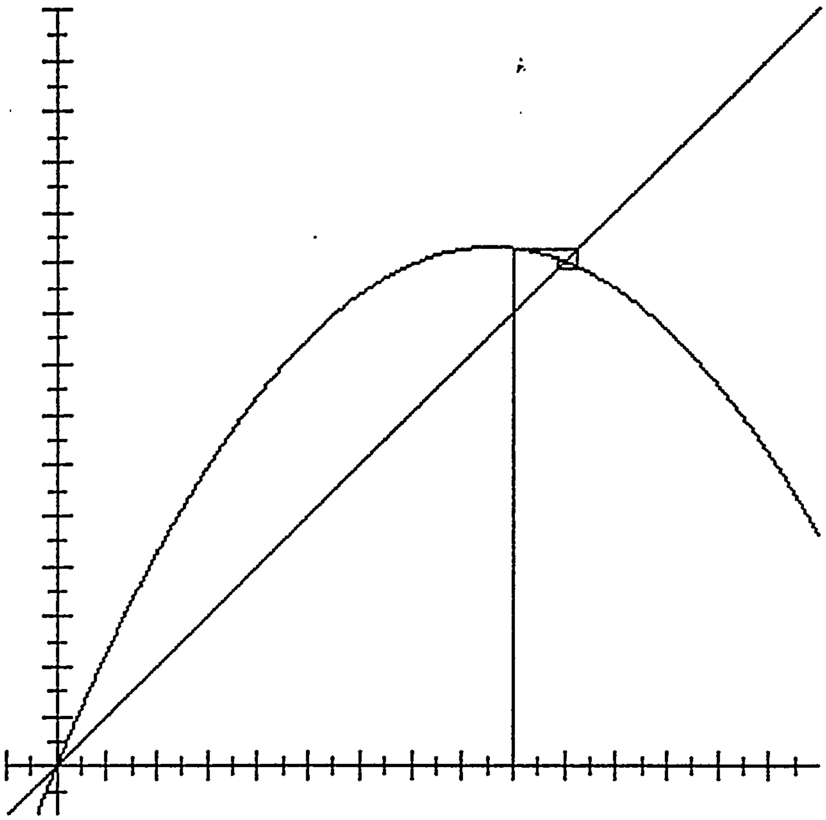
We know that the equilibria are 0 and 10. But what does the cobweb diagram show us?

Home | | | Quit

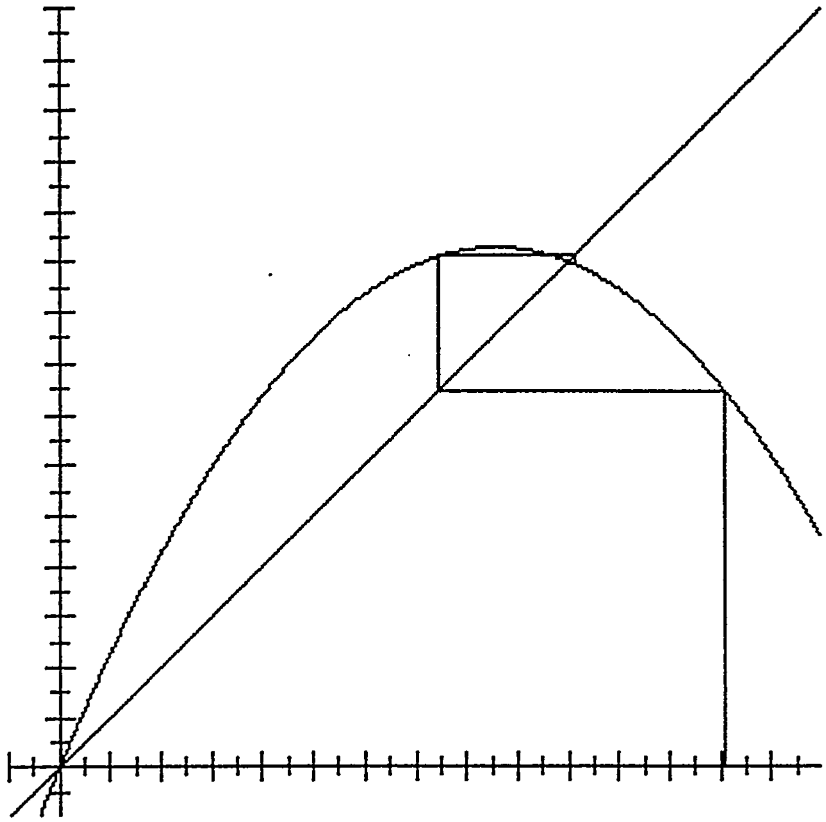


This cobweb diagram shows what happens when the initial value is close to a repelling fixed point, namely zero. The parabola is the characteristic equation (from which we calculated the value of a).

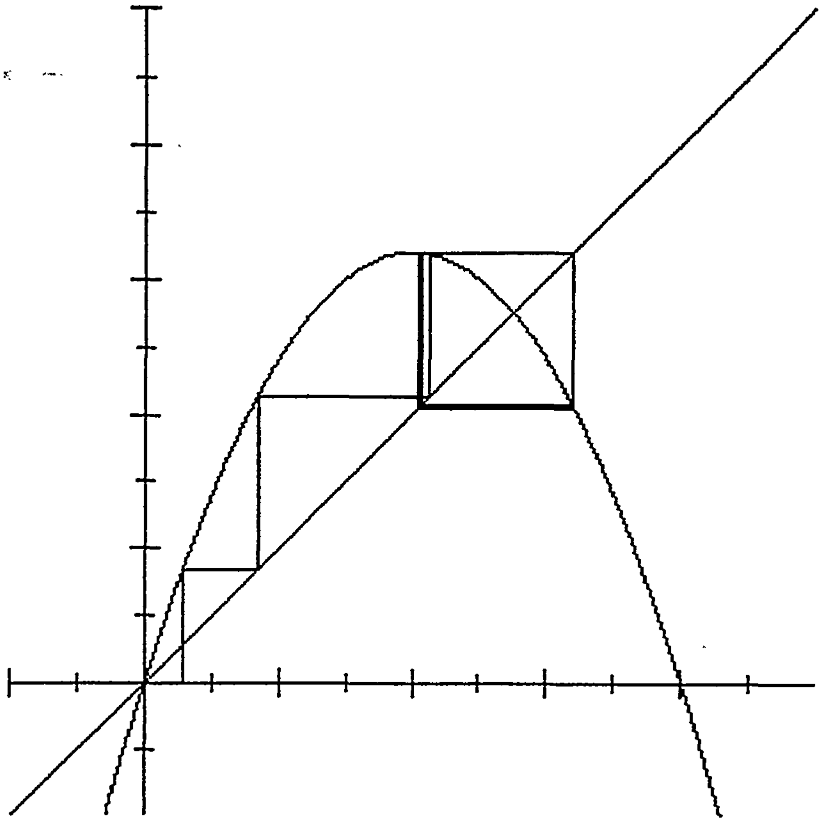
Take the first point, just to the right of the small tic mark. That is A(0). Its intersection with the parabola determines A(1). Now we need to get to the corresponding position for A(1) on the x-axis. The diagonal line (y = x) allows just that. Draw a horizontal line from the position on the parabola to the diagonal and you have the x-position for A(1). Then, repeat the process. Draw a vertical line to the parabola to get A(2), etc.



This cobweb diagram shows the succession of values when the initial value is close to the attracting fixed point, 10. By the way, 10 was also the attracting point when the initial value was close to the repelling fixed point, 0.



This cobweb diagram shows that even when the initial value is past the attracting fixed point — in this case, 10 — the dynamical system brings the values back to the attracting fixed point.

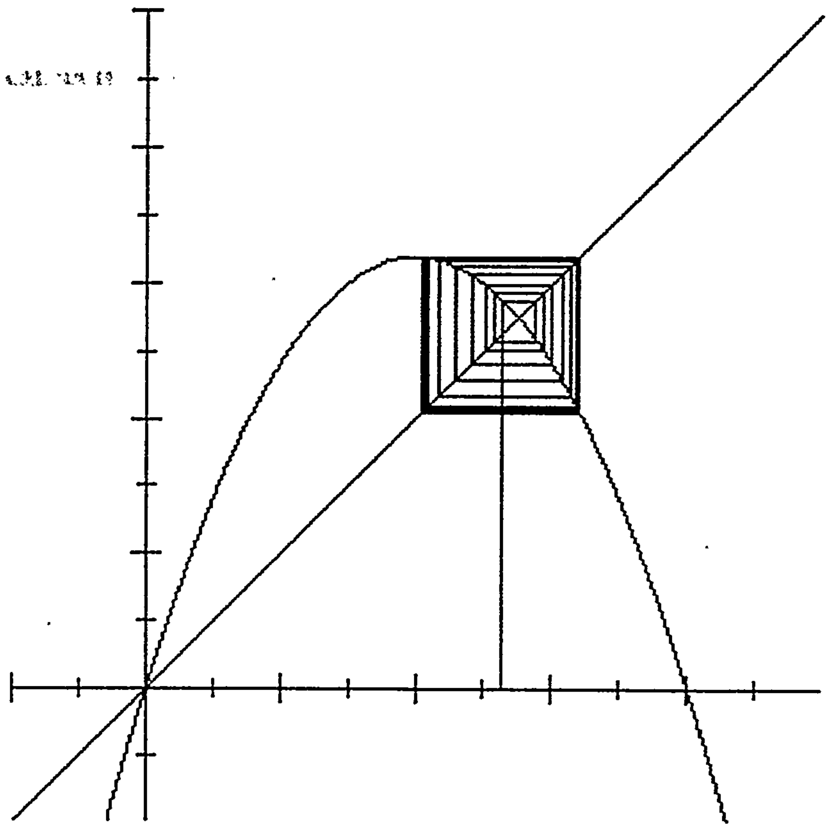


This cobweb diagram is for the dynamical system

$$A(n + 1) = 3.2A(n) - 0.8A^2(n)$$

The fixed points are 0 and 2.75. The diagram shows clearly that zero is a repelling fixed point. However, to our surprise, we see that the system does not home in on the fixed point of 2.75, but rather cycles about this fixed point. This is called a 2-cycle system.

Systems can have cycles with more than 2 elements. It is not clear whether such systems actually exist in real-life.



Again, this cobweb diagram shows that even when the initial value is close to the fixed point of 2.75, the system adopts a 2-cycle as its terminal behavior.

Analysis of the Logistics Models

In this section, we will analyze the two logistics models that were developed in an earlier section:

Analysis of the Model of Population Growth
Analysis of the Model of Belief Systems

The topics should be viewed in the order they are given. Just click on the one you want. To return here, click on "Analysis" at the bottom of the page.

Home | Analysis | | Quit

Analysis of the Model of Population Growth

We already know that the logistics model yields an S-shaped curve (called sigmoid) and the fixed points are zero and the carrying capacity.

Therefore, our analysis of the logistics model of population growth will focus on another aspect of population. We are concerned with the problem that faces all state agencies associated with wildlife — How much hunting should be allowed?

To simplify the results, we will choose units such that the carrying capacity equals one unit. For example, one unit could equal 10,000 deer. Further, let's assume that r, the unrestricted growth rate, is 0.8.

$$A(n+1) = A(n) + 0.8(1 - A(n))A(n) = 1.8A(n) - 0.8A^2(n)$$

Home | Analysis | | Quit

Analysis of the Model of Population Growth

Our first model is called the Fixed Harvest ("harvest" is what wildlife people call hunting). In this model, we allow hunters to kill b units of deer per season, where b is a fraction of a unit (which we're saying is 10,000 deer).

$$A(n+1) = 1.8A(n) - 0.8A^2(n) - b$$

Let b = 0.072 (720 deer are killed each year if one unit equals 10,000 deer).

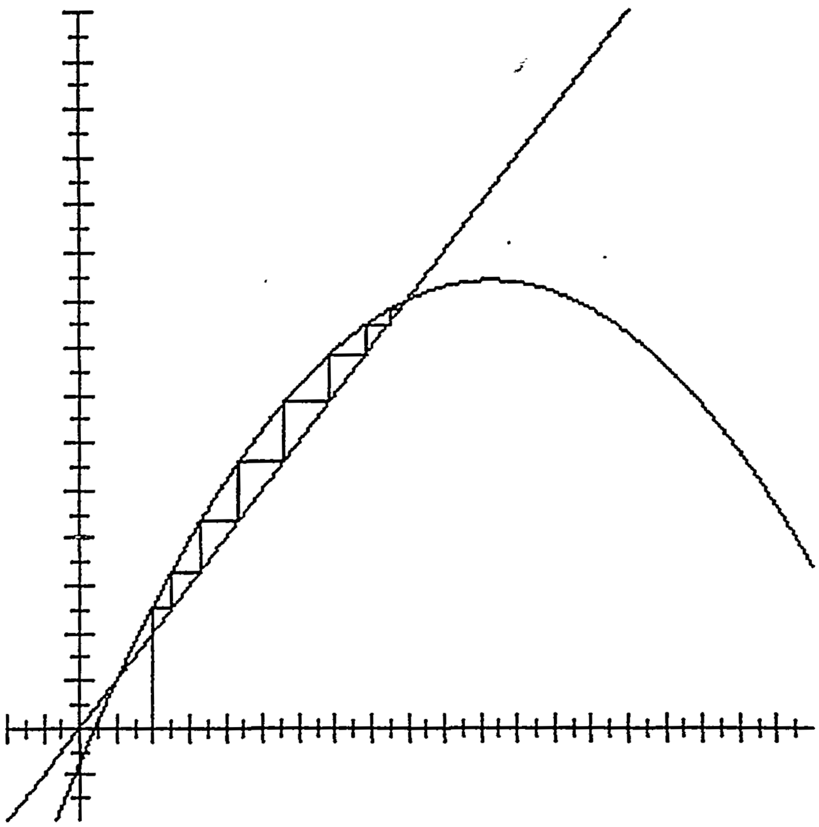
The dynamical system is:

$$A(n+1) = 1.8A(n) - 0.8A^2(n) - 0.072$$

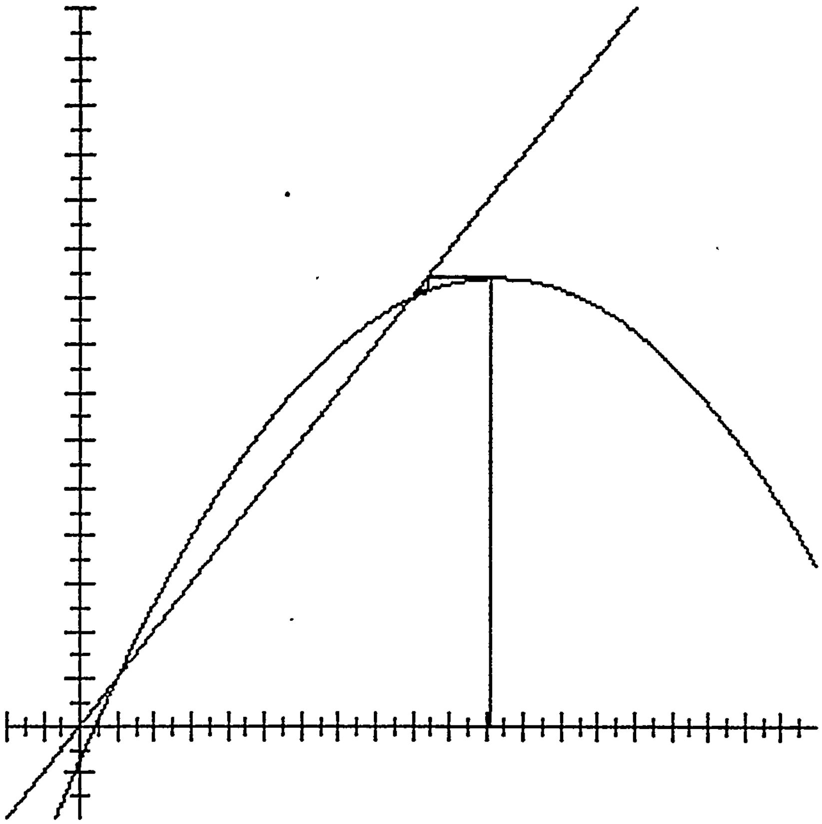
and the fixed points are 0.1 (1,000 deer) and 0.9 (9,000 deer).

Let's look at the cobweb.

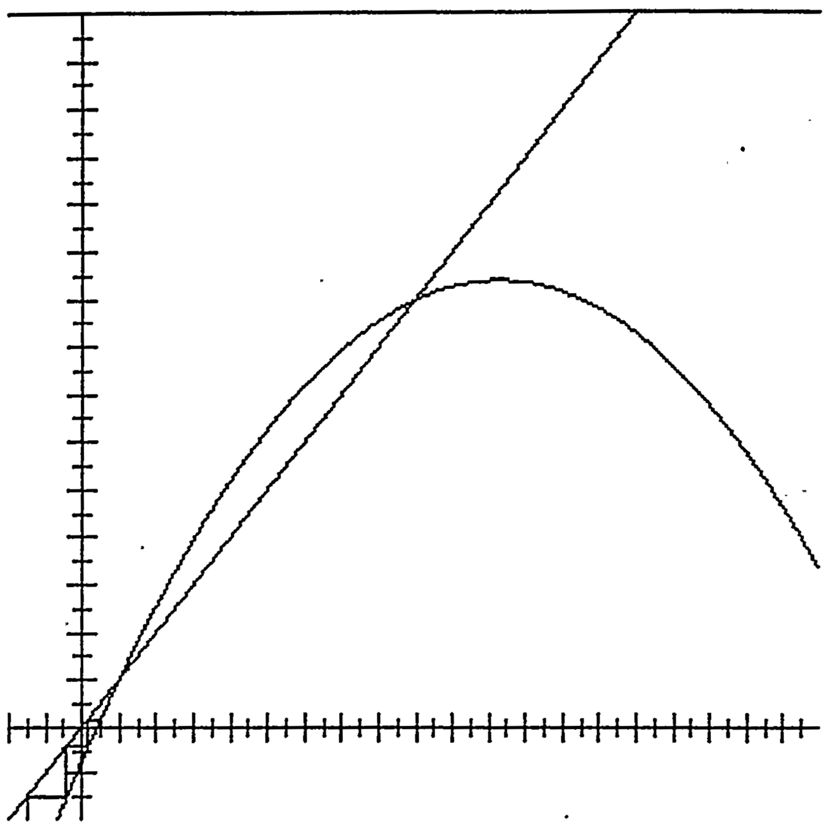
Home | Analysis | | Quit



Remember that the fixed points are 1000 and 9000 deer. Even if the population is close to, but above, 1000 deer and 720 deer are killed per year, the population will survive and eventually reach the attracting fixed point of 9000 deer.



If by some chance, the population of deer exceeds the carrying capacity of the environment, the harvest helps reduce the population to the attracting fixed point of 9000 deer.



However, if the population falls below the repelling fixed point of 1000, the harvest will serve to hasten the extinction of the population.

Analysis of the Model of Population Growth

Now, let's show what happens when the kill (oops, sorry), when the harvest is too large.

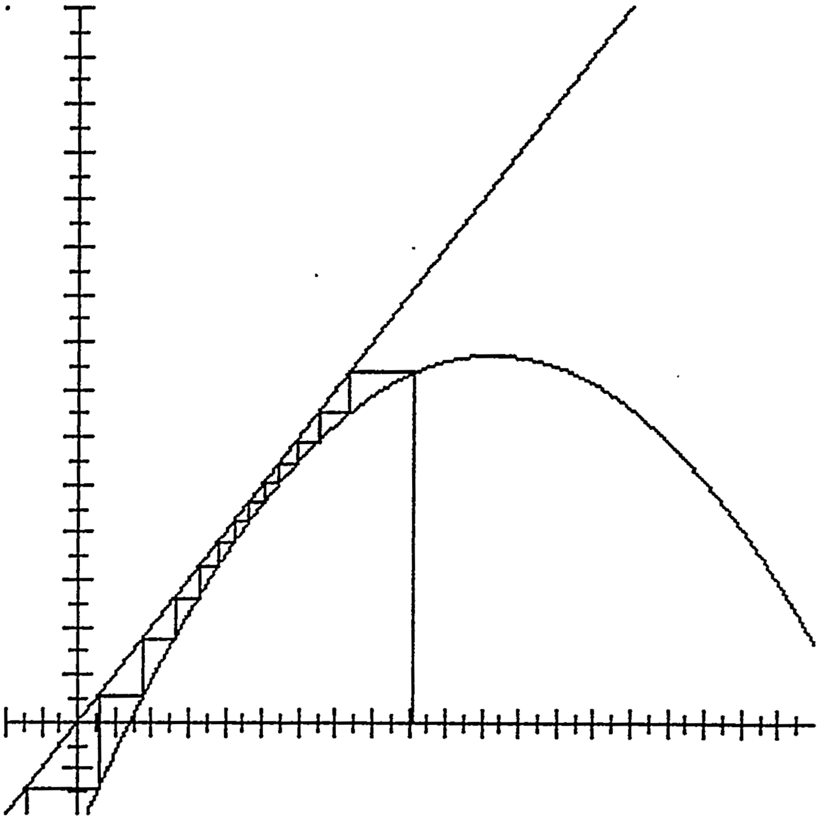
Let b = 0.24 (2400 deer are killed each year).

The dynamical system is:

$$A(n+1) = 1.8A(n) - 0.8A^2(n) - 0.24$$

Here's the cobweb.

◁ Home | Analysis | | Quit ▷



Regardless of the size of the population, a harvest of 2400 deer is more than the population can survive given its parameters for reproduction.

Analysis of the Model of Population Growth

An alternative strategy is to havest a fixed proportion of the population. Let b now represent the proportion of the population to be removed. Then the total number of deer killed will be bA(n)

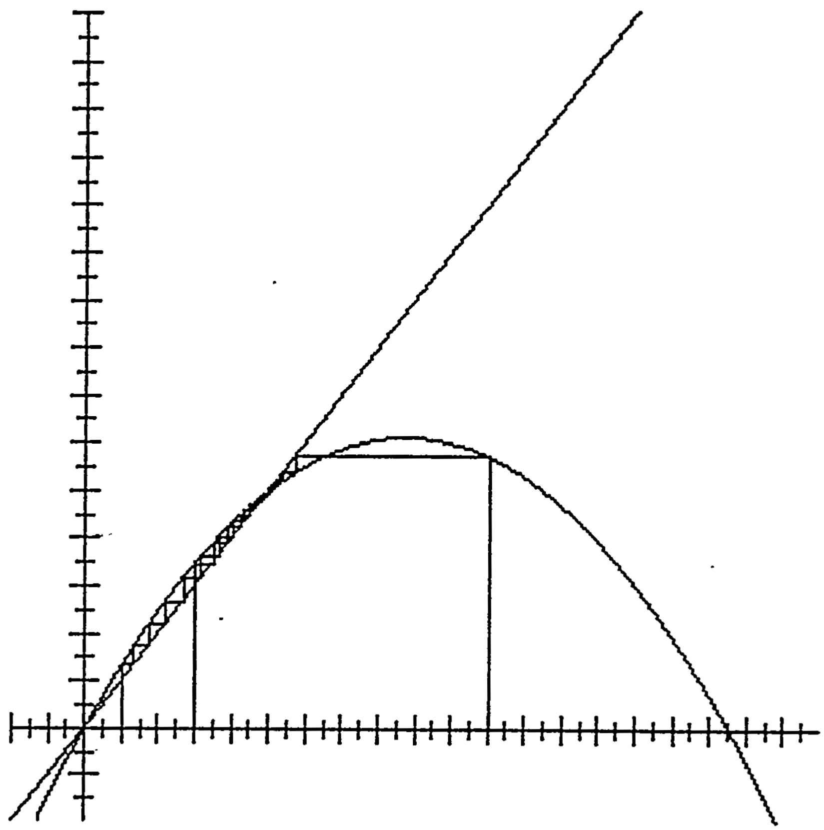
$$A(n+1) = 1.8A(n) - 0.8A^2(n) - bA(n)$$

or

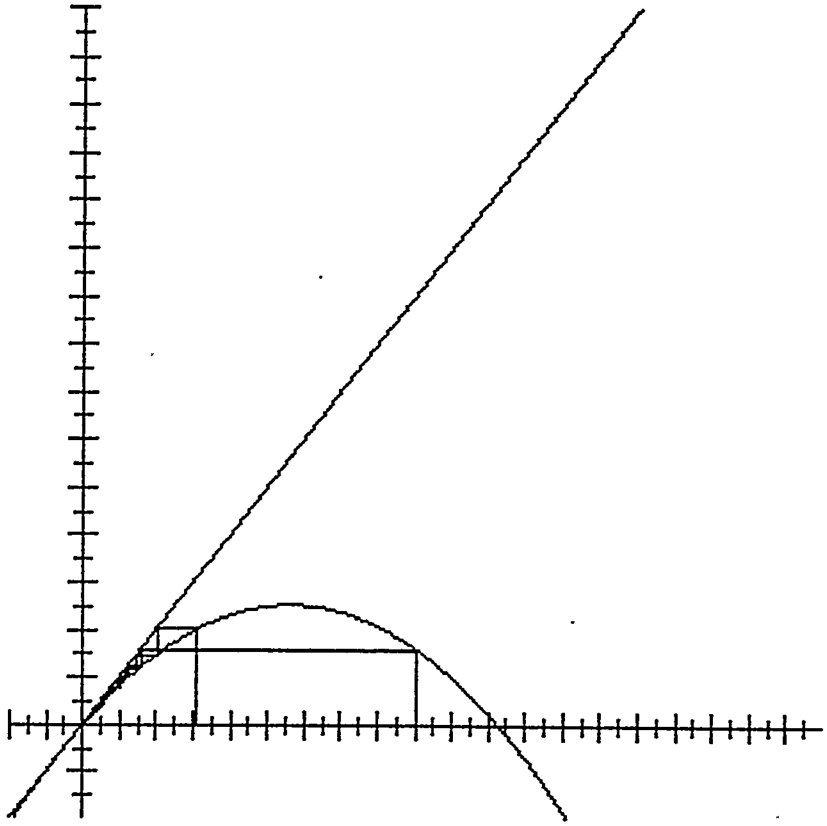
$$A(n+1) = (1.8 - b)A(n) - 0.8A^2(n)$$

We will take representative values for b and look at the cobweb.

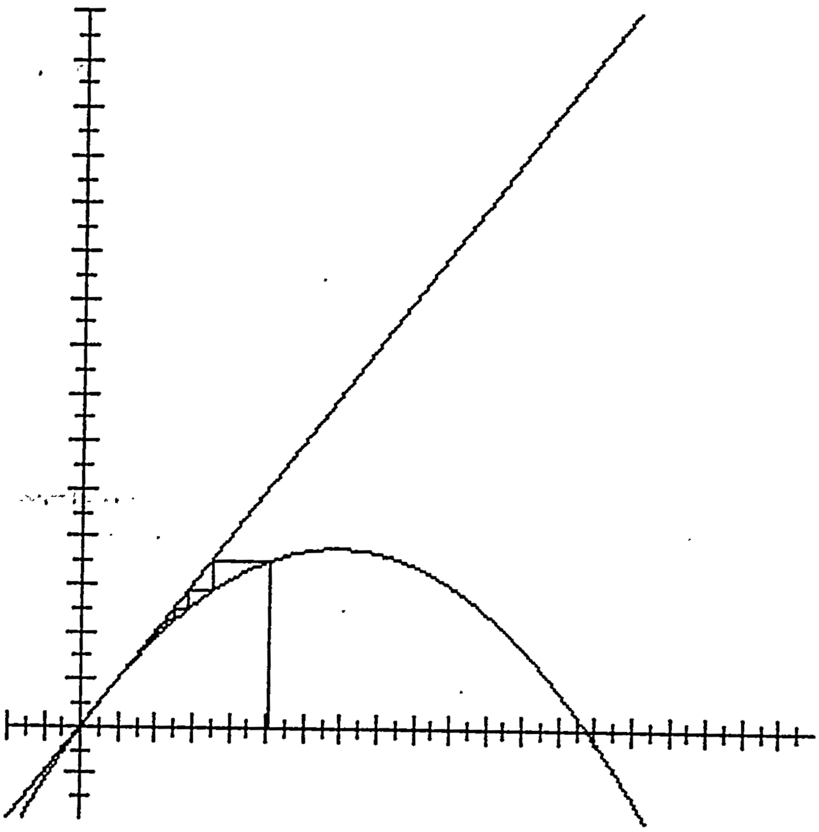
Home | Analysis | | Quit



This cobweb is for a value of b = 0.4, which shows a stable population.



This cobweb is for b = 0.9. It shows that the population will be driven to extinction.



This cobweb is for b = 0.7 and shows a sustainable population. It turns out that b = 0.8 is the critical point between sustainability and extinction.

Analysis of the Model of Belief Systems

This was our model:
Let's also assume that flying saucers do land in certain areas and b per cent (as a fraction) of the people see them each time period. Then we have (up to) 100b per cent of new believers and our dynamical system model becomes:

$$A(n+1) = 0.99A(n) + 0.01A^2(n) + b$$

Our aliens must ask themselves:

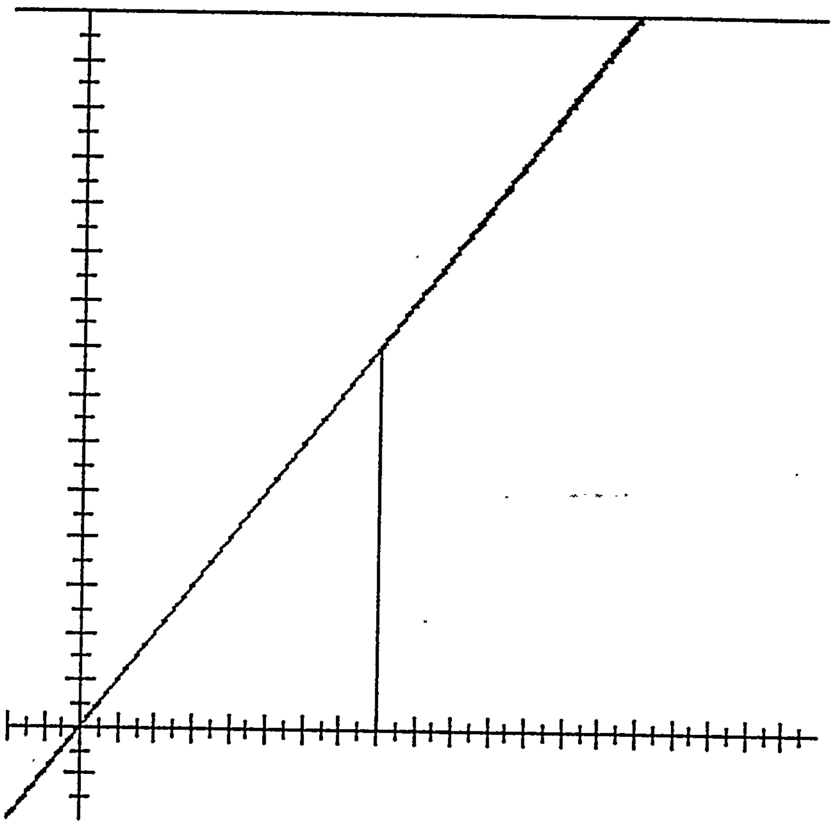
How large can b become without having everyone believe in flying saucers? In other words, they are willing to have some people believe since, if most people do not believe, Earth's behavior will not change.

Home | Analysis | | Quit

Analysis of the Model of Belief Systems

As before, let's take some representative values for b and look at the cobweb.

Home | Analysis | | Quit



Well, this shows the limitation of cobwebbing for analysis. The parameter used was b = 0.005. For that situation, eventually all would believe in flying saucers. The critical value for b turns out to be 0.0025.

TRAFFIC FLOW

Students : Altheia A.Griffin
Kenneth Boders
Jason McCaskill
Cecily Monroe
Andrew Smith

School: Florida A&M University
Development REsearch School

Mentor : Dr. James C. Turner Jr.

Teacher: Mrs. Nikki Turner

Undergraduate Assistants:
Ms. Clarissa Zellars
Ms. Mannietea Curry

Traffic Flow

The traffic flow that we considered is that of the cars moving on one side of a divided highway. Since the behavior of a vehicle is influenced by the cars around it, we can predict the movement of the vehicle through CA. A car can move forward one space, sideways, or it can be at a stand still. In the figure 4 the number corresponds to the site. For instance, the initial site is one, the site ahead is site two, the site adjacent is site four and etc.

Figure 4

[6]	[4]	[5]	->
[3]	[1]	[2]	->

Each site is represented by a one (1) when occupied by a car and zero (0) if it is empty. To determine the movement of the cars we define the Moore neighborhood for our boundary conditions and invoke CA with the following rules.

<u>*Rules*</u>

Rule #1

An occupied site [1] becomes empty when the site ahead of the site [2] is empty.

driveWithObstacle[1, 0, _, _, _, _] = 0

Rule # 2

An occupied site [1] that is blocked by either a car or an obstacle in the site ahead of it, switches lanes when there is an adjacent empty space at site [4] and behind the adjacent site [6] is empty or has an obstacle.

driveWithObstacle[1, 1 | c, _, 0, _.) | c] = 0

Rule # 3

An occupied site [1] remains occupied when the sites ahead of the site [2] and the adjacent site [4] are both occupied or contain obstacles.

driveWithObstacle[1, 1 | c, _, 1 | c, _, _] = 1

Rule # 4

An empty site [1] that is followed by a car becomes occupied by that car.

driveWithObstacle[0, _, 1, _, _, _] = 1

Rule # 5

An empty site [1] that is followed by an empty space or obstacle in site [3] and is next to a car that is blocked by a car or obstacle ahead of it becomes occupied as the car changes lanes.

```
driveWithObstacle[0, _, 0 | c, 1, 1 | c, _] = 1
```

Rule # 6

In situations other than those given above, empty sites remain empty and occupied sites and obstacles remain in place.

```
driveWithObstacle[x_, _, _, _, _, _] := x
```

Now to start the keepOnMoveingOn code in the following way to show the path of the traffic when the obstacles occur at random positions.

```
In [5] := indy500[n_, p_, t_] :=
        Module[{roadWithObstacle, driveWithObstacle},

                roadWithObstacle =
            ReplacePart[Table[Floor[p + Random[]],{2}, {n}], c,

            {Random[Integer,{1, 2}], Random[Integer,{1, n}]}] ;
                    driveWithObstacle[1, 0, _, _, _, _] = 0;
                    driveWithObstacle[1, 1 | c, _, 0, _, | c] = 0;
                    driveWithObstacle[1, 1 | c, _, 1 | c, _, _] = 1;
                    driveWithObstacle[0, _, 1, _, _, _] = 1;
                    driveWithObstacle[0, _, 0 | c, 1, 1 | c, _] = 1;
                    driveWithObstacle[x_, _, _, _, _, _] = x;

        NestList[MapThread[driveWithObstacle, {#,
        RotateRight[#, {0, -1}]
    RotateRight[#, {0,- 1}], RotateRight[#, {1, 1}]},
                2]&, roadWithObstacle, t]]
```

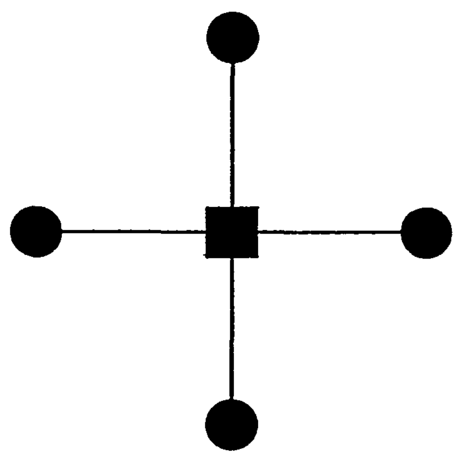
Test the indy500 function by assigning numbers to the three variables. We must not forget to assign the obstacle "c" a color value.

```
test = indy500[30, ,.7, 34];
     Map[Show[Graphics[RasterArray[# /.
                {0 -> RGBColor[0.7, 0.7, 0.7],
                1 -> RGBColor[0, 1, 0],
                c -> RGBColor[0, 0, 1]}]],
          AspectRatio -> Automatic]&,

          test];
```

Definitions

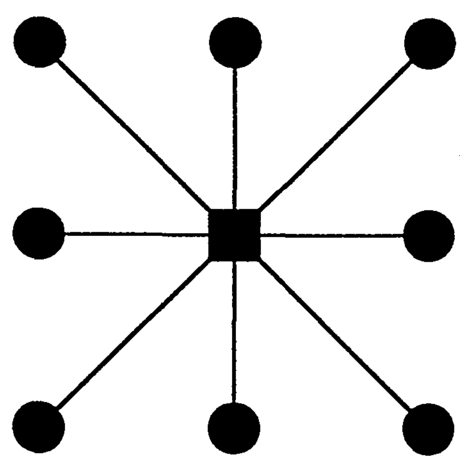
von Neumann neighborhood - consist of the site and the four nearest neighbors, north (above) east (right), south (below), and west (left) of the site is represented in the diagram below.

von Neumann neighborhood



Moore (neighborhood) - consists of the site and the eight nearest neighbor sites, north, northeast, east, southeast, south, southwest, west, and northwest.

Moore neighborhood



cellular automaton - consists of a system sites having various initial values. The sites evolve in time steps as each site assumes a new value based on the values of some local neighborhood of sites. To incorporate a cellular automata models the lattices needs to be well defined for the neighborhoods of sites for various boundary conditions.

lattice - a matrix consisting of n rows and m columns.

maze - a confusing network of passages. The maze's code is defined by rectangular lattice consisting of sites with value 1 (a wall site), 0 (a path site). All turns in the paths and the walls are at ninety degrees.

mathematica - a high lever programming language

Code for the Traffic Simulation

The last two sections of code modeled the traffic going in the same direction on a two lane highway. This was most interesting because we were able to animate our results. The traffic followed the six drive rules outlined earlier. We were able to observe the traffic in normal conditions and when an obstacle was introduced. We could also observe and graph the average velocity when we increased the probability of cars on the road.

```
(roadTest= {{a, 6, 4, 5, b},
        {c, 3, 1, 2, d}})//TableForm

MapThread[rule, {#, RotateRight[#, {0, -1}],
                        RotateRight[#, {0, 1}],
                        RotateRight[#, {1, 0}],
                        RotateRight[#, {1, -1}],
                        RotateRight[#, {1, 1}]},
                             2]&[roadTest][[2,3]]

keepOnMovingOn[n_, p_, t_] :=
        Module[{road, drive},

                road = Table[Floor[p + Random[]],{2},{n}];
                drive[1, 0, _, _, _, _] = 0;
                drive[1, 1, _, 0, _, 0] = 0;
                drive[1, 1, _, 1, _, _] = 1;
                drive[0, _, 1, _, _, _] = 1;
                drive[0, _, 0, 1, 1, _] = 1;
                drive[x_, _, _, _, _, _] := x;

                NestList[MapThread[drive,
                        {#, RotateRight[#, {0, -1}],
                                RotateRight[#, {0, 1}],
                                RotateRight[#, {1, 0}],
                                RotateRight[#, {1, -1}],
                                RotateRight[#, {1, 1}]},
                                2]&, road, t]]

Map[Show[Graphics[RasterArray[# /.
        {0 -> RGBColor[0.7, 0.7, 0.7],
                1 -> RGBColor[0, 1, 0]}]],
                AspectRatio -> Automatic]&,
                keepOnMovingOn[10, .5, 10]];
```

```
(roadTest= {{a, 6, 4, 5, b},
        {c, 3, 1, 2, d}})//TableForm

MapThread[rule, {#, RotateRight[#, {0, -1}],
                          RotateRight[#, {0, 1}],
                          RotateRight[#, {1, 0}],
                          RotateRight[#, {1, -1}],
                          RotateRight[#, {1, 1}]},
                                2]&[roadTest][[2,3]]

keepOnMoveing On[n_, p_, t_] :=
        Module[{road, drive},

                road = Table[Floor[p + Random[]],{2},{n}];
                drive[1, 0, _, _, _, _] = 0;
                drive[1, 1, _, 0, _, 0] = 0;
                drive[1, 1, _, 1, _, _] = 1;
                drive[0, _, 1, _, _, _] = 1;
                drive[0, _, 0, 1, 1, _] = 1;
                drive[x_, _, _, _, _, _] := x;


                NestList[MapThread[drive,
                        {#, RotateRight[#, {0, -1}],
                                        RotateRight[#, {0, 1}],
                                        RotateRight[#, {1, 0}],
                                        RotateRight[#, {1, -1}],
                                        RotateRight[#, {1, 1}]},
                                        2]&, road, t]]

Map[Show[Graphics[RasterArray[# /.
          {0 -> RGBColor[0, 1, 1],
                1 -> RGBColor[0, 0, 1],
                c -> RGBColor[1, 0, 1]}]],
                AspectRatio -> Automatic]&,
                test];

test = indy500[30, .6, 10];

indy500[n_, p_, t_] :=
        Module[{roadWithObstacle, driveWithObstacle},
        roadWithObstacle = ReplacePart[Table[Floor[p +
        Random[]],{2}, {n}], c, {Random[Integer,{1, 2}],
        Random[Integer,{1, n}]}] ;
                driveWithObstacle[1, 0, _, _, _, _] = 0;
                driveWithObstacle[1, 1 | c, _, 0, _, 0 | c] = 0;
                driveWithObstacle[1, 1 | c, _, 1 | c, _, _] = 1;
                driveWithObstacle[0, _, 1, _, _, _] = 1;
                driveWithObstacle[0, _, 0 | c, 1, 1 | c, _] = 1;
                driveWithObstacle[x_, _, _, _, _, _] = x;

        NestList[MapThread[driveWithObstacle, {#,
                RotateRight[#, {0, -1}],
                RotateRight[#, {0, 1}], RotateRight[#, {1, 0}],
                RotateRight[#, {1, -1}], RotateRight[#, {1, 1}]},

                                        2]&,roadWithObstacle,t]]
```

Code and Results

We used the following code to solve the maze we created to represent a section of a city with many cul-de-sac's and dead end roads. This code invoked cellular automation with the von Neumann neighborhood as the boundary conditions . This code can be very useful to traffic engineers when they need an alternate path to get around road work.

```
maze={{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
          {1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,1,1,1},
          {1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1},
          {1,1,1,1,1,0,0,0,0,0,0,0,0,0,1,1,1,1},
          {1,0,0,0,0,0,1,1,0,1,1,1,1,1,1,1,1,1},
          {1,1,1,1,1,0,1,1,0,1,0,0,0,1,0,0,0,1},
          {1,1,1,1,1,0,1,1,0,1,0,1,0,1,0,1,0,1},
          {1,1,1,1,1,0,1,1,0,0,0,1,0,0,0,1,0,1},
          {1,1,0,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1},
          {1,1,0,1,1,0,0,0,1,1,1,1,0,0,1,1,0,1},
          {e,0,0,0,0,0,1,0,0,1,1,1,0,1,1,1,0,1},
          {1,1,0,1,1,0,1,1,0,1,1,1,0,1,1,1,0,1},
          {1,1,0,1,1,1,1,1,1,1,1,1,0,1,1,1,0,1},
          {1,0,0,0,0,0,0,0,0,0,1,1,0,1,1,1,0,1},
          {1,0,1,1,1,0,1,1,1,1,1,1,0,1,0,0,0,e},
          {1,0,1,1,1,0,1,1,1,0,0,0,0,0,1,1,0,1},
          {1,0,1,1,1,0,1,1,0,0,1,1,1,0,1,1,1,1},
          {1,1,1,1,1,0,0,0,0,1,1,1,1,0,1,1,1,1},
          {1,1,1,1,1,1,1,1,1,0,1,1,1,1,0,1,1,1,1},
          {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}};

Show[Graphics[RasterArray[Reverse[maze]/. {e -> RGBColor[0,1,0],
     1 -> RGBColor[1,0,0],0 ->RGBColor[1,1,0]}]],
     AspectRatio -> Automatic];

     PathToEnlightenment[ maze_] :=
Module[ {mazeSolve, VonNeumannValues},

     mazeSolve[0, 1, 1, 1, 0] :=1;
```

```
        mazeSolve[0, 1, 1, 0, 1] :=1;
        mazeSolve[0, 1, 0, 1, 1] :=1;
        mazeSolve[0, 0, 1, 1, 1] :=1;
        mazeSolve[0, 1, 1, 1, 1] :=1;
        mazeSolve[x_, _, _, _, _] :=x;

VonNeumannValues[func_, lat_] :=
        MapThread[func, {#,
                RotateRight[#, {0, 1}],
                RotateRight[#, {1, 0}],
                RotateRight[#, {0, -1}],
                RotateRight[#, {-1, 0}]},
                        2]&[lat];

FixedPoint[VonNeumannValues[mazeSolve, #]&,
        maze]]

Show[GraphicsArray[Map[Show[
    Graphics[RasterArray[Reverse[#]/.
        {e -> RGBColor[0, 1, 0],
        1 -> RGBColor[1, 0, 0],
        0 -> RGBColor[1, 1, 0]}]],
        AspectRatio -> Automatic,
        DisplayFunction -> Identity]&,
    {#,PathToEnlightenment[#]}&[maze]]]];
```

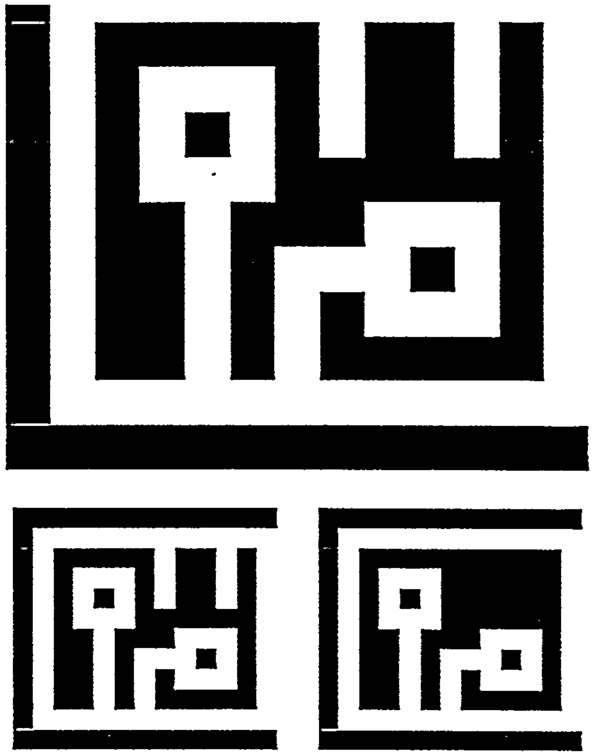
```
maze={{1,1,1,1,1,1,1,1,1,1,1,1,1},
      {e,0,0,0,0,0,0,0,0,0,0,0,0},
      {1,0,1,1,1,1,1,0,1,1,0,1,0},
      {1,0,1,0,0,0,1,0,1,1,0,1,0},
      {1,0,1,0,1,0,1,0,1,1,0,1,0},
      {1,0,1,0,0,0,1,1,1,1,1,1,0},
      {1,0,1,1,0,1,1,1,0,0,0,1,0},
      {1,0,1,1,0,1,0,0,0,1,0,1,0},
      {1,0,1,1,0,1,0,1,0,0,0,1,0},
      {1,0,1,1,0,1,0,1,1,1,1,1,0},
      {e,0,0,0,0,0,0,0,0,0,0,0,0},
      {1,1,1,1,1,1,1,1,1,1,1,1,1}};

Show[Graphics[RasterArray[Reverse[maze]/.
   {e -> RGBColor[0,1,0],
   1 -> RGBColor[1,0,0],0 ->RGBColor[1,1,0]}]],
   AspectRatio -> Automatic];
   PathToEnlightenment[ maze_] :=
Module[ {mazeSolve, VonNeumannValues},

   mazeSolve[0, 1, 1, 1, 0] :=1;
   mazeSolve[0, 1, 1, 0, 1] :=1;
   mazeSolve[0, 1, 0, 1, 1] :=1;
   mazeSolve[0, 0, 1, 1, 1] :=1;
   mazeSolve[0, 1, 1, 1, 1] :=1;
   mazeSolve[x_, _, _, _, _] :=x;

VonNeumannValues[func_, lat_] :=
   MapThread[func, {#,
      RotateRight[#, {0, 1}],
      RotateRight[#, {1, 0}],
      RotateRight[#, {0, -1}],
      RotateRight[#, {-1, 0}]},
         2]&[lat];

FixedPoint[VonNeumannValues[mazeSolve, #]&,maze]]

Show[GraphicsArray[Map[Show[
   Graphics[RasterArray[Reverse[#]/.
   {e -> RGBColor[0, 1, 0],
   1 -> RGBColor[1, 0, 0],
   0 -> RGBColor[1, 1, 0]}]],
   AspectRatio -> Automatic,
   DisplayFunction -> Identity]&,
{#,PathToEnlightenment[#]}&[maze]]]];
```

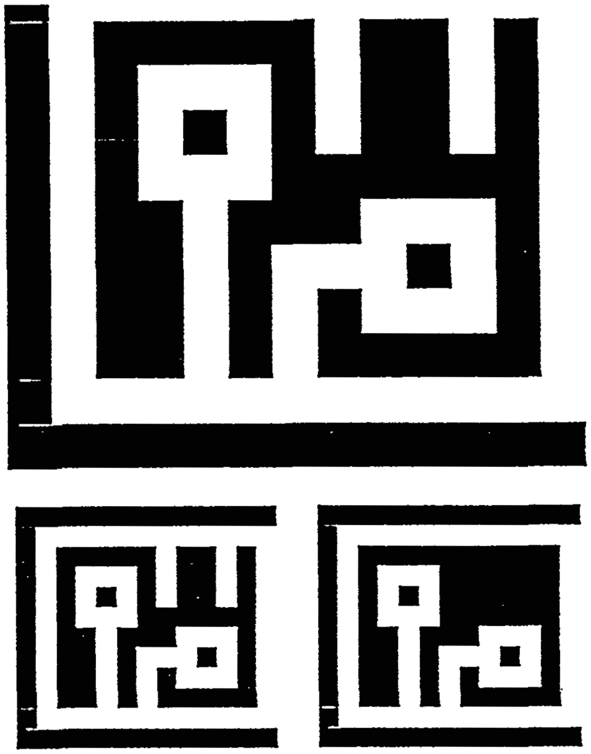
[redacted]

```
maze={{1,1,1,1,1,1,1,1,1,1,1,1,1},
      {e,0,0,0,0,0,0,0,0,0,0,0,0},
      {1,0,1,1,1,1,1,0,1,1,0,1,0},
      {1,0,1,0,0,0,1,0,1,1,0,1,0},
      {1,0,1,0,1,0,1,0,1,1,0,1,0},
      {1,0,1,0,0,0,1,1,1,1,1,1,0},
      {1,0,1,1,0,1,1,1,0,0,0,1,0},
      {1,0,1,1,0,1,0,0,0,1,0,1,0},
      {1,0,1,1,0,1,0,1,0,0,0,1,0},
      {1,0,1,1,0,1,0,1,1,1,1,1,0},
      {e,0,0,0,0,0,0,0,0,0,0,0,0},
      {1,1,1,1,1,1,1,1,1,1,1,1,1}};

Show[Graphics[RasterArray[Reverse[maze]/.
    {e -> RGBColor[0,1,0],
    1 -> RGBColor[1,0,0],0 ->RGBColor[1,1,0]}]],
    AspectRatio -> Automatic];
    PathToEnlightenment[ maze_] :=
Module[ {mazeSolve, VonNeumannValues},

    mazeSolve[0, 1, 1, 1, 0] :=1;
    mazeSolve[0, 1, 1, 0, 1] :=1;
    mazeSolve[0, 1, 0, 1, 1] :=1;
    mazeSolve[0, 0, 1, 1, 1] :=1;
    mazeSolve[0, 1, 1, 1, 1] :=1;
    mazeSolve[x_, _, _, _, _] :=x;

VonNeumannValues[func_, lat_] :=
    MapThread[func, {#,
        RotateRight[#, {0, 1}],
        RotateRight[#, {1, 0}],
        RotateRight[#, {0, -1}],
        RotateRight[#, {-1, 0}]},
            2]&[lat];

FixedPoint[VonNeumannValues[mazeSolve, #]&,maze]]

Show[GraphicsArray[Map[Show[
    Graphics[RasterArray[Reverse[#]/.
    {e -> RGBColor[0, 1, 0],
    1 -> RGBColor[1, 0, 0],
    0 -> RGBColor[1, 1, 0]}]],
    AspectRatio -> Automatic,
    DisplayFunction -> Identity]&,
{#,PathToEnlightenment[#]}&[maze]]]];
```

```
maze={{1,1,1,1,1,1,1,1,1,1,1,1,1},
      {e,0,0,0,0,0,0,0,0,0,0,0,0},
      {1,0,1,1,1,1,1,0,1,1,0,1,0},
      {1,0,1,0,0,0,1,0,1,1,0,1,0},
      {1,0,1,0,1,0,1,0,1,1,0,1,0},
      {1,0,1,0,0,0,1,1,1,1,1,1,0},
      {1,0,1,1,0,1,1,1,0,0,0,1,0},
      {1,0,1,1,0,1,0,0,0,1,0,1,0},
      {1,0,1,1,0,1,0,1,0,0,0,1,0},
      {1,0,1,1,0,1,0,1,1,1,1,1,0},
      {e,0,0,0,0,0,0,0,0,0,0,0,0},
      {1,1,1,1,1,1,1,1,1,1,1,1,1}};

Show[Graphics[RasterArray[Reverse[maze]/.
    {e -> RGBColor[0,1,0],
    1 -> RGBColor[1,0,0],0 ->RGBColor[1,1,0]}]],
    AspectRatio -> Automatic];
    PathToEnlightenment[ maze_] :=
Module[ {mazeSolve, VonNeumannValues},

    mazeSolve[0, 1, 1, 1, 0] :=1;
    mazeSolve[0, 1, 1, 0, 1] :=1;
    mazeSolve[0, 1, 0, 1, 1] :=1;
    mazeSolve[0, 0, 1, 1, 1] :=1;
    mazeSolve[0, 1, 1, 1, 1] :=1;
    mazeSolve[x_, _, _, _, _] :=x;

VonNeumannValues[func_, lat_] :=
    MapThread[func, {#,
        RotateRight[#, {0, 1}],
        RotateRight[#, {1, 0}],
        RotateRight[#, {0, -1}],
        RotateRight[#, {-1, 0}]},
            2]&[lat];

FixedPoint[VonNeumannValues[mazeSolve, #]&,maze]]

Show[GraphicsArray[Map[Show[
    Graphics[RasterArray[Reverse[#]/.
    {e -> RGBColor[0, 1, 0],
    1 -> RGBColor[1, 0, 0],
    0 -> RGBColor[1, 1, 0]}]],
    AspectRatio -> Automatic,
    DisplayFunction -> Identity]&,
{#,PathToEnlightenment[#]}&[maze]]]];
```

[redacted]





1. An Introduction to Computerized Tomography

The word *tomography* derives from the Greek word *tomos* meaning "slice" plus graph meaning "picture". The method of tomography examines the inside of a three-dimensional object by creating two-dimensional images of cross-sections of the object. Each image is created by passing radiation through one plane of the object, measuring its attenuation, and using that attenuation to map the density of the object in that plane. Because computers are used to create the image from the measured data, tomography is often called *computerized tomography* (CT). Originally, the abbreviation CAT stood for *Cross-Axial Tomography*, but now it is interpreted as *Computer Aided Tomography*, we will simply use CT.

In the 1970s, computerized tomography revolutionized diagnostic radiology. In the brain, for example, CT can readily detect tumors and internal bleeding without the need of exploratory surgery. The 1979 Nobel Prize in Medicine was awarded for work in computerized tomography.

For a CT scan, a patient sits or lies inside a ring mounted with an X-ray source directly opposite an X-ray detector. Figure 1 sketches a possible setup for a CT scan. A set of parallel rays of X-ray photons is directed through the patient's body. When a ray passes through a body part, some X-ray photons are absorbed, with dense materials such as bone and tumors absorbing more than soft muscles and skin. The detector measures the number of photons passed through the body and so determines how much the ray was attenuated by absorption. The average density of the body along the path of each ray can then be determined by comparing the incident and transmitted intensities of the ray.

By rotating the ring around the patient, rays can be sent along any number of paths in one plane of the body. Using the attenuation readings for all of the rays, it is possible to create an approximate map of the density of the object within the plane, translating the ring along the body and repeating the procedure builds a sequence of two-dimensional images that together form a rough three-dimensional image of the interior of the object.

Because we pass only a finite number of rays through the body, we cannot find an exact or continuous map of the density throughout the slice. Instead, to reconstruct the cross-sectional image from the thousands of recorded beam measurements, the reconstruction region is subdivided into $n \times n$ small squares, called *pixels* (from "picture elements"). This set of pixels form what is often referred to as a *grid.* The width of each pixel is chosen according to the width of the detectors and/or the width of the X-ray beams. This sets the resolution of the image. The General Electric CT/T system uses 102,400 pixels in a 320×320 array. To each pixel there is associated a number, called its CT number, or its X-ray density number. This number is a measure of photon attenuation as the beam of X-ray photons passes through the pixel; it will be defined more precisely in what follows. The determination of these pixel CT numbers is the basic mathematical problem of computerized tomography.

Once these numbers have been determined, the cross-sectional image can be displayed on the video monitor, since from the CT number of each pixel, a "grayness" value can be assigned, and an image constructed that is made of varying shades of gray. Different structures within the body have different X-ray densities and thus can be distinguished in the image.

If r rays are passed through a $\sqrt{n} \times \sqrt{n}$ grid with $r > n$, we are required to solve an overdetermined linear system, with a $r \times n$ coefficient matrix, to produce a digitized image of the object in one plane. An overdetermined system is one with more equations then unknowns, see figure 3. Such a system typically does not have an exact solution, so the image is reconstructed from the best approximate solution. The mathematical details of how to reconstruct the image from the data will be covered later.

The first published description of CT was authored by Sir Godfrey Hounsfield of EMI Ltd. in London and appeared in 1973. Hounsfield's scans used X-rays of very low intensity, and it took many hours of exposure to gather the data. An eight by eight grid was superimposed on the object, and the attenuations of sixty four rays were measured. The resulting 64 x 64 system took hours to solve on EMI's then state-of-the-art computer.

Present day problems are much larger but take less time to solve. The resolution desired for modern CT scans demands that the grid boxes be at most 1-3 mm on a side. This means that a 148 x 148 or greater grid is used for a typical brain scan. The radiation source is moved to pass X-rays through each row of the grid at many different angles of incidence. When 148 rays are passed at each of 180 different angles, a total of 26,640 rays are passed. This translates into a 26,640 by 21,904 linear system. Recorded intensities from each ray are sent directly to the computer where the image is reconstructed.

In these Modules we will consider an iterative reconstruction technique used in CT. The "Solution" of a large system of linear equations is involved.

The Model

We now concentrate on reconstructing an image via a CT scan. Recall that for a CT scan, X-rays are passed through one plane of an object from various angles. The intensity of each ray is measured before and after it passes through the object. In this section, we review the mathematical fundamentals of the process of mapping the density of the object from the measured ray attenuations.

If we introduce a variable s that measures the distance from the source along a ray, we can write down an expression for how the intensity of a ray changes as it passes through an object assuming that the ray travels in the xy-plane. Specifically, the intensity I changes with respect to the distance s according to

$$\frac{dI}{ds} = -\mu(x,y)I, \tag{1.1}$$

where $\mu(x,y)$ is the density of the object. Because the density and intensity must always be nonnegative, the negative sign in equation 1.1 shows that, if the intensity changes, it decreases with increasing distance s.

To relate the initial and transmitted intensities of the ray to the density, we must group all terms involving the intensity and integrate the resulting equation

$$\frac{dI}{I} = -\mu(x,y)ds. \tag{1.2}$$

The left-hand side is a definite integral in terms of I. If I_o is the initial intensity of the ray and I_T is its final intensity,

$$\int_{I_o}^{I_T} \frac{dI}{I} = \ln(I_T/I_o) = -\ln(I_o/I_T). \tag{1.3}$$

To integrate the right-hand side of 1.1, we must integrate a function of x and y with respect to the variable s. This is not inconsistent because the distance s along the ray is itself a function of x and y. If the beam originates at the point (x_o, y_o) and distance s_o from the origin, the length from that point to any other point (x,y) on the ray is

$$s - s_o = \sqrt{(x-x_o)^2 + (y-y_o)^2}. \tag{1.4}$$

To integrate the right-hand side we must use a special sort of integral known as a *line* or *path integral*. If σ denotes the line in the xy—plane followed by the ray, the line integral is written

$$\int_{\sigma} \mu(x,y)ds. \tag{1.5}$$

The usual definite integral $\int_a^b f(x)dx$ measures the area beneath the curve of the integrand between $x = a$ and $x = b$ by summing infinitesimally small increments of area between those points. In contrast, a line integral measures the "weight" of the curve itself. For example, if the object is of constant density $\mu(x,y) = \gamma$ and the ray is of length s_T, the line integral is just the length of the line times the density of the object

$$\int_{\sigma} \mu(x,y)ds = \gamma s_T. \tag{1.6}$$

Integrating equation 1.2

$$In(I_o/I_T) = \int_{\sigma} \mu(x,y)ds \tag{1.7}$$

then tells us how the ratio of the initial and transmitted intensities of a ray is related to the amount of material through which the ray passes. When the density of the object depends explicitly on x and y, we must rewrite the integral to remove the dependence on s before we can evaluate the integral. Details of this procedure are presented in most calculus books. Line integrals arise in many problems in mathematics and physics, and the path over which one integrates need not be a straight line.

The Discretized Problem

When recreating an image by tomography we do not evaluate the line integral of equation 1.7. Indeed, the density $\mu(x, y)$ is the unknown quantity we are trying to find. In 1917, Radon showed how to extract the density from the right-hand side of equation 1.7 by transform methods, but his formulas are based on continuous projection data instead of the finite set of measurements produced in an actual scan. They are inaccurate when applied to finite data sets, especially those subject to some experimental error. In addition, his formulas do not lend themselves to an efficient computational algorithm. Hence, subsequent research has focused on developing a good computational algorithm. In these modules, we will develop a discretized formulation of the image reconstruction problem that can be solved on a computer. We show how the problem translates into an overdetermined system. We will derive a method that is typically used to solve the resulting discretized problem in medical and other image reconstruction applications.

Because we use a finite number of rays, we cannot obtain a continuous approximation for the density $\mu = \mu(x, y)$ over the entire grid, so we instead approximate the density in all boxes of the grid. If we are studying a $\sqrt{n} \times \sqrt{n}$ grid, we assume that the density is constant within each box. Denoting the density in box B_i by μ_i, for $i = 1, ..., n$, we then compute an approximation to μ of the following form:

$$\mu \approx \mu_1 \text{in } B_1 \text{ and } \mu_2 \text{ in } B_2 \text{ and ... and } \mu_n \text{ in } B_n. \tag{1.8}$$

To make this approximation easier to work with, we introduce a new function δ_i which equals 1 inside box i but equals 0 in any other box. Rewriting the grid itself as a matrix allows us to represent the discrete problem as a system of linear equations. For the 3×3 example, we can write the approximate density in the grid in matrix form as follows

$$\mu \approx \begin{pmatrix} \mu_1 & \mu_2 & \mu_3 \\ \mu_4 & \mu_5 & \mu_6 \\ \mu_7 & \mu_8 & \mu_9 \end{pmatrix} \tag{1.9}$$

$$
\begin{aligned}
&= \mu_1 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \ldots + \mu_9 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&= \mu_1\delta_1 + \mu_2\delta_2 + \ldots + \mu_9\delta_9. \qquad (1.10)
\end{aligned}
$$

For a grid with n boxes, we can write

$$
\begin{aligned}
\mu &\approx \mu_1\delta_1 + \mu_2\delta_2 + \ldots + \mu_n\delta_n \qquad (1.11) \\
&= \sum_{i=1}^{n} \mu_i\delta_i. \qquad (1.12)
\end{aligned}
$$

For ray σ_j, the path integral from equation 1.7 then becomes

$$
\int_{\sigma_j} \mu(x,y)ds \approx \int_{\sigma_j} \sum_{i=1}^{n} \mu_i\delta_i. \qquad (1.13)
$$

If the measured attenuation for ray δ_j is denoted

$$
p_j = \ln\left(I_o/I_T\right), \qquad (1.14)
$$

then we may define the relations

$$
p_j = \int_{\sigma_j} \mu(x,y)ds \approx \int_{\sigma_j} \sum_{i=1}^{n} \mu_i\delta_i = \sum_{i=1}^{n} \mu_i \int_{\sigma_j} \delta_i, \;\; j = 1, \ldots, n. \qquad (1.15)
$$

If we make r attenuation measurements p_j, $j = 1, \ldots, r$, equation 1.15 defines a rectangular system of equations

$$
M\mu = p, \qquad (1.16)
$$

where the $r \times n$ coefficient matrix has elements $M_{ji} = \int_{\sigma_j} \delta_i$ for $j = 1, \ldots, r$ and $i = 1, \ldots, n$. The vector p on the right-hand side has r elements $p(j) = p_j$, and the solution vector μ has n elements $\mu(i) = \mu_i$.

2. An Introduction to Backprojection

Backprojection is a mechanism for deducing the density of an object from measured ray attenuations. How can one recreate an image from measured data? One may use a two-dimensional grid with its boxes colored either black or white and attempt to determine the coloring of each box by passing rays vertically or horizontally through the grid. At its origin, a ray is assigned a value of zero. When it emerges from the grid, it has an integer value equal to the number of black boxes through which it has passed. In this module we will examine this method for 3×3 grids.

Figure 1 shows the values that would be measured by passing three rays horizontally through the given 3×3 grid. To make the source of the measurements clearer, the black and white boxes are also shown. The rays passed through the first and last rows of the grid have the value 3. Thus, all three boxes in those rows must be black. The ray passed through the second row has value 2 and this tells us only that two of the three boxes in that row are black. To determine its exact structure, we must pass more rays through the grid. Figure 2 shows the measured values for three vertical rays. These data tell us that the first and third columns of the grid are colored black. Because the first and third rows are also black, the center vertical measurement 2 means that the center square must be white.

It is not difficult to devise an example for which this simple deductive algorithm fails. Figure 3 shows horizontal and vertical measurements, we can be certain only that the center row of the grid has three white boxes. Either of the two configurations in figure 4 would give the same readings. In this 3×3 case, we can discern the correct pattern by sending one more ray diagonally through the grid from upper left to lower right. This single diagonal ray when passed through the grid on the left in figure 4 will pass through two black boxes. On the other hand, a diagonal ray passed through the grid on the right in figure 4 will pass through only one.

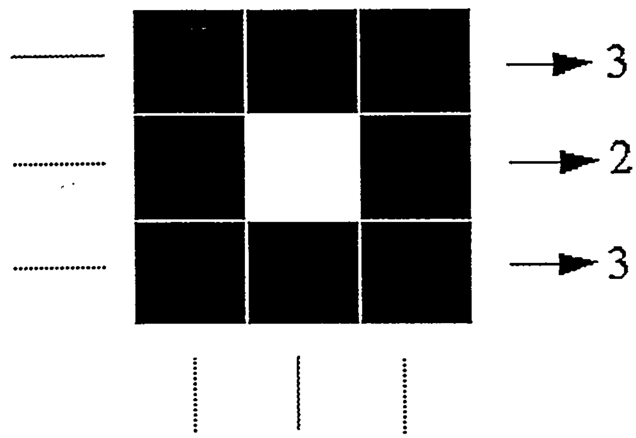


Figure 1: The result of passing horizontal rays through the grid.

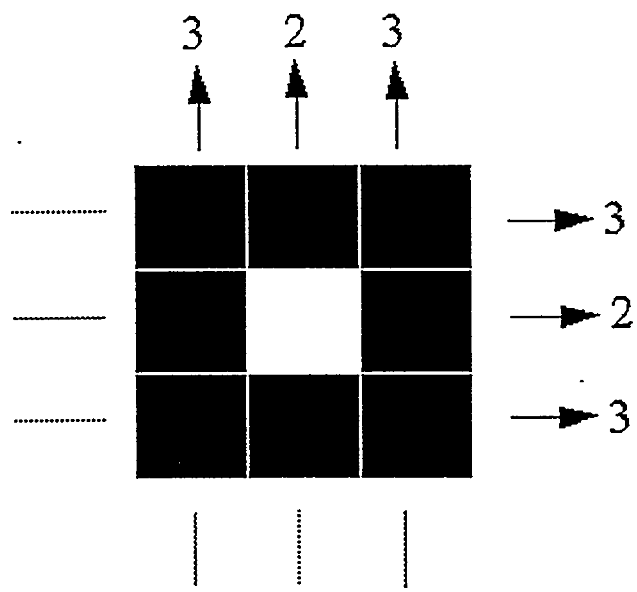


Figure 2: The result of passing horizontal and vertical rays through the grid.

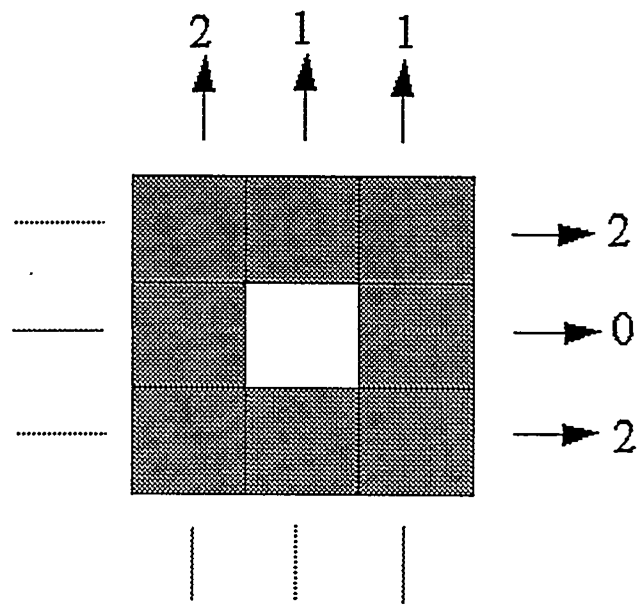


Figure 3: The result of passing horizontal and vertical rays through the unknown grid.

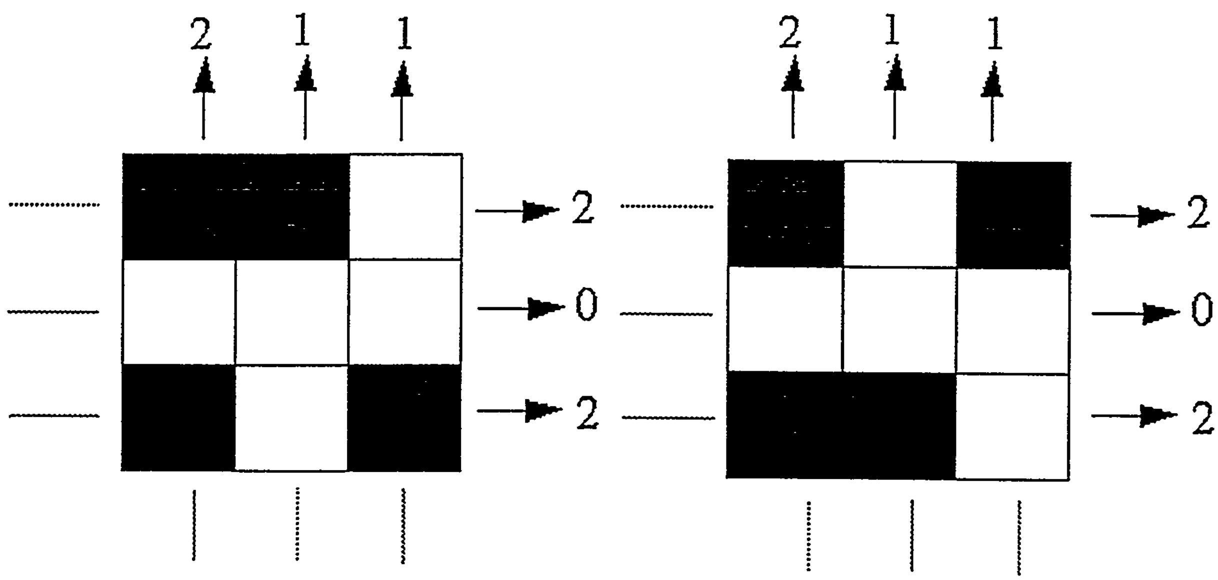


Figure 4: The two possible colorings of the grid.

An alternative to this precise deductive process is to produce a gray scale coloring of the grid. In this case, a ray's value upon exit is divided by the number of boxes through which it passes. All boxes in the ray's path are then colored the same shade of gray defined by that value. In this way, the measured value of the ray is *backprojected* along its path. Repeating this process for many rays produces a rough approximation to the black and white image in varying shades of gray.

The gray scale coloring of the grid of figure 2 is shown at the top in figure 5. In this case, the horizontal rays assign values of 1, 2/3, and 1 to the boxes in the first, second, and third rows, respectively. The vertical rays give values of 1, 2/3, and 1 to the boxes in the first, second, and third columns. Thus, each box has been assigned two values -one from the horizontal ray and one from the vertical ray. Summing these values and dividing by the number of rays per box gives the average value per ray. The shade of gray corresponding to each value appears on the grid in figure 5. While the black bordering rows and columns of the grid are not exactly resolved by this process, the center box correctly appears lighter than the surrounding ones. A better representation could be obtained by passing more rays through the grid or by combining the gray scale and deductive algorithms to recognize such features as a fully blackened row or column.

These procedures give the most fundamental idea behind the process of backprojection. In order to accurately reproduce larger and more complicated images, however, it is necessary to turn to the more sophisticated procedure that is the subject of the next module.

The Filtered Backprojection Method

To present a mathematical formulation of filtered backprojection, we first assign unique angle and distance parameters to each ray as shown in figure 8. The origin is located at the center of the grid overlaying the object. The line L runs through the origin in the same direction as the rays. The ray σ_j is identified by its perpendicular distance t_j from L and the angle θ that the perpendicular to the ray makes with the $x-$axis. The measured attenuation of σ_j after it passes through the object is denoted by

1	5/6	1
5/6	4/6	5/6
1	5/6	1

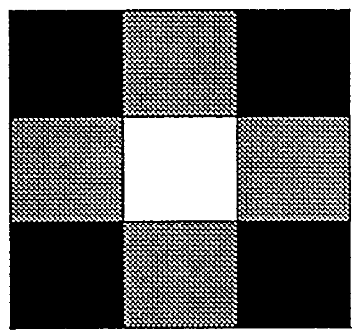


Figure 5: A gray scale rendering of Figure 2

$$p_j = p(\theta, t_j) = \int_{\sigma_j} \mu(x,y) ds. \tag{2.1}$$

Our goal is to derive a mathematical relationship between the measured attenuations p_j and the density $\mu(x,y)$that we are trying to determine.

To reproduce the image, we need to measure the attenuations of a large number of rays. We organize the attenuations by passing K equally spaced parallel rays $\sigma_o, ..., \sigma_{K-1}$ through the object at each of q equally spaced angles $\theta_o, ..., \theta_{q-1}$ for a total of $r = qK$ attenuation measurements. For example, at each angle θ, we send K rays at distances from the line L equal to $t_o, t_1, ..., t_{K-1}$ with $t_j = t_o + j\Delta t$.

The collection of attenuation values measured for one set of the parallel lines comprises a *parallel projection.*

The projection data and the density are related via *Fourier transforms.* In the next module we will develop this relationship.

3. The Fourier Transform and Its Inverse

The *Fourier Transform* is a way to convert a continuous function of one variable to a continuous function of the frequency of that variable. For example, a function of space is transformed to a function of spatial frequency, and a function of time is transformed to a function of temporal frequency. The Fourier transform is generally applied when it is more convenient to do a computation in the frequency domain, and we will see that this is indeed the case for *backprojection*.

If $f(x)$ is a continuous one-dimensional function of distance x and

$$\int_{-\infty}^{+\infty} |f(x)|\, dx < +\infty,$$

then Fourier transform of $f(x)$ is defined by

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-2\pi iux} dx,$$

where $F(u)$ is a one-dimensional function of spatial frequency. We can extract the function $f(x)$ from its Fourier transform by means of its *inverse Fourier transform*

$$f(x) = \int_{-\infty}^{+\infty} F(u) e^{2\pi iux} du.$$

The Fourier transform can be also be applied to higher-dimensional functions. The two-dimensional Fourier transform of the function $f(x, y)$ is

$$F(u, v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x, y)e^{-2\pi i(ux+vy)}dxdy,$$

and the inverse Fourier transform of $F(u, v)$ is

$$f(x, y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u, v)e^{2\pi i(ux+vy)}dudv.$$

A two-dimensional function can also be transformed in only one of its variables. For example, the two-dimensional function $f(x, y)$ can be transformed in the x dimension alone as

$$F(u, y) = \int_{-\infty}^{+\infty} f(x, y)e^{-2\pi iux}dx \tag{3.1}$$

or in the y dimension alone as

$$F(x, v) = \int_{-\infty}^{+\infty} f(x, y)e^{-2\pi ivy}dy. \tag{3.2}$$

This property implies that we can actually replace a two-dimensional Fourier transform by a pair of one-dimensional Fourier transforms taken in turn. One possible organization is as follows:

$$F(u, v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x, y)e^{-2\pi i(ux+vy)}dxdy \tag{3.3}$$

$$= \int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} f(x,y) e^{-2\pi i u x} dx \right] e^{-2\pi i v y} dy \tag{3.4}$$

$$= \int_{-\infty}^{+\infty} F(u,y) e^{-2\pi i v y} dy. \tag{3.5}$$

Alternatively, we can transform the function $f(x, y)$ first in the variable y and then in x to form

$$F(u,v) = \int_{-\infty}^{+\infty} F(x,v) e^{-2\pi i u x} dx. \tag{3.6}$$

In addition, the Fourier transform it not confined to the Cartesian coordinate system. For instance, a function $g(\theta, t)$ expressed in polar coordinates can be transformed in the angular variable θ or the radial variable t or in both by integrating over the full ranges of those variables:

$$G(\eta,\rho) = \int_{0}^{2\pi} \int_{0}^{+\infty} g(\theta,t) e^{-2\pi i(\eta\theta+\rho t)} dt d\theta. \tag{3.7}$$

In some cases, we need to take the Fourier transform not of one function but rather of the special integral of the product of two functions called *convolution* and defined in one dimension by

$$f * g \equiv \int_{-\infty}^{+\infty} f(\alpha) g(x-\alpha) d\alpha. \tag{3.8}$$

The Fourier transform of a convolution is the product of the Fourier transforms of the functions used in that convolution. Thus, if the Fourier transforms of $f(x)$ and $g(x)$ are available, the process of taking the Fourier transform of $f * g$ reduces to a simple multiplication in frequency space.

$$\int_{-\infty}^{+\infty} (f * g)e^{-2\pi iux}dx = F(u)G(u) \tag{3.9}$$

$$= \left(\int_{-\infty}^{+\infty} f(x)e^{-2\pi ux}dx\right)\left(\int_{-\infty}^{+\infty} g(x)e^{-2\pi ux}dx\right) \tag{3.10}$$

This result is a statement of the *Convolution Theorem.* Taking the inverse Fourier transform of both sides of equation (3.14) gives an alternative definition of the convolution

$$f * g \equiv \int_{-\infty}^{+\infty} F(u)G(u)e^{2\pi ux}du. \tag{3.11}$$

In the case of *filtered backprojection*, the functions with which we work represent an image or its Fourier transform. The image $\mu(\theta, t)$ is a map of the density in the xy-plane expressed in terms of polar coordinates. Its Fourier transform $U(\theta, \rho)$ is thus a function of spatial frequency, also expressed in polar coordinates. An image for which $U(\theta, \rho)$ is large when ρ is large is one with rapid variation in density across the xy−plane. If this variation is not actually a property of the depicted object, such an image is termed *noisy.* The Convolution Theorem gives an easy way of improving the quality of noisy images. For example, a function $g(x - \alpha)$ can be constructed so that the convolution of $U(\theta, \rho)$ and g either removes or enhances the contribution to the image $\mu(x, y)$ of certain frequencies. Thus, the Fourier transform $G(\rho)$ of the filter function g acts as a filter in frequency space. Filters are also a necessary part of the most basic backprojection algorithm.

A Continuous Formulation of Backprojection

Now that the basic tools of the backprojection algorithm have been defined, we can return to the problem of how to produce a discrete approximation of the density

$$\mu(x,y) \approx \sum_{j=1}^{n} \mu_j \delta_j \tag{3.12}$$

from the measured projection data $p_o, ..., p_{K-1}$. While these data are clearly discrete, it simplifies the derivation of the backprojection method to assume first that we instead have a continuous parallel projection for a given angle θ. This corresponds to passing an infinity ($K \rightarrow \infty$) of rays through the object at the angle θ and collecting their attenuations p_j, $j = 0, ..., +\infty$, to form a continuous function $p(\theta, t)$. In this case, t is a continuous variable measuring the perpendicular distance from the line L through the origin to the ray. Although t varies continuously from $-\infty$ to $+\infty$, the projection can have nonzero values only for those values of t within the confines of the object. The full range of t is included only for convenience in the derivation.

We construct the relation between the continuous projection and the density by first considering the special case of $\theta = 0$ and then generalizing that result to hold for any value of θ. Both the special and general cases rely upon the Fourier transform and inverse Fourier transform.

The special case $\theta = 0$

We first relate the density and the projection for a continuous projection taken parallel to the y–axis. The angle of this projection is $\theta = 0$, and the line L running through the origin at angle $\theta = 0$ is the y–axis itself. This means that the perpendicular distance t from the ray to the line L is just the x–coordinate of that ray. That is, $t = x$, and

$$p(\theta = 0, t) = p(\theta = 0, x) = \int_{\sigma} \mu(x, y) ds. \tag{3.13}$$

The distance along the ray from its starting point is

$$s - s_o = \sqrt{(x - x_o)^2 + (y - y_o)^2}. \tag{3.14}$$

However, when $t = x$, this distance varies with y alone so that $ds = dy$. Thus, when $\theta = 0$, the line integral along the path of the ray can be written as the definite integral in y

$$p(\theta = 0, t) = p(\theta = 0, x) = \int_{\sigma} \mu(x, y) dy = \int_{-\infty}^{+\infty} \mu(x, y) dy. \tag{3.15}$$

Again, we include the full range of y for convenience even though only those y values between the emitter and detector can actually contribute to the value of the integral.

It is not immediately clear how to obtain an expression for the density from the above equation, but the Fourier transform provides the key. To see this, we first write the two-dimensional Fourier transform of the density

$$U(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mu(x, y) e^{-2\pi i(ux+vy)} dx dy. \tag{3.16}$$

If we then consider the case $v = 0$, we are left with a two-dimensional Fourier transform of the density along the u–axis in frequency space

$$\begin{aligned} U(u, 0) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mu(x, y) e^{-2\pi iux} dx dy & (3.17) \\ &= \int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} \mu(x, y) dy \right] e^{-2\pi iux} dx. & (3.18) \end{aligned}$$

Notice that the expression in the square brackets in this equation is just $p(\theta = 0, x)$ so we actually have the important result

$$U(u, 0) = \int_{-\infty}^{+\infty} p(\theta = 0, x) e^{-2\pi iux} dx. \tag{3.19}$$

That is, taking the one-dimensional Fourier transform of the projection $p(\theta = 0, x)$ along the $y-$axis gives us one line in frequency space (namely, the $u-$axis) of the two-dimensional Fourier transform of the density,

Thus, we can extract one line ($t = x$) of the density $\mu(x, y)$ by taking the one-dimensional Fourier transform of $U(u, 0)$. This operation *backprojects* $U(u, 0)$ from frequency space onto the line $t = x$ in the spatial domain. If we could generalize this result to produce any line of $U(u, v)$, we could use it to find $U(u, v)$ in all of frequency space. By taking the inverse Fourier transform of that representation of $U(u, v)$ for all u and v, we could determine the density $\mu(x, y)$ for all x and y.

Using any old θ

As it turns out, our equations are easily modified to apply to any angle θ. With the simple matrix operation known as the *Jacobi rotation*, we can get other lines of the Fourier transform of the density $\mu(x, y)$ from projections taken at arbitrary angles θ.

A Jacobi rotation $\Im$ is defined by a 2×2 matrix function of an angle θ. Applying this matrix to a vector in the $xy-$plane rotates the vector about the angle θ in that plane as follows:

$$\Im \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x\cos\theta + y\sin\theta \\ -x\sin\theta + y\cos\theta \end{pmatrix} = \begin{pmatrix} t \\ s \end{pmatrix} \tag{3.20}$$

The Jacobi rotation lets us rotate the $xy-$coordinate system into the $ts-$coordinate system. This means that we can treat a projection taken at an angle θ in the $xy-$plane as a projection taken at an angle 0 in the $ts-$plane. The $ts-$coordinate system is a natural one to use for our problem as the variable t represents the distance of the ray from the line L through the origin, and the variable s represents the distance travelled along the ray from its source. For example, the point at location (x_o, y_o) in the figure below has $s-$ and $t-$coordinates s_o and t_o, where $s_o = -x_o \sin\theta + y_o \cos\theta$ and $t_o = x_o \cos\theta + y_o \sin\theta$.

Following the same steps as we used in the $xy-$coordinate system to derive equation (3.14), we can write down the equation for a parallel projection in the $ts-$plane running parallel to the $s-$axis:

$$p(\theta, t) = \int_{-\infty}^{+\infty} \mu(t, s) ds. \tag{3.21}$$

Its one-dimensional Fourier transform is

$$P(\theta, \rho) = \int_{-\infty}^{+\infty} p(\theta, t) e^{-2\pi i \rho t} dt \tag{3.22}$$

$$= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mu(t, s) e^{-2\pi i \rho t} ds dt. \tag{3.23}$$

Rewriting this in $xy-$coordinates gives us

$$P(\theta, \rho) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mu(x, y) e^{-2\pi i \rho (x \cos\theta + y \sin\theta)} dx dy = U(\theta, \rho). \tag{3.24}$$

Discrete Filtered Backprojection

Earlier we defined the Fourier Transform method, which gives us a relationship between the projections and the density of the object. The projection measured at a given angle is a collection of values denoted

$$p = (p_o, p_1, ..., p_{K-1})^T. \tag{3.25}$$

We repeat this measurement for q different angles $\theta_o, ..., \theta_{q-1}$. We then show how to discretize both the Fourier transform and the filtered backprojection algorithm to operate on the discrete data.

The Discrete Fourier Transform and Its Inverse

The discrete Fourier transform is a mechanism for transforming a set of measurements of the spatial function $f(x)$ to a set of values of a function of spatial frequency $F(u)$.

We assume that we have an even number of measurements of the continuous function $f(x)$ taken at the equally spaced $x-$values so that $x_o, ..., x_{K-1}$. That is, $f(x_j) = f_j$ for $J = 0, ..., K-1$. We also assume that the function $f(x)$ is zero outside of the range $[x_o, x_{K-1}]$ so that our samples f_j represent all important parts of $f(x)$. The sum

$$F_l = F(u_l)/\Delta x = \sum_{j=0}^{K-1} f_j e^{-2\pi i l j/K} \tag{3.26}$$

defines the *discrete Fourier transform* of these data at frequency u_l.

Note that the terms of the sum form a periodic series in l. This is, because $e^{-2\pi i j} = 1$ for all integers values of j, $F_{-l} = F_{K-1}$. In particular, $F_{-K/2} = F_{K/2}$, so we need only compute the K values $F_{-K/2}, ..., F_{K/2-1}$ to have all information about F. The K data values $f_o, ..., f_{K-1}$ thus actually lead to K distinct Fourier transform values $F_{-K/2}, ..., F_{K/2-1}$.

Similarly, for $J = 0, ..., K-1$,

$$f_j = (1/K) \sum_{l=-K/2}^{K/2-1} F_l e^{2\pi i l j/K} \tag{3.27}$$

defines the discrete inverse Fourier transform at x_j of values $F_o, ..., F_K$ in frequency space.

The convolution $f * g$ has the K elements

$$h_l = (f * g)_l = \sum_{j=0}^{K-1} f(j)g(l-j). \tag{3.28}$$

This convolution is well-defined for $l = 0, ..., K-1$ when the data f and g are assumed periodic with period K. In that case, the elements of g with negative indices are evaluated by the relation $g(-m) = g(k-m)$. This sort of convolution is termed a *periodic* convolution.

Applying a periodic convolution directly to aperiodic data results in an incorrect result as the terms involving elements $g_{-(K-1):1}$ contribute incorrectly to the result. This *interperiod interference* is remedied by affixing zero elements to the data vectors.

The convolution is related to the Fourier transform of f and g by

$$H_l = \sum_{j=-K/2}^{K/2-1} (f * g)_j e^{-2\pi i l j/K} = F_l G_l, \tag{3.29}$$

for $l = -K/2, ..., K/2-1$.

The discrete Fourier Transform can also be applied in two dimensions. The two-dimensional discrete Fourier transform of the function $f(x,y)$ for samples taken at $f_{jk} = f(x_j, y_k)$ for $j, k = 0, ..., K-1$ is

$$F_{lm} = F(u_l, v_m)/\Delta x \Delta y = \sum_{j=0}^{K-1} \sum_{p=0}^{K-1} f_{jk} e^{-2\pi i (lj/K + mk/K)}. \tag{3.30}$$

The two-dimensional discrete inverse Fourier transform is

$$f_{jk} = (1/K) \sum_{l=-K/2}^{K/2-1} \sum_{m=-K/2}^{K/2-1} F_{lm} e^{-2\pi i (lj/K + mk/K)}. \tag{3.31}$$

As in the continuous case, the 2D discrete Fourier transform can be written as a pair of one-dimensional discrete Fourier transforms:

$$F_{lm} = \sum_{j=0}^{K-1} \left[\sum_{k=0}^{K-1} f_{jk} e^{-2\pi imk/K} \right] e^{-2\pi ilj/K}. \tag{3.32}$$

Continuous Backprojection in a finite frequency Domain

One can developed a continuous formulation of filtered backprojection by relating the density $\mu(x, y)$ of the object to the projection data as follows:

$$\mu(x, y) = \int_0^{\pi} C(\theta, t) d\theta, \tag{3.33}$$

where the transform

$$C(\theta, t) = \int_{-\infty}^{+\infty} P(\theta, t) \left|\rho\right| e^{-2\pi i \rho t} d\rho \tag{3.34}$$

defines a filtered projection.

Discrete Formulation

$$\mu(x, y) \approx \frac{\pi}{q} \sum_{l=0}^{q-1} C(\theta_l, t) \tag{3.35}$$

4. Iterative Reconstruction Technique

It is known that, as an X-ray beam passes through an object, some of the photons of the beam are absorbed by the object (photon attenuation). With this in mind, consider a line of n pixels, through which an X-ray beam passes squarely (See the figure below)

Suppose that the first pixel transmits a fraction f_1 of the incident photons, the second pixel a fraction f_2 of the photons incident to it, and so on, to the n^{th} pixel, which transmits a fraction f_n (*i.e.*, f_i equals the number of photons entering the i^{th} pixel divided by the number of photons leaving).

The total fraction, f, transmitted through this line of pixels will be given by

$$f = f_1 \times f_2 \times f_3 \times \cdots \times f_n. \tag{4.1}$$

Hence

$$\ln f = \ln f_1 + \ln f_2 + \ln f_3 + \cdots + \ln f_n, \tag{4.2}$$

or equivalently,

$$-\ln f = -\ln f_1 - \ln f_2 - \ln f_3 - \cdots - \ln f_n. \tag{4.3}$$

The positive quantity $-\ln f_1$ is called the CT number (X-ray density) of the first pixel and will be denoted by μ_1. Similarly, $\mu_2 \equiv -\ln f_2$ is the CT number of the second pixel, etc., with $-\ln f$ as the total X-ray density of the beam. This last quantity is called the *ray sum* of the beam and will be denoted by s.

Thus, if the i^{th} beam, with ray sum, s_i, passes squarely through a line of n pixels, whose pixel numbers are $j_1, j_2, \ldots, j_n$,then

$$\mu_{j_1} + \mu_{j_2} + \cdots + \mu_{j_n} = s_i \tag{4.4}$$

where s_i is known from the actual and calibration measurements, and the $\mu_{j_1}, \mu_{j_2}, ..., \mu_{j_n}$ are to be determined.

However, not all beams of a scan pass squarely through a line of pixels. Instead, the i^{th} beam may pass "diagonally" through the pixel in its path. In this case, we have

$$\sum_{j=1}^{N} w_{ij}\mu_j = s_i, \tag{4.5}$$

where w_{ij} is a weighting factor that represents the contribution of the j^{th} pixel to the i^{th} ray sum, and $N = n^2$ (the total number of pixels).

If the beam width is the same as the pixel width, then theoretically w_{ij} equals the ratio of the area of intersection of the i^{th} beam with the j^{th} pixel to the area of the j^{th} pixel. However, due to the computational difficulty of finding the area of intersection of the beam and the pixel, other definitions of w_{ij} are sometimes used. Two such definitions are:

1. $w_{ij} = 1$ if the i^{th} beam passes through the center of the j^{th} pixel, and $w_{ij} = 0$, otherwise.

2. $w_{ij} =$ length of the center line of the i^{th} beam that lies in the j^{th} pixel, divided by the width of the j^{th} pixel.

The first definition of w_{ij} is easier to use than the second but is less accurate. Either of these definitions gives rise to the following system

$$M\mu = s.$$

There are various methods for solving linear systems -Gaussian elimination, matrix inversion, the Gauss-Seidel method, etc. However, because of the nature of the applied problem under consideration here, which gives rise to the above system, the following points must be taken into account in solving the system:

- The ray sums $s_o, ..., s_{K-1}$, which form the right-hand side of the system, cannot be measured exactly. There will always be experimental error in the data collected. Hence the system is usually inconsistant, and the best one can hope for is some "approximate solution." Methods of solution which assume that the system is consistent cannot, in general be used.

- In computerized tomography, the number of scans taken in the data collection process forces the system to be overdetermined, i.e., so that $M > N$. Methods of solution which assume that $M = N$, therefore, cannot be used.
- Our system can be so large that direct methods of solution are not feasible, due to computer requirements on storage and time.

Many mathematical approaches are being tried in the area of image reconstruction in computerized tomography. We next describe one such approach, an iterative reconstruction technique, which produces approximate solutions to the linear system.

To understand the *Iterative Reconstruction Technique*, let's first consider the following system of three linear equations in two unknowns:

$$1\mu_1 + 1\mu_2 = 3 \tag{4.6}$$
$$1\mu_1 + 4\mu_2 = 4 \tag{4.7}$$
$$3\mu_1 - 1\mu_2 = -1 \tag{4.8}$$

Geometrically, this system determines three straight lines L_1, L_2,and L_3 in the $\mu_1\mu_2$−plane. These lines do not have a common intersection point, i.e., the system is inconsistent. However, points on the triangle ABC formed by the three lines can be considered as "approximate solutions" of the system. (If the system were consistent, the triangle would shrink to a point, *the solution* of the system).

The following is an iterative procedure that generates points on the triangle ABC ("approximate solutions" of the system):

Choose an arbitrary point P_o in the $\mu_1\mu_2$−plane. Project P_o orthogonally onto L_1 to get the point $P_1^{(1)}$. Project $P_1^{(1)}$ orthogonally onto L_2, to get $P_2^{(1)}$

orthogonally onto L_3 to get $P_3^{(1)}$. The first iteration is now complete and results in the points $P_1^{(1)}, P_2^{(1)}$, and $P_3^{(1)}$on lines L_1, L_2, and L_3.

Now use $P_3^{(1)}$ as P_o, etc., to obtain three sequences of points:

$$P_1^{(1)}, P_1^{(2)}, P_1^{(3)}, \ldots \text{ on } L_1, \tag{4.9}$$

$$P_2^{(1)}, P_2^{(2)}, P_2^{(3)}, \ldots \text{ on } L_2, \tag{4.10}$$

$$P_3^{(1)}, P_3^{(2)}, P_3^{(3)}, \ldots \text{ on } L_3. \tag{4.11}$$

These sequences converge to points P_1^*, P_2^*, and P_3^*, say, on lines L_1, L_2, and L_3; and the three limiting points are independent of the starting points as long as the three lines are not all parallel.

We need a formula for the orthogonal projection of a point onto a line. Suppose that $Q(q_1, q_2)$ is the orthogonal projection of the point $P(p_1, p_2)$ onto the line L in the $\mu_1\mu_2-$plane described by $w_1\mu_1 + w_2\mu_2 = s$.

Using vector notation and dot products with

$$\overrightarrow{\mu} = (\mu_1, \mu_2),\ \overrightarrow{w} = (w_1, w_2),\ \overrightarrow{p} = (p_1, p_2),\ \text{and } \overrightarrow{q} = (q_1, q_2), \tag{4.12}$$

Then the projection point is given by

$$\overrightarrow{q} = \overrightarrow{p} + \left(\frac{s - \overrightarrow{w} \cdot \overrightarrow{p}}{\overrightarrow{w} \cdot \overrightarrow{w}}\right) \overrightarrow{w}. \tag{4.13}$$

HOMEWORK

(1) Applying the iterative procedure to our 3×2 system with $P_0 = (2, 3)$. Doing four iterations.

Iterative Reconstruction Technique Algorithm

If we set $\overrightarrow{\mu} = (\mu_1, \mu_2, ..., \mu_n)$ and $\overrightarrow{w_i} = (w_{i1}, w_{i2}, ..., w_{iN})$, $i = 1, 2, ..., M$, then the system can be expressed as $\overrightarrow{w_i} \cdot \overrightarrow{\mu} = s_i, i = 1, 2, ...M$. In algorithm form, the steps of the iteration are:

1. Choose P_o, or in vector form, $\overrightarrow{p_o}^{(1)}$
2. Set $r = 1$, for the first iterate.
3. Compute

$$\overrightarrow{p_k}^{(r)} = \overrightarrow{p_{k-1}}^{(r)} + \left(\frac{s_k - \overrightarrow{w_k} \cdot \overrightarrow{p_{k-1}}^{(r)}}{\overrightarrow{w_k} \cdot \overrightarrow{w_k}} \right) \overrightarrow{w_k} \tag{4.14}$$

for $k = 1, 2, ..., M.$(r = iteration count, M = # of hyperplanes)

4. Set $\overrightarrow{p_o}^{(1)} = \overrightarrow{p_M}^{(r)}$.
5. Increase the iterate number r by 1 and return to step 3.

From this, M sequences of points are obtained:

$$P_1^{(1)}, P_1^{(2)}, P_1^{(3)}, ... \text{ on the first hyperplane,} \tag{4.15}$$
$$P_2^{(1)}, P_2^{(2)}, P_2^{(3)}, ... \text{ on the second hyperplane,} \tag{4.16}$$
$$.... \tag{4.17}$$
$$P_M^{(1)}, P_M^{(2)}, P_M^{(3)}, ... \text{ on the } M^{th} \text{ hyperplane;} \tag{4.18}$$

and it can be shown that these sequences converge to points $P_1^*, P_2^*, ..., P_M^*$, say, on the M hyperplanes, and that the limiting points are independent of the starting point P_o, as long as the vectors $\overrightarrow{w_1}, \overrightarrow{w2}, ..., \overrightarrow{w_M}$ span R^N.

One of the points $P_1^{(r)}, P_2^{(r)}, ..., P_M^{(r)}$, with r sufficiently large (depending on the desired accuracy), is used as an approximate solution of the system and hence used in the cross-sectional image reconstruction. The decision of which approximate solution to use is based on different kinds of secondary criteria, which are beyond the scope of our modules.