DYNAMICAL SYSTEMS EDUCATION ON THE WWW

PANKAJ K. KAMTHAN

A THESIS
IN
THE DEPARTMENT
OF
MATHEMATICS AND STATISTICS

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER IN THE TEACHING OF MATHEMATICS
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

SEPTEMBER, 1997

National Library of Canada

Acquisitions and Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada

Acquisitions et services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

0-612-44803-7

Abstract

Dynamical Systems Education on the WWW

Pankaj K. Kamthan

The study of dynamical systems is significant in mathematics education. The medium of the Internet and the World-Wide Web (WWW) opens new avenues in dynamical systems education.

The following aspects of the WWW technology pertaining to dynamical systems education are studied: representation of mathematical documentation; issues in design and development of computer programs for carrying out client- and server-side computations; use of multimedia, in particular computer-graphics, animation and sound; significance of locally accessible databases and search strategies; use of interactive and noninteractive communication tools; development of tools for assessment. Usability, advantages and limitations of the tools and techniques involved are pointed out. Applications to teaching dynamical systems topics in traditional areas of mathematics, and to teaching the specific topic of iteration of functions and fixed points in a real-environment of classroom are discussed.

Acknowledgments

I take this opportunity to thank those who directly or indirectly contributed towards the success of this thesis.

Supervisor

Professor Eusebius Doedel, Department of Computer Science, Concordia University supported my work for all these years with patience and kindness. His insight and numerous hours of stimulating conversations gave the thesis its structure.

Faculty, Department of Mathematics, Concordia University

Professors Abraham Boyarsky taught me most of what I know about dynamical systems. He initiated me to the topic of Section 8.5.2. Professor Pawel Góra introduced me to the "mathematics" of fractals. His through reading of Chapter 3 was very useful. Professor Anna Sierpinska's dedication to mathematics education has always been an inspiration. Her careful reading of Chapter 14 was very useful. Professor Joel Hillel critically reviewed the surveys (Chapter 4) included in the thesis prior to distribution. Professor William Byers suggested that I look at mathematical topics from a dynamical systems viewpoint, which motivated Chapter 13.

Faculty, Department of Physiology/Mathematics, McGill University

Professors Leon Glass and Michael Mackey created an outstanding research environment during my years at the Center for Nonlinear Dynamics. Dr. William Anglin gave me the philosophical "eye" towards mathematics. His critical reading of Chapter 3 was very helpful.

Staff, Department of Mathematics/Computer Science, Concordia University

Michael Assels, Paul Gill, Stan Swiercz and Bill Wong created a state-of-the-art computing environment. Nancy, Edwina and Angie cheerfully helped me through my many administrative problems.

Friends

Frank was there when I needed it the most. Kumi encouraged me with her love and support. Paul helped me generate some of the graphic images in the thesis.

Finally, this work was partially supported by Concordia University Part-time Faculty Association in the form of a Professional Development Grant and by the Centre for Teaching and Learning Services, Concordia University, through a Faculty Teaching Development Grant.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last few years, there has been a tremendous increase in the use of the Internet for mathematical communication as indicated by the workshop on *Communicating Mathematics with Hypertext*, at the Geometry Center, University of Minnesota, June 1996. Proceedings are available on the WWW at the URL

`http://www.geom.umn.edu/events/courses/1996/cmwh/`.

Internet-based information systems, such as the WWW, are becoming an accepted standard for disseminating and sharing information.

NCTM's February, 1994, official position statement on *The Use of Technology in the Learning and Teaching of Mathematics* recommends the appropriate use of technology to enhance mathematics programs at all levels. Information technology for representing and analyzing information, computing technology for processing information, and communication technology for communicating this information, have affected ways of teaching and learning of mathematics [28]. The WWW, through its support for hypermedia and multimedia, brings various aspects to these technologies. It is then important that these changes in technology, owing to the WWW, are reflected in mathematics education. Efforts in that direction are shown recently at the *International World Wide Web Conferences*. Proceedings are available on the WWW at the URL

`http://www.w3.org/Conferences/Overview-WWW.html`.

In this thesis, we analyze the effect of the medium of the WWW on dynamical systems education. Throughout the thesis, we ask ourselves the following questions: How has the evolution of the WWW affected the subject of dynamical systems? How has the WWW influenced existing tools and techniques involved in different aspects of dynamical systems education? Specifically, what novelty has WWW brought in dynamical systems education?

The thesis is divided into three parts: Part I (Chapters 2–4) provides the motivation and relevant background needed for further chapters, Part II (Chapters 5–12) studies the issues involved in the design and development of a WWW-based environment for dynamical systems education from a perspective which can be called as WWW-based dynamical systems software engineering, and finally, Part III (Chapters 13–14) discusses some applications of the tools from Part II to the real environment of teaching and learning.

In Chapter 2, we give an overview of the basic concepts of the Internet and the WWW along with their advantages and limitations. Advantages and obstacles of using WWW in education are discussed.

Chapter 3, studies the following questions: Why is dynamical systems' study of significance? Why should dynamical systems be taught? Why should it be learnt?

Chapter 4 presents surveys with an analysis of responses, which might help examine the current beliefs and practices in dynamical systems teaching and learning. Results of the surveys motivated other chapters in the thesis in various ways.

There are various issues that need to be taken into consideration in developing an educational WWW site. Chapter 5 discusses the choices that should be made in WWW hardware and software set up.

Until recently, the primary source of teaching and learning mathematical subjects have been (paper-based) documentation in the form of books, lecture notes, etc. This medium changes dramatically when one uses the WWW-technology to develop such (electronically-based) documentation. In Chapter 7, we explore the design issues that need to be considered in this transition, and the possibilities and problems in representing mathematics on WWW in a dynamical systems context.

Chapter 8, studies the following questions: What aspects/properties do we want to know of dynamical systems and in which of these aspects/properties computation is useful? is necessary? is feasible? and how? The evolution of the WWW has strongly influenced the entire computing environment by unifying many aspects. How can the Internet and the WWW be used for dynamical systems-related computations? It also discusses the issue of didactical engineering when using the WWW for teaching and learning.

In dynamical systems study, information is generated in various formats: text, graphics, and sound, i.e., in multimedia. In Chapter 9, we study the significance, innovative possibilities, advantages and limitations of using multimedia in dynamical systems education and the role of the WWW in its delivery.

One of the educational values of the WWW, since its inception, has been the availability of teaching and learning resources on it. As WWW continues to grow rapidly, the complexity of accessing this vast pool of information has increased many-fold, and developing local databases on specialized topics has become significant. Furthermore, it requires teachers and students to be familiar with existing search tools and techniques, particularly when searching mathematically sensitive information. Chapter 10 discusses advantages and examples of locally accessible databases, and a comparitive analysis of WWW search tools and techniques that can assist in successful searching.

Teaching and learning can be considered a highly interactive processes, requiring different levels of communication between the teacher and the student. Chapter 11 discusses the educational uses, advantages and limitations of various means of WWW-based interactive and non-interactive communication.

In order to evaluate learning, a measure of assessment is central to an educational environment. In Chapter 12, we discuss possibilities of using the technology of the WWW for developing tools for assessment.

Chapter 13 asks the following question: How can dynamical systems topics be integrated in teaching *existing* areas of mathematics? We shall undertake certain topics from traditional areas of mathematics and discuss how certain dynamical systems concepts can be integrated in them naturally.

Chapter 14 is based on the following question: How can dynamical systems be introduced in the curriculum (in the simplest possible manner)? The topic of iteration of functions and fixed points was chosen for its simplicity from a dynamical systems perspetive: both from the viewpoint of the functions (one–dimensional from a closed interval into itself) and the asymptotic behaviour they can exhibit (fixed point).

In Chapter 15, we indicate some directions for future development and finally, Appendix A presents an outline for a dynamical systems course WWW 'home page'.

Part I

Dynamical Systems Educational Environment on the WWW: Motivation and Background

Chapter 2

Internet, WWW and Education: Preliminaries

The purpose of this Chapter is to give the requisite background of terms and concepts from Internet and WWW, which have been used in this thesis.

2.1 The Internet

Transmission control protocol/Internet protocol (TCP/IP) referes to an entire suit of data communication protocols. Internet is a global network (or network of networks) which uses TCP/IP family of protocols to communicate. This network could consist of computers. Other views of the Internet are described in Section 10.3.1.

2.2 The World-Wide Web

The World-Wide Web (WWW) is an Internet-based distributed hypertext information system. This information can consist of various different file formats (multimedia) and can contain links (known as *hyperlinks*) to other media (hypermedia). The unit of organization for the WWW is a WWW page. A WWW page is a document that is retrieved and displayed by a WWW browser in response to a single request by the user.

2.3 Hypertext and Hypertext Markup Language

The term *hypertext* describes a text which contains hyperlinks to other media. Hypertext Markup Language (HTML), is a document-layout and hyperlink-specification language. It is designed to structure documents and make their content more accessible rather than format documents for display purposes. HTML has content-based style tags that attach meaning to text passages. The text within the tags is interpreted and displayed in a certain manner by the browser. For example, the text between <TITLE>...</TITLE> as the title of the document, <FORM>...</FORM> indicates a fill-out-form.

2.4 WWW-Specific Protocols

Hypertext Transfer Protocol (HTTP) is the primary protocol used by the WWW to communicate and is built on the top of TCP/IP. For backward compatibility, provision has been made for other existing high-level protocols (for example, FTP, GOPHER, WAIS, etc.) also to be used for communication over the WWW. This facilitates connecting servers (and services) based on these protocols to the WWW.

2.4.1 HTTP

When a WWW browser is instructed to fetch an HTTP URL, it opens a connection to the indicated HTTP server, sends its request, receives a response, and displays the contents of the response to the user. The response consists of a line containing the protocol version, a three-digit numeric status code, and a text explanation of the status. The status codes are divided into four categories: codes in the range 200 to 209 indicate a successful transaction, codes in the range 300 to 399 are used when the URL can not be retrieved because the document has moved to a different location, codes in the range 400 to 499 range are used when the browser has made an error, such as making an unauthorized request, and codes 500 and up occur when the server can not comply with the request because of an internal error. For some of the frequently occuring errors in these categories, HTML documents or scripts can be designed with appropriate messages and the user can be directed to them automatically using the WWW server configuration files (for example, `srm.conf` in NCSA and Apache servers).

Limitations

The HTTP protocol is stateless. Connections to the WWW server are made only at the user's request, and between connections the WWW server does not maintain any information for or about individual users.

2.5 Multipurpose Internet Mail Extensions

Multipurpose Internet Mail Extensions (MIME) is an extensible system developed for sending multimedia data, over Internet mail. The WWW having similar needs, adopted MIME as a part of HTTP. The MIME typing system allows virtually any document to be sent over the Internet and displayed (or played, or executed) on the user's computer. Both WWW browsers and servers use MIME. On the client-side, the client software can specify a list of preferred file types when it requests a document from the server. If the server has choices available to it, it can preferentially pick one of the formats requested by the browser. When a WWW server transmits a document to the browser, it precedes the body of the document with a short header that includes the document's MIME type. WWW browsers make it possible to accomodate new document types.

2.6 WWW Clients and Servers

The WWW is based on a client–server relationship. One program (client) makes the request using a WWW protocol and another program (server) complies to the request. (Often, the computer on which the client and server programs are running are identified by the same correponding name.) On the WWW, the client is a WWW browser. A WWW browser is an interface to the WWW.

2.7 Uniform Resource Locators

A Uniform Resource Locater (URL) indicates the protocol, host and the location of an Internet resource. Its format is

`protocol://hostname:port/path.`

URLs are of three types: absolute, partial and relative. Complete URLs contain all parts of the URL, including the protocol part, the host name part, and the document path. In partial URLs, the protocol part and the host name part are ommitted and the URL begins with the path name part. In relative URLs, only the document name (in the path) is included.

2.8 The Common Gateway Interface

The Common Gateway Interface (CGI) is a standard for interfacing external applications with WWW servers. A plain HTML document that the WWW browser retrieves is static, i.e., it exists in a constant state. A CGI program, on the other hand, is executed in real-time, so that it can output dynamic information. Figure 1 illustrates the cycle of CGI communication that takes place between the WWW browser and server.

Since a CGI program is executable, it allows others to run a program on one's system which can compromise system security. For security reasons, CGI programs need to reside in a special directory (`cgi-bin` on NCSA and Apache servers) so that the WWW server knows to *execute* the program rather than just display it to the browser. This directory has restricted access. More details on CGI security are given in Section 5.6.

A CGI program can be written in any language that allows it to be executed on the system. It is preferable to write CGI *scripts* instead of programs, since they are easier to debug, modify, and maintain than a typical compiled program. Also, the script itself only needs to reside in the CGI directory, since there is no associated source code. Even though almost any programming language can be used to write CGI scripts, Perl [41] remains the language of choice due to its text manipulation and system programming abilities, and rich set of module libraries.

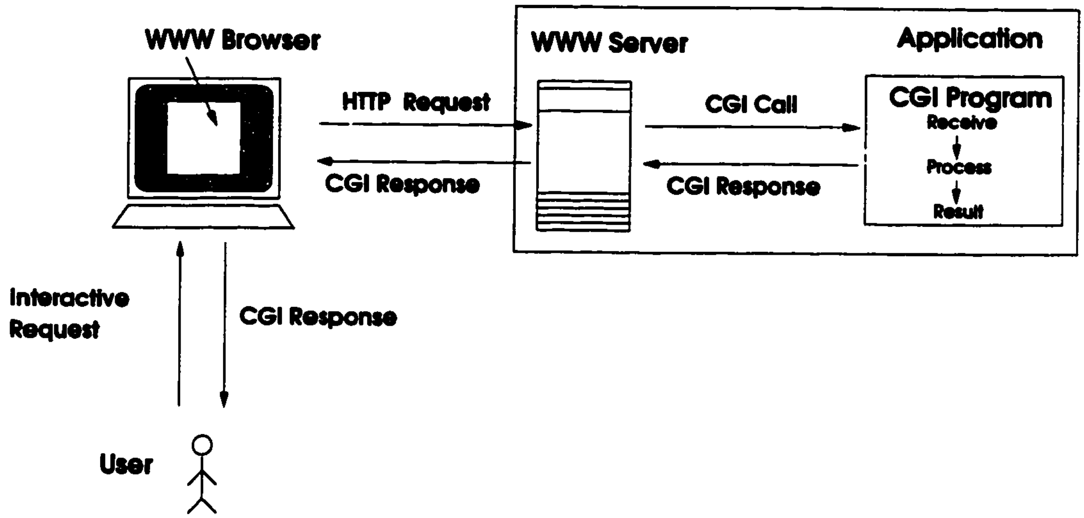


Figure 1: CGI Communication between the WWW Client and Server.

2.8.1 Limitations

Every CGI transaction involves a two-way communication with a remote server which can result in a lengthy wait. Also, if the server machine becomes overloaded/is down, it is unable to serve new users. CGI programs thus become inaccessible. With CGI programs, results can not be saved locally on the client-side. Furthermore, the user interactions and the program output in CGI programs are generally limited to the small set of user interface tools provided by HTML forms. It therefore is important that teh access to a CGI program is restricted to users with priority and should not take too long to process.

2.9 Java and the WWW

Java is a programming language from Sun Microsystems that has been developed in recent years. Java's origins go back to 1991, when Sun Microsystems began looking for ways to create platform-independent code to support consumer electronic products.

Advantages of Java are reflected in its definition [23]:

> A simple, object-oriented, distributed, interpreted, robust, secure, architectural neutral, portable, high-performance, multithreaded, and dynamic language.

Java is a network language for distributing executable content. With appropriate security measures, this process revolutionizes the WWW by transforming the networks into software distribution system and the WWW browser into an operating system for running all kinds of applications.

Java Program Cycle

There are four different types of programs one can write using Java: *applications, applets, content handlers*, and *protocol handlers*. Java applications are standalone programs that

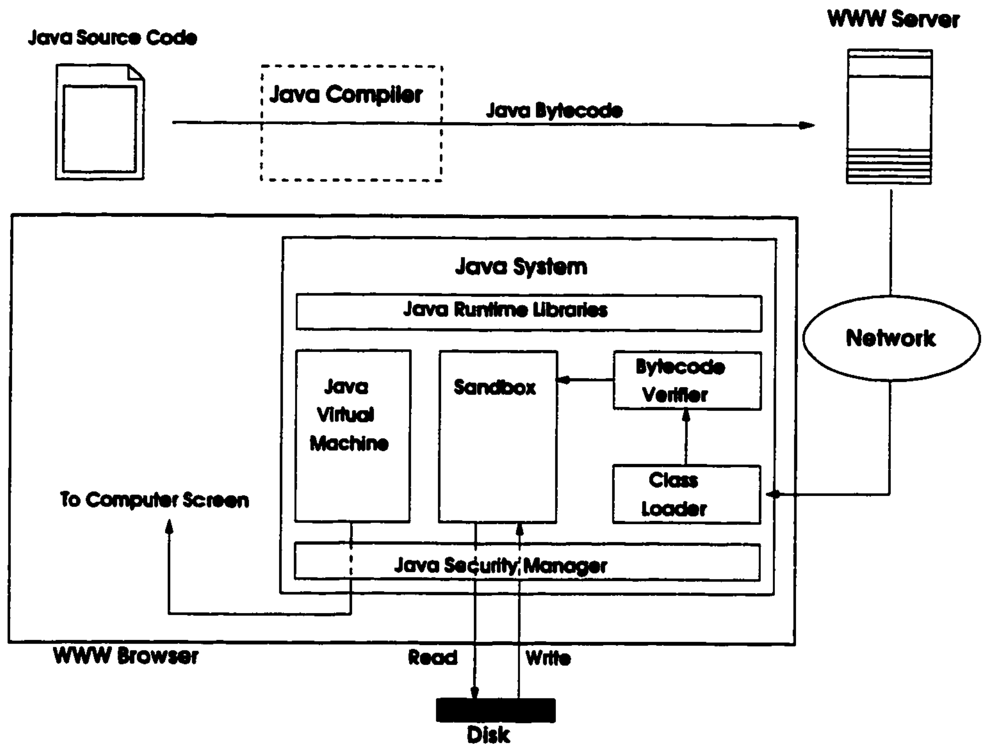


Figure 2: The Life Cycle of a Java Applet.

require the assistance of the Java interpreter to run. However, they run independently of any WWW browser. Java applets are small applications written in Java. A Java applet requires a WWW browser (or another Java application) to run. Applets can be embedded in HTML pages. Java applets are more security-conscious than Java applications; there are many restrictions on the behaviour of applets (see Section 5.6). The remaining two types of Java programs, content and protocol handlers, are special-purpose Java programs that allow WWW browsers to dynamically understand new data types and new protocols. In this thesis, we shall be concerned only with Java applets.

A Java program (a `*.java` file) is compiled into a platform-independent *bytecode*(a `*.class` file). This bytecode is loaded into a computer's memory by the *Java Class Loader*. Finally, the bytecode is run on a Java runtime environment *Java Virtual Machine* (JVM). The JVM can run Java programs directly on an operating system; alternatively, the JVM can be embedded inside a WWW browser, allowing programs to be executed as they are downloaded from the WWW. The JVM can execute the Java bytecode directly using an interpreter; or it can use a Just-In-Time (JIT) compiler to convert the bytecode into the native machine code of the computer it is running. Figure 2 shows the stages that take place in the life cycle of a Java applet: from when it is written to when it is displayed on the WWW browser.

2.9.1 Java Applets vs. CGI Scripts

Java applets, using the WWW browser support client-side programming as opposed to server-side programming with CGI scripts. There are various advantages of Java applets over CGI scripts. CGI scripts run only on server-side and so can be slow, do not provide rich set of client-side interactions, are restricted with the limited interface features of HTML. Java applets, on the other hand, can be run on the client-side, do not have any overload problems, are highly interactive, feedback is immediate and the user does not have to deal with a series of page reloads.

2.10 JavaScript

JavaScript is a object-based scripting language as a result of a joint effort from Netscape Communications and SUN Microsystems. Some advantages of JavaScript are:

- **Interactivity.** JavaScript through its built-in event-handling mechanisms can be used to drive many "events" that occur when a user interacts with an HTML page. JavaScript statements embedded in an HTML page can recognize and respond to user events such as mouse clicks, form input, and page navigation. One can use JavaScript to verify that users enter valid information into an HTML form or to perform an action (such as play an audio file, execute an applet, or communicate with a plug-in) in response to the user opening or exiting a page. Another useful application is to write *cookies*, which are programs which maintain client-server state persistence (and thus circumvent statelessness of HTTP).
- **Portability.** JavaScript code, like that of HTML, is platform-independent.

2.10.1 JavaScript and CGI

One of the most useful features of the WWW is its ability to foster interaction. Until recently, this interaction has been achieved almost exclusively by sending messages back and forth between the client machine and the server machine, via a CGI. The slower the connection (or greater the traffic) between the two machines, the longer one had to wait for the response to the request.

JavaScript, for most part, is designed to complement CGI pre-processing. For example, one can write a JavaScript function to check variables in input fields. Without any network transmission, an HTML page with embedded JavaScript code can interpret the entered text and alert the user with a message dialog if the input is invalid. JavaScript executes on the client at the request of the user and verifies that all of the input boxes in a given HTML form are filled in or have valid data ranges. It then either prompts the user to correct the data or if correct, sends the data back to the WWW server to execute the CGI script. This integration of the two standards saves valuable server-side CPU time and Internet bandwidth, as the script will not have to run multiple times. In comparison with CGI, JavaScript code is much more efficient and platform-independent.

JavaScript also can be used for small scripts that handle quick functions or calculations that would otherwise require the aid of (time consuming) server-based scripts. This is also useful for those developing scripts and do not have direct access to CGI directories on the server. There is a special HTML tag, SCRIPT: all information in <SCRIPT>...</SCRIPT> is interpreted as JavaScript code. If the SCRIPT tag is placed in the HEAD section, of the HTML file the code within that script is executed before the body of the HTML page is completely downloaded. This allows interpretation of any code to be done *before* the user enters any text into the form itself, so events can be handled properly. The JavaScript script need not be located within the same HTML file; it can be located elsewhere and downloaded separately when an HTML page is invoked by a WWW browser.

2.10.2 JavaScript vs. Java Applets

Java applets do not generally interact with the HTML of a WWW page. Although an applet can be made to communicate with several other applets on the same page, it cannot, however, *change* the text of the same HTML page in which it is located. JavaScript, on the other hand, is designed to bring HTML interactivity. An input box in one HTML form can modify the HTML information inside another HTML page. There are ways of controlling a browser from within Java applets. However, for simple tasks like calculations within forms, changing other frames within the browser, etc., takes a substantial amount of coding. JavaScript is made to implement such functionality without spending long time writing Java code.

JavaScript is immediately interpreted by the browser from the source code, whereas Java applets are precompiled into a class file before actual use. This means that the scripts may run slower; each line is interpreted separately, rather than having faster compiled code that can be immediately executed by the JVM within the browser.

Since JavaScript is object-based, there does not exist a large set of class libraries that one can use with JavaScript. This limits the scripts to simple calculations and event processing.

2.10.3 Limitations

JavaScript is an interpreted language which allows flexibility but also sacrifices performance. The cross-platform ability comes from each browser being able to interpret the code into platform-specific instructions. The drawback is that extremely complicated scripts are not possible. JavaScript also has some limitations, imposed mostly for security purposes that restrict its utility. The most severe of which is that JavaScript can not write a file to a WWW server's hard disk. Thus, CGI must be used to create interactive forms that append data to a file.

2.11 Virtual Reality Modeling Language

The ability to *navigate* the WWW in 3D allows us to experience information in an intuitive way that matches the human model of the world. The success and impact of HTML on the deployment of multimedia on the Internet and the significance of 3D graphics (in form of virtual reality) motivated the design of Virtual Reality Modeling Language (VRML) [8] which began in 1993.

VRML is a interchange format for describing 3D shapes and interactive environments on the WWW. VRML is object-oriented. Objects in these worlds can be hyperlinked to text, image, audio, video or HTML files, as well as to other VRML "worlds".

VRML gives world authors the ability to create compelling and interactive experiences, with animation, multimedia and multi-user capabilities. Users can directly interact with and manipulate objects in the VRML world, through the use of sophisticated animations and programmable behaviours, all controlled by common WWW scripting languages. In addition, VRML offers authors the ability to incorporate a variety of digital media, create 3D user interfaces, and extend the language for their own needs. They can then "package" these components for re-use by other world creators. In addition to 3D animation, VRML supports such multimedia features as text, graphics, audio and video. It also gives developers the capability to embed URLs in virtual worlds, so that a user could click on an object, for example, and be transported to another VRML world on the WWW. VRML has been designed to work in conjunction with HTML, Java and JavaScript.

2.12 WWW and Education

2.12.1 Advantages of the WWW in Education

There are various advantages of a WWW-based learning environment:

- **Open Access.** Many remote resources, including library and other databases are openly accessible via the WWW. Using the WWW, large amounts of complex information can be disseminated over long distances at extremely fast speeds.
- **Open Collaboration.** The WWW forms world-wide communities in many ways: by sharing information (such as through WWW documents, e-mail) and by helping others (such as answering posted questions on Usenet newsgroups). It links groups of teachers and students into "communities of interest" to pursue learning.
- **Open User.** WWW is multiuser environment which supports learners from a wide age group and skill range.
- **Open Communication.** WWW provides a medium for asynchronous communication and synchronous communication.
- **Open Publication.** The WWW, by supporting multimedia, creates an environment for publishing information in various formats that the information can be represented.

- **Open Classroom.** The concept of asynchronous distance learning[1] is designed to help overcome the limitations inherent in traditional instructional techniques, limitations which involve both time and space. The growth of the WWW has opened new potential for asynchronous and distant instructional 'classroom'.

  Asynchronous learning can lead to students forming communities and interacting with other students (and teachers) even if they never physically interact. If students can learn course material while working at their computers in their own time, the need for physical proximity is reduced. This can lead to open global classrooms. Teachers and students can access information *any time* and *anywhere*.

- **Open Computation.** The WWW supports the notion of "write once, run anywhere" for computer programs. Such programs, however, may not necessarily be the most (time or space) efficient but they suffice for educational purposes. With the interface of a WWW browser, students and teachers from all grade levels can engage in discovery-based exploration.

In the subsequent chapters, we discuss how these advantages can benefit various aspects of dynamical systems education.

2.12.2 Obstacles to Use of WWW in Education

Presently, the use of WWW in education faces the following obstacles:

- **Training.** There is a lack of teacher/student training of WWW-related technology.
- **Accessibility.** At present, the access to the Internet is limited to certain countries around the world. There also, it is accessible mostly to Universities, and less to schools and colleges. This situation can be expected to improve with time.
- **Reliability of Information.** There is no guarantee of the accuracy of information on the WWW. Issues of accuracy are, for example, mistakes (typographical, factual, incorrect URLs, etc.) and outdated information.
- **Searching.** There is a lack of specialized search engines. Often, significant amount of time needed to obtain any meaningful information.

Some of the above problems might be solved in due course of time. However, as more and more information is added on the WWW and the number of users increase, following problems will continue to arise:

- Connection capabilities, speed of network connections and servers.
- Searching for relevant information in an acceptable stipulated amount of time.
- Managing information.

[1] Instruction is asynchronous when it does not constrain the student to involvement in the learning process at a specific time, for example, when lectures are presented at a *fixed* time of the day. Instruction is distant when it does not constrain the student to be physically present in a specific location, for example, when lectures are presented only in a *fixed* classroom.

Chapter 3

Significance of Dynamical Systems Education

Chaos and Fractals — The real importance of these ideas is not in the applications that will stem from them [...] I feel that these ideas will have their biggest impact in mathematics education.

— Robert L. Devaney

A century ago, it was known that some deterministic dynamical systems, such as the *three body problem* of celestial mechanics, can exhibit extremely complex behaviour. One of the remarkable discoveries of mathematics of the last few decades is that very simple systems, even systems depending on only one variable, can have very 'rich' dynamical behaviour ranging from the existence of a fixed point to chaos.

Change in the practice of mathematics forces re–examination of mathematics education. About two decades ago, in his review article, May [49] analyzed the dynamics of the *logistic equation*

$$\tau_a(x) = ax(1-x), \quad 0 < a \leq 4, \quad x \in [0,1],$$

which has become a paradigm for one–dimensional systems with chaotic behaviour. There he states:

> I would [...] urge that people be introduced to the logistic equation early in their mathematical education. This equation can be studied phemenologically by iterating it on a calculator, or even by hand. Its study does not involve as much conceptual sophistication as does elementary calculus. Such a study would greatly enrich the student's intuition about nonlinear systems.

Inspite of May's emphatic message, the inclusion of dynamical systems in mathematics education has taken a while and it is only recently that mathematicians have felt that some knowledge of the subject is imperative. As a result, recommendations have been made to offer dynamical systems related topics/courses in the curriculum at various levels ranging from high school to university.

In this Chapter, we address the following questions: How can the teaching and learning of dynamical systems benefit mathematics education? What is the role of the computer and the WWW technology in this endeavour?

3.1 Significance of Dynamical Systems

We here assess the significance of dynamical systems from two points of view: their *omnipresence* and *applications* within and outside mathematics.

3.1.1 Dynamical Systems Everywhere

Mathematics Root System

The root system of mathematics consists of deep ideas that nourish the growing branches of mathematics. A sound education in mathematics requires encounter with virtually all of these very different perspectives and ideas. They can be listed [67] as follows:

Mathematics structures. Numbers, *algorithms*, ratios, shapes, *functions*, *data*.

Attributes. *Linear*, *periodic*, *symmetric*, *continuous*, *random*, maximum, appropriate, *smooth*.

Actions. *Represent*, *control*, prove, *discover*, apply, *model*, *experiment*, *classify*, *visualize*, *compute*.

Abstractions. Symbols, infinity, optimization, logic, *equivalence*, *change*, *similarity*, *recursion*.

Attitudes. Wonder, meaning, *beauty*, *reality*.

Behaviours. *Motion*, *chaos*, *resonance*, *iteration*, *stability*, *convergence*, *bifurcation*, *oscillation*.

Dichotomies. *Discrete vs. continuous*, *finite vs. infinite*, *algorithmic vs. existential*, *stochastic vs. deterministic*, *exact vs. approximate*.

The perspectives indicated above which are emphasized appear specifically as a part of the dynamical systems terminology. This reflects the ubiquity of dynamical systems in the mathematics root system.

Natural and Artificial Phenomena

We live in a world which is *dynamic* and gives rise to various dynamical phenomena: hydrodynamical *turbulence* in a stream running down a mountain, *rhythms* in the heart, *fluctuations* in the Dow Jones stock index, are just a few examples. The omnipresence of dynamical systems in natural and artificial phenomena is one of the factors why the field has recently received so much attention. Furthermore, there are many reasons why we need to understand the dynamics: *to forecast* the weather, *to prevent* heart diseases, *to control* agricultural pests, *to foresee* the consequences of human activities on the environment, *to design* more reliable and efficient machines, *to improve* communication in a network, *to predict* the stock market, and so on.

3.1.2 Relevance of Dynamical Systems

Advocating the Platonic vision 'For the Good', Browder and Mac Lane [7] state the following in context the of relevance of the mathematics:

> The relevance of mathematics involves both the various applications of mathematics and the position of mathematics in the spectrum of human values. ... the customary division of mathematical research into pure and applied mathematics is not the most effective way to understand the relevance of mathematics. One and the same mathematical idea can apply to totally different disciplines. [... An] example of interrelation of "pure" and "applied" mathematics is the study of the long–term properties of the solutions of differential equations.

Therefore, if one considers the contribution of dynamical systems to the subject of mathematics by categorizing it as a branch of *applied* mathematics, it might be a restrictive view of assessing its relevance. The field of dynamical systems is *inherently* interdisciplinary: it borrows tools from various different areas in mathematics for its existence. Its study has given birth to many areas of mathematics, both 'pure' (e.g., topology) and 'applied' (e.g., bifurcation theory) and has brought various results from them together.

Another perspective that has brought significance to the study of dynamical systems is its *utility* in fields other than mathematics. For example, the concept of 'chaos' has stimulated important technical developments in the way we can analyze and interpret medical time series data which offer evidence of novel diagnostic approaches that could lead to new preventive techniques and treatment strategies [26].

3.2 Mathematical Connections and Dynamical Systems

> *Connections give mathematics power and help determine what is fundamental. Pedagogically, connections permit insight developed in one [concept] to infuse into others.*
>
> — Lynn A. Steen

Poincaré made important contributions to the field of dynamical systems but his contributions to mathematics in general went well beyond that[1]. His advocacy of geometrical and topological methods in ordinary differential equations (ODEs) opened up whole news branches in mathematics such as algebraic and differential topology. Many concepts and techniques from dynamical systems arise naturally in *other* areas of mathematics and vice versa. Such interlinks within mathematics become transparent when one observes, e.g., 'finding the cube–root–of–8' problem can be approached in at least three different ways [54] — via factoring, and solving the cubic equation; via trigonometry, using the de Moivre Theorem; and via *dynamical* method of iteration, using say Newton's method.

[1]Poincaré is dubbed by E. T. Bell [3] as the "Last Universalist" (a man who is at ease in all branches of mathematics, both pure and applied).

The National Council of Teachers of Mathematics (NCTM) *Curriculum and Evaluation Standards for School Mathematics* [52], stresses the importance of mathematical connections and their interplay:

> The mathematics curriculum should include investigations of the connections and interplay among various mathematical topics and their applications so that all students can
> - recognize equivalent representation of the same concept;
> - relate procedures in one representation to procedures in an equivalent representation;
> - use and value the connections among mathematical topics;
> - use and value the connections between mathematics and other disciplines.

Furthermore, it is emphasized that the curriculum must invoke the full spectrum of the mathematical sciences. This gives an opportunity for students to see how different fields in mathematics can be applied in unison.

In [31, 32], specific concepts from different areas of mathematics which arise in topics in dynamical systems — such as iteration of functions, chaos, Mandelbrot set, etc. — are given. This allows drawing connections to these previously known concepts during teaching a dynamical systems topic to be *integrated* in those areas. On the other hand, dynamical systems concepts arise in some well known mathematics areas thus allowing these areas to be taught in an alternate way, from a *dynamical systems viewpoint.* We shall consider this in detail in Chapter 13.

3.3 Accessibility

When a certain branch of mathematics is introduced in the curriculum, it is necessary to question its *accessibility* by the existing educational standards. Many dynamical systems related concepts appear in various areas of mathematics which are already present even in the pre–university (high school/college) curriculum. Since certain sophisticated topics can be learned rather easily at an elementary level, it seems useful to incorporate them in a pre–university curriculum. For example, with only the knowledge of elementary calculus, students can comprehend basic mathematical ideas behind topics as the Julia set. This can give the idea to the students that powerful results in mathematics can be established using very basic tools and can be accessible at an elementary level. As Devaney [17] points out, one of the reasons for the interest in the areas of 'chaos' and 'fractals' is that many topics in these fields are indeed accessible.

3.4 Dynamical Systems, Activity and Recreation

I hear, I forget; I see, I remember; I do, I understand.

— Paul R. Halmos

Mathematics as envisioned in *Curriculum and Evaluation Standards* of the NCTM [52, page 7] is epitomized in student activities: "*knowing* mathematics is *doing* mathematics" and instruction should persistently emphasize that. Much mathematics can be learned by activities which are 'alive' with action. Dynamical systems education provides ample opportunity for that. A didactical study of teaching difference equations with emphasis on student activity has been carried out in [45].

There are various classroom activities that can be carried out to introduce various dynamical systems concepts which besides being 'academic' could also be 'recreational'. We list some of such activities:

- **Astronomy.** Astronomy is one of the areas which has long been a motivation for the study of dynamical systems. There are now computer softwares available which simulate planetary motion for two, three or more bodies. They can be used to carry out *computer experiments* by students to study the *behaviour* of regular elliptic *orbits* of two–body systems and complex behaviour of three or more bodies.

- **Physics.** Poincaré studied a fundamental question in astronomy: Is the solar system *stable*? In the course of doing so, he created what now is known as the modern theory of dynamical systems in which *stability* of the orbit or the equilibrium state (such as the attractor) of the system is a central topic of study. *Physical experiments* with devices such as pendulum, magnets and gyroscopes can help develop sound intuition among students about the notions of stability. Activities can be arranged to demonstrate different equilibrium behaviours. For example, a pendulum can be employed to observe how a *fixed point* arises in its motion.

- **Chemistry.** *Chemical experiments* can be carried out — such as the Belousov-Zhabotinskii reaction — by students to generate *periodic* patterns such as spirals. Patterns formed in the competition between reaction and diffusion serve as illustrative examples of *symmetry breaking* phenomena.

- **Geology.** *Geological experiments* with sandpiles to study *self-organized-criticality* and with various rocks with self-similarity to study *fractals* could be illuminating.

- **Music.** There are musical activities, such as playing music generated using the logistic equation or Chua's equations, that besides forming a learning experience, could also provide recreation. Further connections between dynamical systems and music have been drawn in Section 9.10.

3.5 Dynamical Systems and Experimental Mathematics

When we teach mathematics, it is important to question our approach: Is it an expedition or a guided tour that we are taking the students to? Since mathematics is a deductive study of patterns, we must view teaching not just as guiding students through the results discovered in the past but also introducing them to the excitement of exploring the unknown. Teaching dynamical systems offers mathematicians an opportunity to expose students to contemporary ideas in mathematical *research*. When equipped with the component of computer-based experiments, this gives a unique opportunity to students of combining rigorous mathematics with experimental ideas. These experiments can motivate 'dynamical ideas' and illustrate in a 'dynamic fashion' what the theorems *mean*. Such experimentation could assist students in developing an attitude of independent learning, in instilling a heuristic approach towards mathematics with the spirit of enquiry, in viewing mathematics as alive and growing, and hopefully in finding answers to some of the many unanswered questions. Since these experiments can give quick and interesting results, it can encourage students to pursue the subject.

3.6 For your Eyes

3.6.1 The Renaissance of Geometry and Dynamical Systems Education

The language of dynamical systems is geometry.

— John Guckenheimer

In the last two centuries, there has been a steady and progressive degradation of the geometric and kinesthetic elements of mathematics. In 17th century, due to the pioneering work of Descartes, geometry for the first time met with wide recognition and success and gave rise to the celebrated Cartesian geometry. This image receded and was replaced by algebraization. In the early 19th century, with the discovery of analytical objects such as continuous, nondifferentiable curves this situation declined further. By the end of the 19th century, geometry lost the notion of 'space' and became the study of abstract deductive systems.

The limits of deductivism have at last dawned on mathematics and a renaissance in geometry has occurred in recent years. Pattern detection and formation in mathematics which used to be algebraic has now become more and more visual. An example of this ferment in geometric ideas includes connections between developments in the theory of dynamical systems and the geometry of fractal sets. The role of computers with graphic capabilities has been pivotal in this endeavour by helping recognize the eye as an organ of discovery and inference.

Yet, as Malkevitch [47] says, "in terms of the way geometry is represented in the undergraduate curriculum, there has been no renaissance." Whether there will be one, is yet to be seen but a possible avenue for generating excitement about geometry lies in the rapidly growing field of 'computer-based visualization' of nonlinear dynamical systems and by teaching Visual Theorems[2].

[2] A visual theorem is the graphical or visual output from a computer program, which the eye organizes into a coherent and identifiable way [14], e.g., the self-similarity of the Mandelbrot set.

The introduction of dynamical systems equipped with computational tools can play a significant role in a reform of geometry: it enables many of the recommendations made in recent geometry reform projects [47] to be followed naturally. In Chapter 13, we shall discuss this in detail.

3.6.2 Dynamical Systems, Art and Beauty

> *It's like asking why Beethoven's Ninth Symphony is beautiful. If you don't see why, someone can't tell you. I know numbers are beautiful. If they aren't beautiful, nothing is.*
>
> — Paul Erdös

The interrelationship between mathematics and art is an exciting chapter of cultural history, which gives us a continuous and ever more comprehensive understanding of the world around us. From Matisse to Kandinsky's *Beitrag zur Analyse der Malerischen Elemente*, many modern artists have longed for a scientific approach to art; correspondingly, scientists have sought artistic inspiration from their work. After Kant published his *Kritiken*, the criteria for work of art and science were classified — that aesthetic information is subjective, while the scientific information is objective — and clearly separated. Inspite of that, mathematics continued to play a mediating role in work of artists like Escher [57].

Mathematics is the science of patterns which have various forms: numbers, equations, etc. The advent of computer graphics technology has given this relation a new form — computer art, in form of visualization of quantitative data resulting from dynamics of various systems. As an example, cellular automata, which are dynamical systems discrete in state, space and time, give rise to various *tiling* patterns (and so they have also been called *tessellation automata*) in their dynamics. From an artistic standpoint, these figures are reminiscent of Persian carpet designs, ceramic tile mosaics, Peruvian striped fabrics, brick patterns from certain mosques, and the symmetry in Moorish ornamental patterns [57, page 305]. A collection of pictures illustrating pattern formation in evolution of various discrete dynamical systems is documented in [46]. The Mandelbrot set (Figure 3), a frequently studied example in complex analytic dynamical systems, has been called the most *beautiful* object in mathematics [30].

Mathematics teaching should encourage aesthetic moments of mathematics by informing about the proximity of mathematics to beauty, and by being a means of visualizing artistic composition. This requires looking at art according to mathematical laws to discover art as a system of rules. For example, teaching Escher's art (such as shown in Figure 4) can be a motivating introduction to fractals and to dynamical systems exhibiting symmetric chaos, via the concepts of limit, infinity and symmetry. Some exercises (classroom computer experiments) which can be carried out at a pre–university level using computer graphics to illustrate various dynamical systems concepts (iteration, recurrence, attractor, chaos, symmetry, symmetry breaking) have been suggested in [58]. These exercises brings mathematics and art together by drawing connections between dynamical systems concepts

Figure 3: A Computer–Graphic Image of the Mandelbrot set.

and tiling patterns reflected in artistic compositions by Escher and Truchet. In general, learning concepts from chaos and fractals, resulting from nonlinear dynamical systems can help students 'see' and hopefully appreciate the beauty and complexity of various forms which exist within and outside the realm of mathematics. This gives an aestheticized, and thus humanized, representation of complex mathematical phenomena and gives a global and orderly (in form of dynamical systems equations) view of the subject.

3.7 Dynamical Systems and Science

Various branches in science such as physics and chemistry rely on mathematics for rigor and use mathematical results for qualitative and quantitative description of underlying concepts. On the other hand, mathematics teaching seldom refers to science, particularly the experimental component, We therefore ask: How can current scientific research topics and methodologies possibly be introduced into a pre–university curriculum?

Dynamical systems has its origins in problems that arise in the domain of science. This gives an opportunity for integrating science and mathematics via dynamical systems–related experiments.

As an example, consider fractal growth phenomena [72], which has found numerous applications in physical and biological sciences with wide ranging diversity from galactic distributions to sequencing of nucleotides in DNA. Constructing models of these growths can be carried out by scientific and computer experimentation. Using the notion of fractal dimension[3], we can characterize the patterns. An experiment that can be carried out in a simple setting is the *investigation of the fractal dimension of electrodeposition patterns.* See

[3]The dimension D is an exponent that relates the characteristic dimension L of an object to the amount

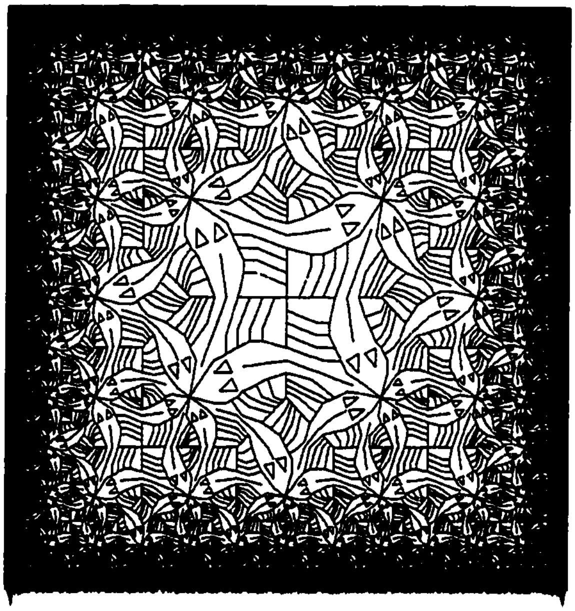


Figure 4: An Escher Composition with Self-Similarity, Symmetry, More?

On Growth & Form: Learning Probability Concepts By "Doing Science" available on the WWW at the URL

`http://polymer.bu.edu/~trunfio/cps-ogaf.html.`

The mathematical object being explored is a (fractal) set and the mathematical operation is measurement (of the dimension). Students watch the pattern develop while taking data on the current and overall size of the pattern. An interactive aggregation kit program displays a computer model of the growth of the aggregation pattern. When electrodeposition is complete, the pattern can be scanned directly as a computer graphic image file into a fractal dimension software for analyzing the dimension of the image. Students can vary experimental conditions and parameters to explore the variety of structures that result. Computers play a central role in this experimentation by carrying out the data analysis in the science classroom. They enhance independent investigations by students and by student groups, and help the teacher support different groups. In this sequence, the essential features of modeling, exploration, and development of intuition come *before* formal mathematics which arises naturally as a *necessity*.

---

of "substance" S it "contains", such as length, area, or volume, and is given by:

$$S = CL^D,$$

where C is a constant that depends on the shape of the object but not on its dimension. Taking the logarithm of both sides of this equation, we get: $\log S = D \log L + \log C$. This is the form of a straight line $y = mx + b$, with m (or D) as the slope, accessible at a high school level. Then the *slope* of the log–log plot gives the dimension of the object under study. For many fractals, the dimension D is *not* an integer.

3.8 Problems in Mathematics Education: Solutions via Dynamical Systems Education

3.8.1 Is Mathematics 'Living'?

Mathematics is a *living* subject with a *spirit* of discovery, inquiry, application and development which seeks to understand patterns that permeate both the world around us and within us. But one comes across various remarks made in the context of mathematics education that might suggest otherwise. In outlining the inadequacies and deficiencies in undergraduate education, it is stated by the National Research Council [12, page 17] that "the way mathematics is taught [...] has changed little over the past 300 years [...]" or as Peterson [55] remarks: "Much of the mathematics encountered by high school and college students seems carved in stone, passed on unchanged from one generation to another. Missing is the sense of how mathematics has evolved since its origins in the distant past and how new mathematics is constantly being discovered and created." There is often a lack of communication of the *vitality* of contemporary mathematics to students.

Even though mathematics is a cumulative science, we should strive for every opportunity to give the students a glimpse of what is new and exciting in the subject. As one of its proposed transitions, it has been recommended by the National Research Council [11, pages 82–84] that the teaching should provide evidence that mathematics is alive and exciting. Teaching should emphasize the study of mathematics as an exploratory, dynamic, evolving discipline — this will encourage students to see mathematics as a science and to recognize that mathematics is really about patterns.

Teaching basic notions from dynamical systems can provide such an opportunity. Computers can play a vital role in this reinvigoration. Topics such as simple iteration of nonlinear functions, when combined with the computer (with graphic capabilities) as an experimental tool, can yield fascinating insights into problems and ideas in contemporary mathematics, thus giving an exhilarating mathematical experience to students. Using these tools they can quickly reach the forefront of knowledge in these fields. To a pre–university student, who has always encountered centuries–old mathematics, it can come quite a surprise that even 'simple–looking' functions such as the quadratic function, when analyzed from a dynamical systems viewpoint, is not completely understood. Learning that much of the behaviour that they see in the bifurcation diagram of the quadratic family was observed for the first time just 20 years ago can be quite intriguing to students. When coupled with the natural beauty of this subject, this knowledge can give students a very different — alive and active — view of mathematics as a scientific discipline.

3.8.2 The Compartmentalization Problem

It has been expressed that a major source of many problems being faced in mathematics education is the *compartmentalization* of the mathematics education system [70] and a *lack of communication* between the compartments [71]. Teaching dynamical systems at an early stage such as in the high school/college, may provide a solution. There is an ample opportunity for and high possibility of new discoveries and open questions even at a pre–university level. High school/college teachers and university professors can communicate and collaborate, which can result in mutual benefits. The WWW provides a timely medium to do that. In Chapter 11, we shall explore this in detail.

3.8.3 'Mathematical-Stratification' and the 'Layer-Approach' in Teaching

> *The mathematician should not be for the physicist a mere purveyor of formula; there should be between them a more intimate collaboration. Mathematical physics and analysis are not merely adjacent powers, maintaining good neighbourly relations; they mutually interpenetrate and their spirit is the same.*
>
> — Henri Poincaré

The beginning of this century saw a 'polyfurcation' among the sciences, particularly mathematics, resulting in another kind of compartmentalization — emergence of many different branches in mathematics (e.g., number theory and ergodic theory) as independent disciplines. These fields behaved like compartments and drifted apart over the time with lack of communication. Also, once these fields were independently established, the response of the mathematical community toward 'interdisciplinary' endeavours has been conservative in the past. The mathematical community has suffered from this subdivision.

Recent work in nonlinear dynamical systems has resulted in a convergence of interest and brought many fields in mathematics together. The applicability of the ideas of nonlinear dynamics to real–world situations have shown that the gap between mathematics and science is artificial. Holmes [35] states:

> Because of the so called 'Chaos Theory', [...] there is a great ferment of excitement and activity. The artificial distinction between pure and applied mathematics is weakening. Mathematics and scientists from different fields are talking to one other. Some are even listening.

'Mathematical–stratification' has had its impact on mathematics education. Often topics in mathematics are taught without exhibiting or emphasizing any interconnections with other areas resulting in a stratified view, lacking a global view of the subject. The 'usual' or traditional way of teaching mathematics has been via a 'layer–approach' — often one course is taught as a prerequisite for a future course which takes away any element of surprise. As Steen [67] expresses: "The layer–approach to mathematics education effectively prevents informal development of intuition along the multiple roots of mathematics ... and making the study of mathematics largely an exercise in delayed gratification."

As a solution to the above problem, the National Research Council [11, page 83] suggests that the teaching perspective should shift from this layer–approach to teaching more diversified curriculum with topics that are relevant to students' present and future needs. One suggestion is of model–building of complex situations. Dynamical systems belongs to that domain. Teaching dynamical systems offers a view which is different than the above tradition due to its inherent interdisciplinary nature. Students can be motivated during teaching because of their ability to experiment and to discover patterns they did not expect to find. This also motivates them to learn the mathematics *necessary* to continue the investigation.

Chapter 4

Dynamical Systems Teaching and Learning Environments: Surveys

In order to evaluate current dynamical systems teaching and learning environments, two surveys — one on teaching and one on learning of dynamical systems — were conducted during the period of January to December 1996 at all the universities in Montreal:

The survey on teaching was distributed among faculty members in Mathematics and other Departments, which have been involved in dynamical systems education.

The survey on learning was distributed among students who had prior experience with dynamical systems–related courses ranging from undergraduate to graduate level. Students came from diverse backgrounds — Mathematics, Physics, Engineering and Biology.

The analysis of responses presented here is based on 10 responses to the survey on teaching and 25 responses to the survey on learning.

In the following, we present the surveys in the format of: Query (Q), Response (R), and Conclusion (C).

4.1 A Survey on Teaching of Dynamical Systems with an Analysis of Responses

The purpose of this questionaire was to survey

- current practices (tools, techniques, methods being employed) in teaching, and
- problems (pedagogical, bureaucratic) that instructors face during teaching.

The goal was to identify, analyse and suggest possibilities that might lead to an improvement of present teaching environments.

Teaching Experience

Q. *How long have you been teaching course(s) related to dynamical systems?*

□ less than 3 years □ 3 or more years.

R. All the faculty surveyed did have 3 or more years of experience teaching dynamical systems–related courses.
C. Teaching experience of 3 years or over can be considered sufficient to draw any concrete conclusions from responses to this survey.

Bureaucratic Problems
Q. *How has the response of the administration at your institution been towards integrating dynamical systems courses in the curriculum?*
□ encouraging □ neutral □ not encouraging
Did you face any problems? □ Yes □ No
If yes, what kind?
R. Most of the responses were "neutral". Few responses indicated that an effort to have a joint course with other departments were not successful due to their lack of interest.
C. It seems that administrative authorities display a certain conservativeness towards interdisciplinary collaborations in education. Such problems have been pointed out elsewhere as well [76].

Learner Background.
Q. *While teaching a course, did you observe that students found certain concepts difficult to understand?*
□ Yes □ No
If yes, what do you think could have been the reason(s)?
R. All responses to the question were "yes". In general, weak calculus, geometry and analysis background were given as primary reasons for lack of comprehension of the concepts involved on the part of students.
C. It would seem important that teachers ascertain student background prior to the commencement of the course and adjust the syllabus accordingly.

Motivation
Q. *Can you state measures that you took in motivating your students towards a the course?*
R. Some suggestions were: emphasis on applications of concepts, use of real–world examples.
C. It might be useful to refer to the applied context of the topic during teaching.

History
Q. *Did you refer to the historical development of the concept in your teaching?*
□ Yes □ No
R. Mention reasons for long lapses of activity between early work and the present state of nonlinear dynamical systems, Mention historical mistakes such as misapplication of Grassberger–Procaccia algorithm.
C. It could be useful to point out to students difficulties mathematicians faced in bringing the subject of nonlinear dynamical to its current state and caveats in studying the subject. This has motivated Chapter 6.

Mathematical Connections and Dynamical Systems

Q. *Have you introduced dynamical systems concepts in courses other than that of dynamical systems (e.g., calculus, linear algebra, numerical analysis)?*

□ Yes □ No

R. Most responses to the question were "no".

C. It appears that integrating dynamical systems–related concepts in teaching traditional mathematics topics is not widespread. This has motivated Chapters 13 and 14, where certain pedagogical suggestions are presented.

Q. *During teaching a dynamical systems course, did you draw connections of any topic in it to other areas in mathematics (e.g., calculus, differential equations, numerical analysis)?*

□ Yes □ No

R. Most responses to the question were "yes".

Q. *Did you find that this approach enhanced learning and/or interest in the topic?*

□ Yes □ No

R. All the responses to the question were "yes".

C. It can be concluded that drawing connections with previously known concepts in mathematics while teaching a topic in dynamical systems can assist learning and instill interest in the subject among students.

Dynamical Systems Applications

Q. *Were the dynamical systems models you considered during the course motivated by any 'real-world' applications in any way?*

□ Yes □ No

R. Most of the faculty responded "yes".

C. All the faculty from non–Mathematics Departments responded "yes", which would likely be the justification for an application oriented approach. It also suggests that use of applications is not made regularly in dynamical systems courses taught in Mathematics Departments.

Dynamical Systems Modeling

Q. *Did the dynamical systems models in your course correspond to any experimental or observational data?*

□ Yes □ No

R. Similar to the previous query.

C. Similar to the previous query.

Visualization of Dynamical Systems

Q. *Did you employ any visualization equipment during teaching (e.g., slides, film, animation, computers with graphic displays, etc.)?*

□ Yes □ No

R. All responses to the question were "yes".

Q. *Did you find any distinct advantage (e.g., improved understanding, performance, etc.) in using the equipment?* □ Yes □ No

R. All responses to the question were "yes".

Q. *Did you notice any drawbacks (e.g., misconceptions owing to optical illusions)?*
□ Yes □ No

R. Most of the responses to the question were "no". Drawbacks suggested were: such equipment is less interactive than lecturing and that this problem could be overcome by appropriate software; curves are often projections which can be misleading.

C. The responses mostly indicated the use of colour slides and computers with graphic displays. They also indicated that the presence of 'colour' was particularly useful. It would seem that reference to pitfalls in using computer–graphical approach should be made at appropriate places to students.

Computation in Dynamical Systems

Q. *Was computer programming as integral part of any of your course(s)?*
□ Yes □ No

R. Most responses to the question were "yes".

Q. *Did you use any of the available dynamical systems softwares in your course?*
□ Yes □ No

R. Most responses to the question were "yes".

Q. *If yes, did you experience any disadvantages of their usage?*
□ Yes □ No

R. Most responses indicated that it is imperative to use computers when teaching dynamical systems concepts, encouraging students to experiment on computer and mentioned that there is not sufficient time to do them in class.

C. Using computation can be useful in teaching dynamical systems. However, to do that time constraints need to be overcome. In Chapter 8, we pursue this in detail.

Communication Technology

Q. *Did you use any means of communication technology (e.g., e-mail, 'local' newgroups, bulletin boards on the computer network etc.) in your course(s)?*
□ Yes □ No

R. Most responses to the question were "no".

C. It can be concluded that the use of such technology is not widespread in instruction.

WWW

Q. *Did you make any use of WWW technology in your course(s)?*
□ Yes □ No

Do you see any advantages of such technology in dynamical systems education?

R. Most responses to the question were "no".

The advantages mentioned were: convenient way of accessing vast pool of information, which can be quite helpful in student in their projects, the multimedia–nature of the Web

can be a motivating experience.
C. It can be concluded that the use of such technology is not widespread in instruction. It seems that the use of WWW in instruction is in its infancy stage.

An Early Introduction to Dynamical Systems
Q. *Do you recommend introducing a dynamical systems course in pre-university curriculum?*
□ Yes □ No
If yes, why do you think there is a need?
R. Responses strongly suggested that some basic dynamical systems could be a part of a calculus course, mathematical modelling could be introduced at an accessible level, dynamical systems' viewpoint is often more natural than the classical approach to certain concepts say in calculus. It provides background for analysis.
C. Chapter 13 was motivated by these responses.

4.2 A Survey on Learning of Dynamical Systems with an Analysis of Responses

The purpose of this questionaire was to survey

- student response (such as suggestions for improvement) to current approaches being employed in teaching, and
- problems (cognitive/epistemological) that students face during learning.

The goal was to identify the deficiencies and suggest possibilities that might lead to an improvement of present learning environments.

Experience
Q. *How many dynamical systems related course(s) have you taken (including the current ones)?*
□ less than 3 □ 3 or more?
R. Among all the responses received 55% had the experience of 3 or more courses and 45% has the experience of less than 3 courses.
C. Students with experience of less than 3 courses can be considered "beginners", while those with 3 or more courses as "advanced". This helped identify problems faced by students at different levels.

Epistemological Inquiry
Q. *Among the topics you studied, which ones were difficult to understand? Which ones were easy?*
R. The responses reflected variety of topics with repeatition. Topics which were found hard were symbolic dynamics, notion of bifurcation, notion of chaos. Topics found easy

were discrete dynamical systems, notion of fixed points and periodic orbits. Iteration of functions was found easy by some and difficult by others.
C. The last response was one of the motivations for development of lesson plans of Chapter 14.

Motivation in Introduction of a Concept
Q. *Is it always clear to you why a certain concept is introduced? For example, can you tell what made the concepts of fixed point, iteration, etc., necessary in the development of the theory?*
R. Most of the responses implied that the reason why a certain concept was introduced was that it was necessary for the development of further theoretical background. Some responses indicated the reason as a result of the emergence of a *pattern* during the study.
C. We have argued against this stratified view in Chapter 3.

Mathematics is Living
Q. *Are you aware of the fact that lot of important discoveries in nonlinear dynamical systems are relatively new? Has this influenced your view of mathematics that you have encountered during your years of study?*
R. Most of the responses were "yes". Responses indicated that it helped change view of the subject from linear to nonlinear.
C. This has been one of the motivations behind this thesis. In Chapters 13 and 14, we discuss ways of introducing novelty in mathematics through teaching topics from dynamical systems.

Mathematical Connections and Dynamical Systems
Q. *Are you familiar with the fact that various concepts from dynamical systems theory occur in many classical fields in Mathematics?*
□ Yes □ No
R. All responses were "yes".
Q. *Has this influenced your view of concepts in* those *fields in any way?*
□ Yes □ No
R. Most responses were "yes". Some specific remarks were: "It helped me see the old concepts in a new way qualitatively; I could see a broader picture of mathematics as a whole; It makes me realize that Mathematics is everywhere; It has given me a whole new meaning of numerical methods such as Newton's method; Mathematics and Physics go hand in hand; I found topology was useful after all while describing the structure of an attractor of a nonlinear ODE; Calculus was never the same."
C. It indicates that introduction of dynamical systems can help students view previously known traditional topics in a novel way. One of the objectives of the thesis is to formalize a methodology which creates an environment to realize that. Chapter 13 discusses this issue in detail.

Motivation for taking a Course

Q. *What were the reasons that motivated you to take the course(s)?*

R. Some specific remarks were: "It seemed that dynamical systems was practical and somewhat less dry aspect of analysis; Applications of dynamical systems in other fields instigated curiosity to learn about their foundations; Reading popularizations of chaos and fractals made me curious; Universality of dynamical systems concepts was appealing."

C. It seems that the "applied" aspect of dynamical systems can be a useful motivation for students to take and continue a course.

Visualization in Dynamical Systems

Q. *Was any visualization equipment used in your courses (e.g., slides, film/animation, computer-graphic displays, etc.)?*

□ Yes □ No

Did you find any specific advantages (e.g., improved understanding, stimulated interest in the course, lead to a new discovery, etc.) over algebraic explanations, in using the equipment?

□ Yes □ No

R. All responses were "yes". Some specific remarks were: Computer displays were extremely helpful not only in stimulating interest but also in understanding ideas; Increased motivation.

C. This motivated Chapter 9.

Computation of Dynamical Systems

Q. *Did you carry out any programming tasks as part any of your course(s) for numeric, symbolic and/or graphical analysis of a dynamical system?*

□ Yes □ No

If yes, can you indicate any available programming environments/softwares for analyzing dynamical systems? Any particular recommendations?

R. Most responses were "yes". Programming languages used were: FORTRAN, C. Softwares listed were: Maple, Mathematica, AUTO, dstool, PHASER. Use of Maple and PHASER was recommended.

C. We have introduced possible exercises with PHASER in Chapter 14.

Q. *Did you find any specific advantages of the computational approach over other approaches in learning a concept? In what way?*

□ Yes □ No

R. Most responses were "yes".

Q. *Did you find any difficulties in usage of any of the above (e.g., absence of a good user-interface, graphics capabilities, etc.)?*

□ Yes □ No

R. As disadvantages, it was remarked that there is lack of good and organized computer training for students in the field, some packages assume that user has a knowledge of a certain language.

C. Throughout this thesis, we have emphasized introduction of necessary prerequisites for using any tools during instruction. Importance of a good user–interface that can minimize "computer problems" and lets the learner focus on the application, is also made in Chapters 7 and 8.

Communication Technology

Q. *Do you think that modern communication facilities have influenced the field of dynamical systems in any way?*

□ Yes □ No

R. Responses were: Information is easily and readily accessible; large computations can be carried in short time; using e–mail, discussion can be carried out at a global level.

C. One of the objectives of this thesis is to illustrate the utility of such technology in dynamical system education, and is discussed in Chapter 11.

WWW

Q. *Did you use World–Wide Web for any dynamical systems related study? Do you see any specific advantages? Did you see any tools that might be lacking?*

R. The responses were: Use of Mosaic, Netscape for searching information related to dynamical systems. Advantages mentioned were: Fast and easy access to multimedia information. Problems mentioned were: Easy to get lost while 'browsing'; information overload can be overwhelming; quality of information retrieved is not always relevant and so improvement in methods for accessing *really* needed information is necessary.

C. It seems that the use of WWW in dynamical systems education holds promise though it is still in its state of evolution. To provide a sound basis for use the WWW in dynamical systems education is one of the primary objectives of this thesis.

Usenet

Q. *Are you aware of and/or do you participate in any Usenet newsgroups related to dynamical systems?*

□ Yes □ No

R. Most responses admitted awareness and all denied any participation.

C. In general students seem unfamiliar with the use of such facilities. Chapter 11 discusses the use of Usenet in dynamical systems education.

Part II

Dynamical Systems Educational Environment on the WWW: Design and Development

Chapter 5

Hardware and Software Set–Up

In creating a WWW–based educational environment, there are many hardware and software considerations. In this Chapter, we discuss these issues.

5.1 Intranet vs. Internet in Education

The *Intranet Design Magazine* available on the WWW at the URL

`http://www.innergy.com/ifaq.html`

defines an Intranet as:

> In'tra net - n. 1) a network connecting an affiliated set of clients using standard internet protocols, esp. TCP/IP and HTTP. 2) an IP–based network of nodes behind a firewall, or behind several firewalls connected by secure, possibly virtual, networks.

An intranet is essentially an Internet *within* an organization. These type of computer networks are rapidly growing in the business world but are still in their infancy in educational institutions.

5.1.1 Advantages of an Intranet

Why do we need an intranet? What are the benefits? There are several advantages (which have been behind the success of the Internet) in creating an educational Intranet:

- **Freedom of Choice.** WWW technology is based on open standards and is available for nearly all widely used hardware platforms and operating systems. It therefore doesn't compel institutions into limited, proprietary choices.
- **Ease–of–Use.** With Intranet clients, such as Netscape Navigator, a *single* front–end can be used to access all internal and external resources; users don't need to learn multiple software packages.

- **Cost–Effectiveness.** Intranet tools are inexpensive in initial purchase and deployment. The Intranet's platform independence eliminates the need to create different versions of the same applications. Setting–up an Intranet can save resources, for example, the student calendars can be made available in an electronic form on the Intranet, instead of printing them.

- **Reliability.** Internet technology is proven, highly robust and reliable.

- **Standards.** The adoption of standard protocols and APIs allows infrastructures to be built and managed to meet *changing* technological needs in an educational environment without substantial overhead. As an example, educational institutions typically have a variety of clients, including Macintoshes, PCs, UNIX systems, and even mainframes. By implementing Intranet technologies, the problems with sharing data between systems become nonexistent — TCP/IP can become the common backbone protocol and HTTP can be the data delivery mechanism protocol.

Should an existing WWW site be used or a local customized site be created, and if so what are its advantages? There are various specific advantages of setting–up an educational Intranet over using the Internet:

- **Information Access.** By making available necessary information locally gives independency. External sites may go out–of–date or change address, or when access is really needed, their server may be down. By setting–up and maintaining an internal WWW site, one can be assured of up–to–date information.

- **Performance.** Speed has always been one of the limitations of the WWW to the extent that it has been called as "World–Wide Wait". Accessing external servers can be very slow at the time when performance is really needed, such as during a class or seminar demonstration. An Intranet resolves this problem to a large extent.

- **Security.** Protecting information, even within an educational network, is critical. Intranets are protected by a *firewall*, a network configuration usually created by hardware and software, that forms a boundary between networked computers within the firewall from those outside the firewall. This enables institutions to make only necessary information available to the outside world. The issue of WWW security is discussed in detail in Section 5.6.

- **Learning.** By setting–up an environment on a smaller scale (Intranet), gives the students an opportunity to develop their skills and protocol to communicate on a larger scale (Internet).

One instance of an Intranet could be within a Department of an educational institution as shown in Figure 5. Intranet applications within an educational institution have been discussed on the WWW at URL

`http://indy.cs.concordia.ca/www/intranet/intranet_edu.shtml`.

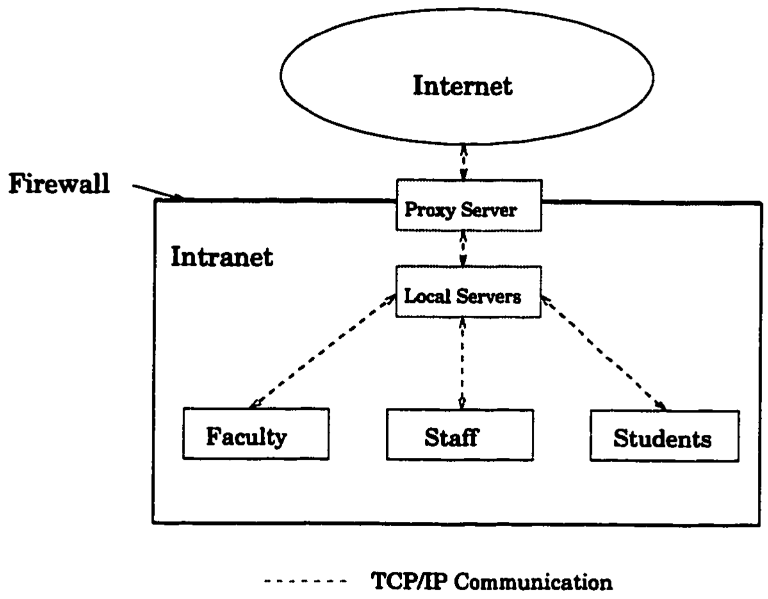


Figure 5: Structure of a Departmental Intranet.

5.2 Hardware/Software Requirements

What kind of hardware/software is needed? The physical infrastructure (cabling, routers and servers); the computer network operating software supporting TCP/IP; DNS and IP addressing; Mail (SMTP/POP); News (NNTP); WWW (HTTP) server; WWW browser with browser support applications; WWW tools for helping set-up and maintaining WWW sites (specific software tools needed are pointed out in future chapters).

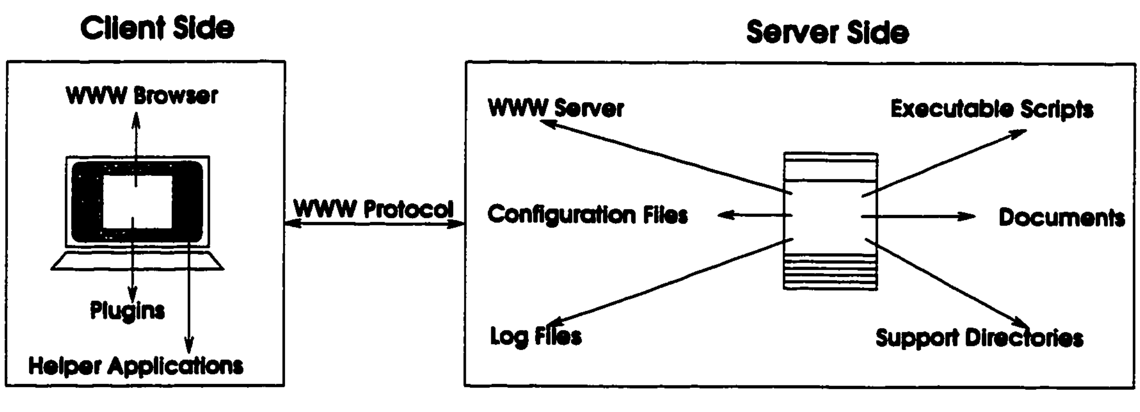


Figure 6: The Client–Server Software Environment for Dynamical Systems Education on the WWW.

5.2.1 Cost

The platform of computers can be chosen among any of the ones broadly used, though it is important to choose an operating system which supports a wide variety of WWW-specific softwares. Unless stated otherwise, we shall refer to UNIX-based WWW software.

Computers need to be connected on a local area network (LAN) with an infrastructure which supports a high-speed data communication (e.g., ethernet). The LAN itself should also have a high-speed connection to the Internet (e.g., T1). This is often already in place in many Departments. Otherwise it need to be purchased, accounting for most of the (monetary) expense. Most of the Intranet software is either freeware/shareware or available at an moderate educational price.

Time is the second biggest expense. The expertise of a system and network administrator, WWW server administrator, WWW script developer, WWW document designer and developer is required. Sites would need regular maintenance. This process would initially be time consuming, but with various long-term potential gains.

5.3 WWW Server Set-Up

Choosing a Server

The server is chosen according to the operating system of the computer on which the WWW server is to reside. The following characteristics need to be taken into consideration: speed, size, security, properties/facilities such as CGI support. Based on these features, from the *Netcraft Web Server Survey* on the WWW at the URL

`http://www.netcraft.co.uk/survey/`,

Apache is the leading WWW server for UNIX-based platforms.

Installing and Configuring the Server

Most of the WWW servers are available in an executable format; otherwise they need to be compiled. According to the services one wishes to provide, appropriate directives in server configuration files need to be changed (from their default settings). This includes setting the path for document root directory, enabling CGI scripts, clickable impagemaps and server-side includes. In [43, Chapter 1], [68, page 68], steps involved in installation and configuration of an Apache server are described in detail.

5.4 WWW Browser Set-Up

WWW is an environment which is based on freedom of choice and independence of the underlying hardware and software. Even so, it is difficult to ignore any significant improvements in the field of browser development.

Choosing a Browser

According to recent surveys, the following browsers are most in use: Netscape Navigator and Microsoft Internet Explorer among graphically-based browsers and Lynx (University of Kansas) among text-based browsers.

In this section, we will discuss only Netscape Navigator and Lynx. Tutorials for both Netscape Navigator and Lynx have been developed and are available on the WWW at the URL

`http://indy.cs.concordia.ca/www/browsers/.`

Installing and Configuring the Browser

Browsers, particularly commercial such as Netscape Navigator and Microsoft Internet Explorer, are available only in executable formats for a variety of platforms. Otherwise, they need to be compiled for the appropriate operating system (such as Lynx).

5.4.1 Netscape Navigator

Configuring Netscape Navigator

Most of the necessary configurations relevant in our case can be done via the [Options] menu on the menu bar. The default 'home page' URL, TELNET/rlogin clients and plug-ins/helper applications can be set-up through the submenu [General Preferences], mail and news servers can be set-up through the submenu [Mail and News Preferences], and security-related options (showing alert before accepting a cookie, enabling Java and JavaScript, showing alert before submitting an HTML form, password to use the browser) can be set-up through the submenus [Network Preferences] and [Security Preferences].

The educational advantages presented by Netscape Navigator can be divided into the following categories: communication, multimedia support, client-computing platform, management and security,

- **Communication.** Netscape Navigator has built-in facility for sending/receiving e-mail messages and browsing Usenet newsgroups. Video-conferencing plug-in support turns the computer into a real-time Internet-based phone. Chat tool and shared whiteboard allow easy data collaboration for global communication.

- **Multimedia Support.** Navigator extends its ability to present dynamic interactive documents by integrating a VRML viewer and tools for video and audio playback in the browser. Plug-ins, Java applets, JavaScript and HTML all communicate and interact seamlessly. Rich multimedia applications can be build by linking these technologies together.

- **HTML Capabilities.** In addition to supporting most of current HTML standard, Navigator includes support for additional HTML layout capabilities such as frames, multiple text columns, control over vertical and horizontal whitespace, table background colors and font face styles supported by the operating system. The Gold version of Navigator also includes a WYSIWYG editor.

- **Client Computing Platform.** Navigator provides possibilities for client-side computing by support for Java applets and JavaScript scripts.

- **Management and Security.** Navigator allows for customizability through the

Netscape Administration Kit which can be used to set and lock common user preferences, personalize buttons and menus.

5.4.2 Lynx

Configuring Lynx
Most of the necessary configurations relevant in our case can be done via the configuration file `lynx.cfg` (setting default 'home page' URL, address of Lynx help file) and Lynx Options menu (setting the editor, such as vi or Emacs; searching type, case sensitive/insensitive; user mode, which defines the level of the user).

Some advantages of a text-based browser such as Lynx are:

- **Efficiency.** Since there is no delay in downloading multimedia files, it speeds-up the downloading process.
- **Editing.** Text-based editors such as vi or Emacs can be invoked from within a Lynx session.

5.5 WWW Browser–Support Applications

There are various file formats not supported by (any of the above mentioned) browsers. These include many of mathematically significant ones such as Maple (`*.m`) and PostScript (`*.ps`). To invoke such files on the client-side, use of plug-ins or helper applications can be made[1]. Plug-ins and helper applications can be considered as complement to each other. They can be configured in the Helpers dialog box found under General Preferences submenu in the Options.

5.5.1 Plug-Ins

Plug-ins are dynamic code modules, native to a specific platform on which Netscape Navigator runs. The primary goal of the plug-in API is to allow existing platform dependent code to seamlessly integrate with and enhance Navigator's core functionality by providing support for new data types. The Netscape Plug-in API allows third party developers to create a 'viewer' of the information inside the Navigator window and called via the HTML tag <EMBED attributes> ... </EMBED>. This gives the users, for example, the ability to play audio samples or view video movies from within Navigator.

Installation and Configuration
Once downloaded for the appropriate architecture, a plug-in is installed by placing it in the `plugins` subdirectory in the Navigator folder, for example, `.netscape` in UNIX. The file format and the corresponding plug-in should be indicated in the MIME types file (`.mime.types` in UNIX). When the Navigator starts up, it checks for plug-in modules in the `plugins` subdirectory in the same folder or directory as the Navigator application.

[1] We can also use ActiveX programs. We shall not discuss the ActiveX technology as, among the graphical browsers, it is at present only supported by Internet Explorer.

Embedded plug-ins are loaded by Navigator when the user encounters an HTML page with an embedded object with a MIME type registered by a plug-in.

5.5.2 Helper Applications

Helper applications start external programs or display files with formats *not* supported by the browser. Helper applications are not integrated into WWW pages and don't display inside the browser window. For example, when Netscape Navigator encounters a sound or video file on the Internet, it passes on the data to the appropriate helper applications, to run or display the file.
Installation and Configuration
In contrast to a plug-in, a helper application corresponding to a file format usually already exists. If not, it need to be installed like any other software on the system. However, path to invoking the helper application need to be set, which is usually the user's home directory. Reference to required plug-ins/helper applications is made at appropriate places.

5.6 WWW Security

Apart from the general subject of computer security, WWW security requires special attention. WWW client/server-side software have security flaws. Usually, the students (and even teachers) using WWW-based services are unaware of its potential dangers. WWW structures can be altered, software (e.g., WWW server or browser) crash, important files can be removed, service can be disrupted, private information (such as password) can become public. The current generation of software calls upon users to make security relevant decisions yet they are not given sufficient information to make informed choices. We need to be aware of issues of WWW security since it is considerably more expensive and more time consuming to recover form a security accident than to take preventative measures ahead of time. Wee need to ask:

- What type of security problems exist and how can they affect a teacher/student?
- What are the origins of such problems and what are their solutions?

WWW security problem consists of three parts:

1. Securing the WWW server and the data on it.
2. Securing the user's own computer.
3. Securing information that travels between the WWW server and the user.

In this section, we shall concern ourselves only with the first two.

5.6.1 Server–Side Security

WWW security starts with host security, i.e., the security of the computer on which the WWW server is running. Security is defined by policy whose role is to guide the users (students) in knowing what is allowed and WWW content providers (teachers) in making choices about system configuration and use. Such a guideline has been given in [25, page 256].

This is important if one provides an online examination along with its solution in the same directory.

WWW Server Security Problems/Solutions

Second, the WWW server must be secured by (a) minimizing the number of services provided by the host on which the WWW server is running (b) restricting access to the WWW server (locating the server in a secure facility, limiting the number of users who have the ability to log into the computer). The easiest way to restrict access to information and services is by hiding their URLs, i.e., by storing files in hidden locations. For examples, when using the NCSA or Apache servers, we can have an `index.html` file in each directory which serves documents to hide the contents of that directory. The problem with this 'hidden URL' approach is that anybody who knows the URL's location has full access to the information it contains.

We can use the server's configuration directive (such as <Limit> directive in NCSA or Apache servers) to control which files are accessible and by whom. This can be done in two ways:

- One can place restrictions to a directory (and all of its subdirectories) by placing an `.htaccess` file in that directory.
- One can place all the access control restrictions in a single file — `access.conf` (NCSA server)/`httpd.conf` (Apache server).

Most WWW servers allow us to restrict access to particular directories to specific computers on the Internet. We can specify these computers by their IP addresses or by their DNS hostnames.

One can implement host-based restrictions using a firewall to block incoming HTTP connections to particular WWW servers that should only be used by authorized users. Such a network is illustrated in Figure 5. Various examples of firewall topologies are given in [68].

However, firewalls can give the misleading impression of total security and can be are frequently misused. Often, security problems are internal rather than external. Thus, it has been suggested [25] that a firewall should be used only to gain additional security that works in conjunction with additional internal controls and not as a replacement of them.

Another approach is identity–based access, based on username/password which authenticates the person. One advantage of user–based access control over host–based access control is that the authorized user can access the WWW server from anywhere on the Internet. This is useful since many students may be using the WWW through an Internet provider rather than through the educational institution, in which case it is very difficult to keep track of all the hostnames.

CGI/API Security Problems/Solutions

WWW servers are extensible. This extensibility makes it possible to connect to databases and to have other programs running on the network. If not properly implemented, modules that are added to the WWW server can compromise the security of the entire system.

CGI was the first and remains the most popular means of extending WWW servers. CGI programs run as subtasks of the WWW server; arguments are supplied in environment variables and to the program's standard input; results are returned on the program's standard output. Another way to extend WWW servers is by using a proprietary APIs. APIs are faster to interface custom programs to the WWW servers as, unlike CGI, they do not require a new process be started for each WWW interaction. Instead, the WWW server process itself runs application code within its own address space that is invoked through a documented interface.

As a result of their power, the CGI and API interfaces can compromise the security of a WWW server and the host on which they are running. The following should therefore be taken into consideration in their use:

- The programs should be designed to ensure that they can perform *only* the desired functions. A guideline of general principles by which to code, including specific rules for specific programming languages, is given in [25, page 300].
- The programs should be run in a restricted environment to limit the damage in case of an unexpected behaviour. Such a restriction is possible in a UNIX environment; unfortunately, Windows95 and Macintosh operating systems do not have the notion of restricted users.

5.6.2 Client–Side Security

Using a WWW browser can pose a variety of problems. A WWW browser is extensible using technologies such as ActiveX, Java, JavaScript, VBScript, helper applications and plug-ins. These technologies can also be subverted and employed against the browser's user.

Browser–Related Problems

WWW Browsers have bugs, many of which are data dependent. Such bugs can lead to browser failure such as a crash. On a computer without memory protection, a browser crash can take down the entire computer, creating an effective denial-of-service attack[2].

Java Security Problems/Solutions

Java applets along with their ability of providing a variety of client-side applications, pose serious security threats, particularly if used over the network. In 1996, it was discovered in Netscape Navigator that a Java applet could open connections to any Internet host,

[2] A denial-of-service attack is an attack in which a user (or a program) takes up so much of a shared resource that none of the resource is left for other users (or uses).

potentially allowing applets running behind firewalls to attack other computers behind the firewall.

Netscape Navigator 3.0 now allows the following to ensure Java security:

- Java can be disabled from the list of user preferences.
- Java applets that are downloaded from the Internet are restricted in a number of ways. This allows not being allowed to access the local file system, and being allowed to create network connections to the computer from which they were downloaded.
- Java applets that are loaded from the user's local hard disk have full access to all features of the language.

Netscape Navigator 4.0 opens the Java sandbox (Figure 2), allowing downloaded applets more functionality and flexibility. It identifies a variety of different kinds of privileges that a Java program might need, which are given to the program on a case–by–case basis.

JavaScript Security Problems/Solutions

JavaScript programs are inherently more secure than programs written in Java since there are no "methods" for directly accessing the client computer's file system or for directly opening connections to other computers on the network. Inspite of this, security problems have occurred with JavaScript in two main areas: denial–of–service attacks and privacy violations.

Netscape Navigator's JavaScript implementation suffers from the fact that one can not break out of a running JavaScript program. The only way to terminate a program is to exit the browser itself, which sometimes may be possible only by killing the window system (such as using `CTRL-ALT-DEL` under Windows95).

Since a JavaScript program runs inside a browser itself, it potentially has access to any information that the browser has. This creates the possibility of privacy violations, such as inappropriate use of cookies.

Helper Applications Security Problems

Most WWW browsers can only understand a small predefined set of data types. There are many kind of data types that can not be readily translated into HTML or images.

One way to extend the browser is through the use of helper applications. However, they can also create security problems. A helper application is run on the user's own computer and takes input from information provided by the WWW server. If the helper application has sufficiently powerful features, it could be used against the user's interests. A list of such helper risky applications is given in [25, page 31].

Plug–ins/ActiveX Security Problems

Plug–ins were introduced by Netscape as an alternative to helper applications and a means of extending Netscape Navigator with executable programs that are written by third parties and loaded directly into the browser. ActiveX is a Microsoft proprietary technology for downloading executable code over the Internet. ActiveX controls are plug–ins that are automatically downloaded and installed in the browser (for now only in Internet Explorer).

The security problems inherent with the use of plug-ins/ActiveX controls are in many ways similar to that of helper applications.

Remedies to client-side security problems involving downloaded code are to rely on trusted vendors and to minimize the privileges available to the execution context in which the downloaded code runs.

5.7 Acceptable Use Policies

Acceptable use policies (AUP) is a documented guide agreed to by teachers, students and parents that governs how the Internet and WWW are to be used by the school and its community. It states acceptable uses of online materials, rules for online behaviour, and access priviledges. It is a good practice for educational institutions to form an AUP which conforms to their standards. The ERIC database available on the WWW at the URL

`http://www.aspensys.com/eric/`

provides guidelines to creating AUPs.

Chapter 6

Dynamical Systems: History on the WWW

The true method of foreseeing the future of mathematics is to study its history and its actual state.

— Henri Poincaré

History is more than a mere collections of facts; it is an organized understanding of how we got to be what we are. Therefore, the history of mathematics can provide some enrichment of the subject and present a view that mathematics is connected with the development of our culture [66]. Dynamical systems has a long and distinguished history as a branch of mathematics. In this Chapter, we discuss the significance of this social aspect in education and how can it be communicated to the students via the WWW.

6.1 Teaching and Learning History of Mathematics

The importance of integrating history of mathematics in teaching has been recognized and promoted recently[1]. As one of the objectives of undergraduate education [12, page 58], it is suggested that "all mathematics students should engage in serious study of the historical context [...] of mathematics".

Teaching the history of a mathematical subject has various useful educational implications:

- **Mathematics is Cumulative.** History is the accumulated experience of mathematicians. It has shaped what we know and is the best source of information as to what is possible, or even impossible. Formation of knowledge can be considered as

[1] An entire issue of *For the Learning of Mathematics : An International Journal of Mathematics Education* (Vol.11, No.2, 1991) has been devoted to this topic.

a iterative process, where present is determined from the past (i.e., history). Piaget stated, "knowledge is never a state but a process, which is influenced by preceding developmental stages" [56]. It is therefore important to mention the history behind the construction of a concept. Also, it is useful to teach students the fields' *empirical* aspect which would reflect the fact that even the fundamental notions in the field were once subject to question and speculation.

- **Motivation.** It can be hoped that a historical approach to teaching would increase motivation in students by showing more 'human' side of the subject — that the mathematicians who are credited for the development of the subject were human beings, who did not come up with solutions to all problems immediately. This can also inspire students to continue the subject.

- **Mathematical Connections.** History shows that there are links between different areas of mathematics. Topics in dynamical systems such as ODEs were invented in a physical context, cellular automata in a computational context and certain discrete dynamical systems (logistic map) in a biological context.

- **Mathematics is Living.** Teaching history can make the students realize that mathematics is not once and for all discovery, but is constantly changing. By ignoring history, individual mathematicians, or cultures, mathematics does not emerge as a human creation and discovery. By introducing the history, students can learn more about the concepts and constructs of the subject: *where* they came from and *why* they are studied. The advantage of knowing history gives them the chance to read the masters of the past and learn lessons from their experience. As they struggle with the subject, and change as a result, they in a sense recapitulate the history of the subject.

6.2 WWW: A Conveyor of History and Culture of Dynamical Systems

Biography

In dynamical systems, as in other fields of mathematics, there are various concepts (and constructs) that bear a mathematician's name: fractal (Mandelbrot), Julia set (Julia), horseshoe (Smale) etc. Also there are mathematicians who are credited in many ways for pursuing and establishing the concept (construct) after it was introduced. While teaching such a concept (construct), a WWW browser can be used to give the students a historical tour in the classroom in real-time by presenting short biographies of those mathematicians who have been part of this heritage. A collection of biographies of mathematicians dated as far back as 16th century B.C. is available on the WWW at the URL

`http://www-groups.dcs.st-and.ac.uk/~history/`.

Pictures of mathematicians and snapshots from their lives can be shown. Difficulties they faced in pursuing the subject and the circumstances they worked in can be pointed out.

Figure 7 shows a short biography of Julia who worked on iteration of complex rational functions in 1920's.

Geography

The part of the world from which these mathematicians came from can also be shown on an *interactive* map (which is clickable imagemap, see Section 9.5.2). Such a facility is available on the WWW at the URL

`http://www-groups.dcs.st-and.ac.uk/~history/SensitiveMap/`.

For example, Figure 8 shows birthplaces of mathematicians in Europe, where many of the early workers in dynamical systems theory came from.

A short history of computation in dynamical systems is available on the WWW at the URL

`http://indy.cs.concordia.ca/ds/history/hcds.shtml`.

Gaston Julia

1893 - 1978
Born Algeria. Died France.

Welcome page	Instructions
Birthplace map	Mathematicians of the day
Short Biographies Index	History Topics Index
Previous (Chronologically) Next	Chronological Index
Previous (Alphabetically) Next	Alphabetical Index

Gaston Julia was one of the forefathers of modern dynamical systems theory. He is best remembered for what is now called the Julia set.

When only 25 when he published his 199 page masterpiece *Mémoire sur l'iteration des fonctions rationelles* which made him famous in the mathematics centres of his days.

As a soldier in the First World War, Julia had been severely wounded in an attack on the French front designed to celebrate the Kaiser's birthday. Many on both sides were wounder including Julia who lost his nose and had to wear a leather strap across his face for the rest of his life. Between several painful operations he carried on his mathematical researches in hospital.

Later he became a distinguished professor at the École Polytechnique in Paris.

In 1918 he published a beautiful paper *Mémoire sur l'itération des fonctions rationnelles* (Journal de Math. Pure et Appl. 8 (1918) 47–245) concerning the iteration of a rational function f. Julia gave a precise description of the set J(f) of those z in C for which the n th iterate $f^n(z)$ stays bounded as n tends to infinity. It received the Grand Prix de l'Académie des Sciences.

Seminars were organised in Berlin in 1925 to study his work and participants included Brauer, Hopf and Reidermeister. H Cremer produced an essay on his work which included the first visualisation of a Julia set.

Although he was famous in the 1920's, his work was essentially forgotten until B Mandelbrot brought it back to prominence in the 1970's through his fundamental experiments.

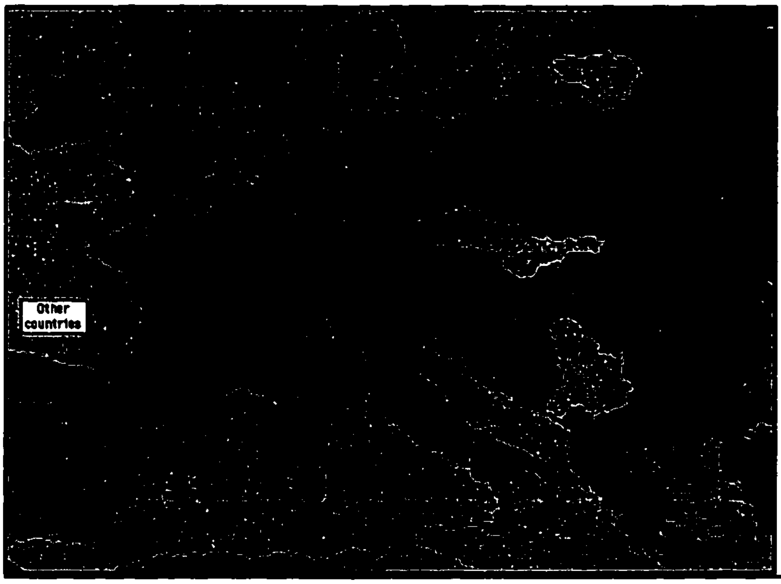
Welcome page	Instructions
Birthplace map	Mathematicians of the day

Figure 7: A Short Biography of Gaston Julia.

Birthplaces of Mathematicians (Europe map)

Click on one of the buttons on the map and you will be told the name of the place and names of the mathematicians in our list of long biographies who were born there. If you click on a country, you will be told its name and given a list of mathematicians born in that country. Both those with long and short biographies are listed.

Places with one mathematician are marked • , whilst
places with more than one mathematician are marked • .



Welcome page — Instructions
Chronological Index (Long Biographies) Alphabetical Index
Short Biographies Index — History Topics Index

JOC/BS May 95

Figure 8: Birthplaces of Mathematicians in Europe.

Chapter 7

Dynamical Systems: Documentation on the WWW

WWW creates a multimedia and hypermedia-based electronic publishing environment. It provides solutions to current problems in publishing (electronic or otherwise).

7.1 Advantages of WWW-Based Publishing

Specifically, the advantages of the WWW towards developing dynamical systems-related documentation are:

- **User.** Hypertext supports reading and studying in different ways and on different levels depending on the user.
- **Teaching and Learning.** Hyperlinks in a WWW document facilitate explanation of concepts related to other sections in the course by *immediate reference* within or outside the document. This brings clarity to the lecture by not referring back-and-forth to transparencies (if they are being used). It also helps student recall concepts already learnt, access prerequisites for a topic in a natural manner, not possible in a plain-text environment.
- **Content.** The WWW, through multimedia, creates an open environment for publishing information in various formats that the information can be represented. Paper publishing is insufficient — it can not represent sound, 3D-computer graphics, animations, which is possible in a multimedia document.
- **Portability.** Often, documents published in a traditional electronic format (such as Microsoft Word) lack portability. On the other hand, as HTML is platform-independent, with a WWW browser, the documents can be studied in different environments (e.g., UNIX, PC and Macintosh).

- **Storage.** Even if information is easily accessible, there are restrictions to disk space when storing large amounts of data, such as graphic files. With the WWW hypermedia support, documents can have HTTP connections to other sites where such data can reside at remote repositories and can be *shared.* This also avoids unnecessary duplication of data.

- **Cost.** WWW provides an inexpensive medium to publish and transfer course material to different sites. Textbooks and CD-ROMs are usually expensive. Photocopying and printing large amounts of course material (transparencies), lecture notes) also tends to becoooome expensive. WWW publishing is almost free or cheap, and thus affordable to students. Since a single WWW document can be accessed by many, there is no need of publishing multiple copies of the same document which saves natural resources (such as paper).

- **Maintenance.** When the nature of information is static, once published, the content can not be modified for maintenance (for updates, etc.). Due to the object-oriented design of the WWW, isolated editing of documents is possible and the user always receives the *latest* version of the document.

7.2 HTML Authoring

HTML is the standard for publishing documents on the WWW. The syntax of HTML is similar to other typesetting languages such as TEXor LATEX. See [51] for details. In paper and certain electronic documents, the nature of information is sequential or linear. Due to multimedia and hypermedia capabilities, the nature of information produced in a WWW document is *nonlinear.* Therefore, the issue of HTML authoring requires special attention.

7.2.1 HTML Extensions

HTML extensions are HTML tags that are not part of the current HTML standard. Some such HTML tags have special implications towards creating educational documents (see Section 7.5.1).

7.2.2 HTML Style

In this section, we discuss stylistic issues in creating HTML documents for educational purposes. The definitive reference for HTML style is *Style Guide for online hypertext*, by Tim Berners-Lee, available on the WWW at the URL

`http://www.w3.org/Provider/Style/Overview.html`.

See also [68, Chapter 7].

- **User-Interface Design in WWW Document Development.** WWW documents can be viewed either within a WWW browser (by using a interface that is HTML-based, plug-in-specific, Java applet-based) or outside a WWW browser (by using an

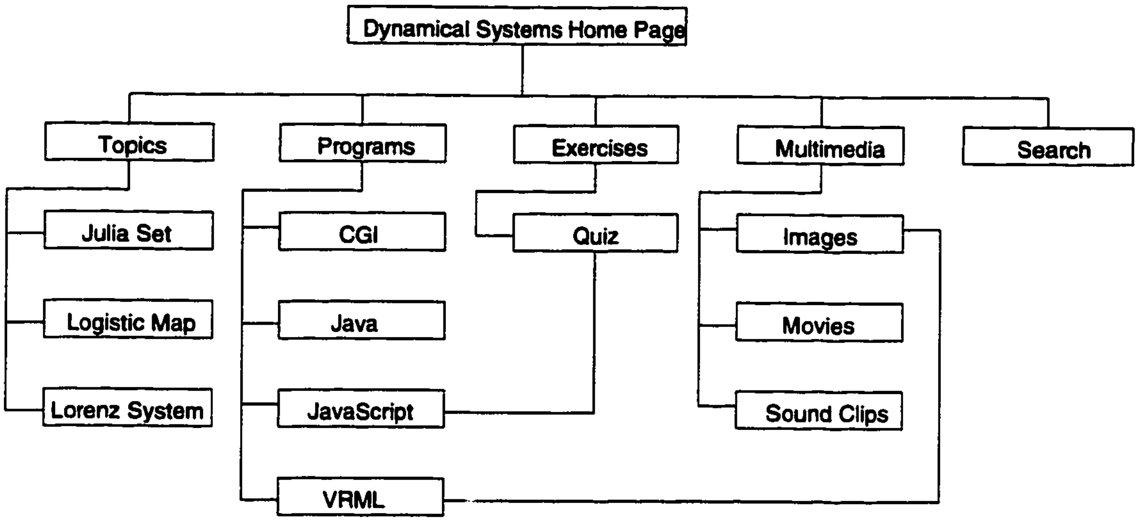


Figure 9: An Example of a Dynamical Systems WWW Site Map.

interface that is a helper application-specific). Therefore, there are WWW document design issues from a user-interface viewpoint that should be taken into consideration. Some such issues have been dealt in detail in *Yale C/AIM Web Style Guide* available on the WWW at the URL

`http://info.med.yale.edu/caim/manual/`.

- **Browser Dependency.** As far as possible documents should be written with the lowest common denominator to make the information available for access by both graphical as well as text-based browsers, without significant differences.

- **Document Structure.** Students often get lost (known as the "hyperspace syndrome") while navigating hypermedia documents. The document design should be intuitively simple, for example, based on a hierarchical acyclic 'tree' structure for easy navigation. The notions, if used, of [Next], [Up], [Previous], and [Back], on the navigation panel should follow standard conventions or clearly defined otherwise. For a WWW site with a large number of documents, a *site map* (Figure 9) can be created. A site map makes the cognitive connections between different documents by showing what documents are available at the site and how they are linked.

  Due to the nonlinear nature of the WWW, students can access the hyperlink of *any* document they come across. It is then important that they have the necessary prerequisites for it. A *guider* could be of help. The guider helps the students in their learning path: when they take a wrong direction (such as an attempt to access a topic for which they do not have the prerequisites), it displays an appropriate message (a warning and the correct route).

Some other stylistic issues are discussed in Section 7.3.

7.3 Mathematical Typesetting on the WWW

The issue of mathematics typesetting on the WWW currently poses various difficulties and is non-trivial. The challenge in putting mathematics on the WWW is to capture both notation and content in such a way that documents can utilize both the notational practices of print as well as the emerging capabilities of the electronic medium. There have been many efforts to make mathematical notation easier to display on the WWW. However, there is currently no standard method for representing mathematics notation on the WWW. In this section, we discuss related problems, and some available solutions along with their advantages and limitations. We also make suggestions towards WWW authoring style, as it pertains to mathematics. A detailed discussion of the issue of mathematical typesetting on the WWW has been carried out on the WWW at the URL

`http://indy.cs.concordia.ca/math/publ/math_typeset_www.shtml`.

7.3.1 What and for Whom

When faced with the issue of choosing an appropriate method, one should take into account one's needs and capabilities, as well as those of one's audience, in our case students. It may help to ask the following questions:

- What's the *nature* of the information being communicated?
- Does that information exist in electronic form? If so, in what format? What is its MIME type? Is that format "WWW representable"?
- Are the students familiar with the mathematical notations that will be used? Do they have the necessary tools to read it?

7.3.2 HTML-Based Approach

We can take advantage of ASCII and ISO Latin 1 character set along with certain HTML tags to include mathematical notation in WWW pages. HTML 2.0 and up provides various mathematics-specific tags[1].

Mathematics-Specific Tags in HTML

One can use the HTML-specific <SUP> and <SUB> tags to get superscripts and subscripts in WWW documents. These tags are not allowed inside preformatted text, even if they are nested inside <TT> tags. Also, not all browsers recognize these tags.

ISO LATIN 1

ISO Latin 1 is the character set that is the default standard in HTML. In addition to the ASCII character set, there are only a few mathematical symbols in ISO Latin 1. The HTML code for inserting an entity has the following components:

- an ampersand &.

[1]The only browser which supports these currently is Arena, a test-bed browser from W3 Consortium. But then Arena, unlike say Netscape Navigator, does not support various multimedia, Java applets and JavaScript.

- either a hash mark # followed by the character number, for example, © is generated by `&#nnn;` for `nnn=169` or the character name (much less support for this), for example, & is generated by `&xxx;` for `xxx=amp`.
- a semicolon ;.

It can be used with the tag <FONT FACE=SYMBOL>. This, however, does not work on all systems, and so is not an acceptably portable solution.

ASCII

The ASCII character set (which forms the first 128 characters of the ISO Latin 1 repertoire) is available to all WWW browsers. ASCII characters that have been commonly used in e–mail communication and Usenet newsgroups in mathematics are:

```
1 2 3 4 5 6 7 8 9 0 + - x * / : ^ ! = < > ( ) [ ] { } ' " | \ / _ ... ~ @ %
```

Writing mathematical expressions with ASCII is appropriate if one is restricted to the use of only standard characters such as in an HTML form. The limitations of the available characters make discussing mathematics challenging and requires one to be inventive in use of ASCII.

Guidelines for Using ASCII

Here are a few important rules that can help typeset mathematics notation, in a manner which can be accurately interpreted by other people. Some of these notations are inspired by the way they are used in programming languages (e.g., FORTRAN, C) and typesetting languages (e.g., TEX, LATEX).

- **Formatting.** When using ASCII, one should set type inside the <TT>...</TT> tags, so that it appears in a monospaced font. If one want exact positioning of characters (so that multiple spaces don't collapse into a single space), then <PRE>...</PRE> tags must used to preserve the formatting of the text.
- **Tabs and Spacing.** The key to getting ASCII mathematics notation to look visually pleasing is mostly a matter of spacing, especially when an expression takes up more than one line of text. It's important to use the space bar to line up text and not tab since the number of spaces equal to one tab can vary between computers.

Pictures with ASCII

Drawings in ASCII can be used to illustrate simple geometric figures. It is essential that monospace font (such as Courier) be used, and that tabs and variable–space fonts (such as Times) not be used. For example, a figure such as

```
    -----
   /\    \
  /  \    \
 /    \----\
 \    /    /
  \  /    /
   \/----/
```

can get distorted at the end it is received and appear as:

```
-----
/\    \
/  \    \
/    \----\
\    /    /
\  /    /
\/----/
```

Advantages and Limitations

The advantages of the HTML–based approach are reliability, portability, accessibility, searchability, efficiency, usability — it can be used in e–mail, Usenet newsgroups, WWW page. The limitations are restricted layout, limited support for mathematics notation, and limited interactivity.

7.3.3 HTML Extensions–Based Approach

An emerging new standard as a solution to the problem of mathematical typesetting on the WWW is the Mathematical Markup Language (MathML). It is an application for describing mathematical notation and capturing both its structure and content. The goal of MathML is to enable mathematics to be served, received, and processed on the WWW, just as HTML has enabled this functionality for text. MathML can be used to encode both mathematical notation and mathematical content using tags which describe abstract notational structures and tags which describe a way of unambiguously specifying the intended meaning of an expression. MathML seems to hold promise for future, though the standard exists only in form of a draft on the WWW at the URL

`http://www.w3.org/TR/WD-math/`

and is far from being finalized. Once it is, it will also be important for its broad acceptance that it is implemented in widely–used browsers.

7.3.4 Image–Based Approach

Images can be used to set equations on WWW pages. There are various editors and softwares which convert a mathematical equation into an image file. Then using the HTML image tag <IMG>, the image could be displayed and viewed using a graphical browser. However, for a successful rendering of images certain guidelines need to be followed.

Guidelines for Using Images

The quality of an image, and the way in which it is installed in a WWW page, has a significant impact on the quality and usability of that page. Following suggestions can be useful when using equations as images on WWW pages:

- **Alternative Text.** For users visually impaired or using text-based browsers, the ALT= attribute should be used to help them navigate through pages and understand the content. The alternative text, wherever possible, should convey either the visual content (for a non-inlined image) or the destination (for an inlined image).

- **Platform and Browser Independency.** A WWW page with images should be visited using different platforms and various settings on different browsers. Colors, fonts, font size, window size, etc., should be changed to check the legibility of the content.

- **Image Size.** Images dimensions should be reduced as much as possible. A WWW browser can lay out a page much faster if it knows the dimensions it must reserve for an image. These dimensions should be given in absolute pixels. The ALIGN=ABSMIDDLE attribute in the <IMG> tag is usually useful for in-line equations.

- **Image Reuse.** Wherever possible, images should be reused. This could be done, for example, for well known formulas and equations. Because images are usually kept in the memory cache of the browser, images will then appear rather quickly on subsequent pages on which those images reappear.

There are various HTML translators which, given a file in another format, will convert it into a format which can be displayed on a WWW browser. From the mathematical viewpoint, one of the most useful one is LaTeX2HTML.

LaTeX2HTML

LaTeX2HTML is a translator that takes a LaTeX file and runs all of its equations through a (Perl) server-side processing engine to make all of the mathematically typeset equations into image files (GIFs), automatically encoded into the HTML markup.

LaTeX2HTML provides several forms of output depending on the HTML version indicated in the command-line. LaTeX2HTML handles picture environments, tables, and most macros. The presence of LaTeX source file gives other alternatives: for example, it can be viewed directly on the WWW using HyperTeX browser or can be converted to a PostScript/PDF format.

The advantage of LaTeX2HTML is that one can take just about any LaTeX file and run it through the processor without knowing the intricacies of HTML itself. For those who know HTML, many LaTeX–extensions have been developed to include various HTML constructs in the LaTeX source document which are ignored by LaTeX but will processed by LaTeX2HTML.

The limitation with LaTeX2HTML is that if the document has a lot of mathematical typesetting, then a lot of image files will be produced, and could be a major download speed overhead. However, the overall size of the translated document is comparitively less than that of its PostScript form.

Advantages and Limitations

The advantages of the image–based approach is that anyone who views the equation on the WWW will see it with image parameters exactly as *the author* has specified. There are, however, the following limitations:

- If one wants to make a WWW page that has lots of text and images interspersed, the process can get tedious. Small images embedded in a line of text, can give the misleading impression of a paragraph break.
- Even if the image looks fine on one browser, it might not on another where someone may be viewing the page with different image settings.
- This method requires that one has (and knows how to use) a software other than the usual HTML editor.
- Not all users may see the images for various reasons: visual impairment, or that some users are viewing the page using a text–based browser, or are reading with images option turned off in the browser, for faster performance.

7.3.5 Browser–Support Applications–Based Approach

Documents typeset using a familiar language TeX/LaTeX could be made available on the WWW by using a plug–in (such as Techexplorer) or a helper application. There also exist plug–ins or helper applications for representing Maple and Mathematica syntax and PDF and PostScript files.

HyperTeX is a proposed protocol for embedding links in dvi files as using the `special` facility in TeX. Viewing HyperTeX files requires a helper application. The actual mechanics of this is typically handled by macro packages. Whereas with a TeXplug-in, one can only view the static document, with a HyperTeX plug–in/helper application one can actually have hyperlinks in the document.

Advantages and Limitations

Plug–in or helper application approach is quite helpful since one can use existing tools. However, both plug–ins require a *different* version of the program be developed for *each platform*. It is also difficult to search and interface with other WWW content.

7.3.6 Java Applet–Based Approach

Java applets allow mathematical expressions and geometrical figures to be inserted in HTML pages. One such example is the WebEQ Java applet. When a reader wants to look at the page, the WebEQ applet downloads with the page, and displays the mathematics notation in the browser. The problems of using WebEQ are: each equation has to be embedded in an HTML applet tag, which can be prohibitive for a document with a large number of equations, the rendering itself is quite slow, the Java rendered equations can not be printed since WebEQ is implemented in Java 1.0 (unless an applet which converts raster displays to, for example, PostScript is used). Converting Java applet displays into printable formats is possible in Java versions 1.1 and up.

Advantages and Limitations

The advantage of the Java–based approach is that the mathematical expressions can be represented *inside* the browser without substantial overhead. The limitations of this approach are that not all browsers have Java compatibility, and those users who do can turn it off, so this is not yet a completely portable solution. Also, applets preserves interactivity at the expense of download time.

7.4 WWW Publishing Tools

7.4.1 HTML Editors

To create an HTML file, any text–editor can be used. However, inserting and validating tags, and browsing the resulting document for testing purposes can become a daunting task. To facilitate HTML editing, special–purpose editors have been developed. A list is available on the WWW at the URL

`http://www.w3.org/Tools/.`

However, traditional text–editors, such as (GNU) Emacs, can be 'configured' to HTML editing. This is described on the WWW at the URL

`http://indy.cs.concordia.ca/html/doc/emacs_html.shtml.`

7.4.2 HTML Translators

There are programs for translating documents in other file formats into HTML, such as LaTeX2HTML (see Section 7.3). They can be used to convert existing dynamical systems documents into HTML. A list of such tools can be found on the WWW at the URL

`http://www.w3.org/Tools/.`

7.4.3 HTML Validators

HTML authoring is error prone where possible errors are similar to that which can occur in using typesetting languages (e.g., LaTeX). The result is a document that will display correctly up to a certain point and then display incorrectly, or even stop abruptly. Although one can check the documents using a WWW browser, this may not reflect all the errors in the document. This is because some browsers (particularly, Netscape Navigator) are quite forgiving and can recover from errors. A solution is to use a syntax-checking HTML editor; another is to an HTML syntax checking program (such as Weblint) with a traditional editor.

7.5 Dynamical Systems Lessons

Using above mentioned tools and techniques, introductory WWW-based lessons (also known as hyperbooks) on dynamical systems can be developed. Guidelines for lessons plans (though at an elementary level) are presented in [60]. These lessons can represent the information *dynamically* and *interactively*. They could also include executable programs and multimedia; this is discussed in Chapters 8 and 9, respectively. Such lessons on certain topics have been developed and are available on the WWW at the URL

`http://indy.cs.concordia.ca/ds/doc/lessons/`.

These lessons could be integrated into a course where they can serve as a supplement to the traditional course material.

There is usually a *one way* communication between the teacher and the student in this process: the teacher puts the documents on the WWW and students retrieve them. To foster interaction, particularly feedback, student should have a means to contact the teacher. This can be done, for example, by providing an e-mail address in the documents (using the HTML MAILTO tag).

7.5.1 Educational Uses of HTML Frames

HTML Frames[2] allows one to subdivide the main browser window into several panels. Each panel can contain separate documents or different parts of the same document. HTML Frames offer the opportunity to create learning environments which aid in the following ways:

- **Presentation of the Cognitive Structure of the Topic of Inquiry.** Traditional WWW documents present interface and content in separate windows. Interface structures and content pages, presented by *overwriting* the previous document in the *same* window, fragment the information and obscure cognitive connections.
- **Learning of the Presented Material.** Frame-based environments offer interface and content areas inside the same window, hence enhancing learning through visual association.
- **Navigational Opportunities for the Learner.** Using Frames, navigation can be made easier.

Frames are most useful in self-contained presentations. They are crucial when it is necessary to look at two or more windows *simultaneously*, such as topic description and corresponding

[2]Frames are proprietary to Netscape Communications; to make them a part of HTML standard is in progress.

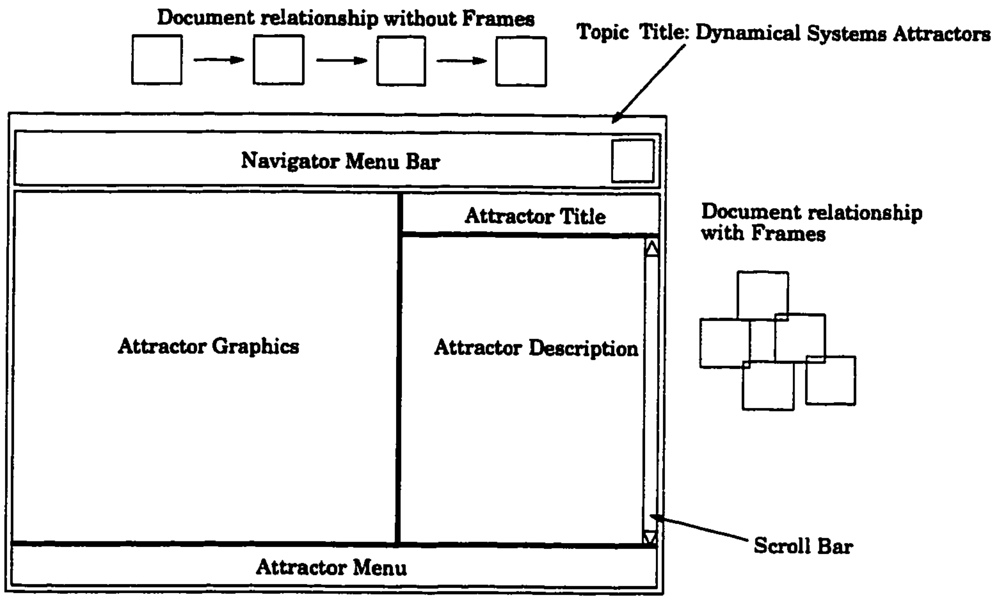


Figure 10: Use of HTML Frames in Dynamical Systems Education.

graphics. Figure 10 shows a schematic of the use of Netscape Navigator with Frames in dynamical systems education. When learning about attractors, the one of interest can be chosen from the attractor menu window. This will display the attractor title, description and corresponding graphics in separate windows. Description and graphics windows are dynamic — they are scrollable and interact with each other, while the title and menu windows are static. The menu window is always displayed throughout the session with any attractor.

While using Frames, "over-framing" can result in navigational and maintenance problems which should be taken into consideration.

7.6 Maintenance

One of the important features of the WWW is its timeliness. It is therefore essential that WWW documents be maintained on a regular basis, particularly, if they contain time-sensitive information and absolute URLs which may have changed or even ceased to exist. When moving documents from one place to another, a page with the 'forwarding address' (URL) should be provided. Otherwise, it may lead to HTTP response of "404 Document not found". Using the HTML <META> tag with HTTP-EQUIV attribute, this forwarding process can be automated.

As the number of documents grow, maintenance becomes difficult. It is then important that the entire set of documents be properly structured (such as in a tree hierarchy). In cases where a large number of documents share a common information (such as the author's name and address, hyperlink to the home page, institutional logo), use of *server-side includes* or HTML *cascading style sheets* can be made.

7.7 Copyright

It is still unclear in the area of WWW publishing as to what is freeware, public domain, shareware, and intellectual property rights. It is mostly written in legalese, which is not very helpful for the average teacher. As a bottomline, at least the protocols of paper–publishing should be respected, such as giving credits at appropriate places.

Chapter 8

Dynamical Systems: Computation on the WWW

Our present analytical methods seem unsuitable for the solution of [...] virtually all types of nonlinear problems in pure mathematics [...] efficient high-speed computing devices may, in the field of nonlinear partial differential equations as well as many other fields which are now difficult or entirely denied of access, provide us with these heuristic hints which are needed in all parts of mathematics for genuine progress.

— John von Neumann

About half a century ago the era of the use of computers in solving problems in nonlinear dynamical systems dawned, giving birth to what now is known as *experimental mathematics.* Since then, computers have played a crucial role in the development of the field of dynamical systems by facilitating detailed analysis of specific systems, suggesting new phenomena and completing mathematical proofs of outstanding conjectures. It is due to computation that the field revived from a long period of dormancy.

Reluctance towards Computation

It took time for computation as an acceptable tool in dynamical systems due to the following reasons:

- Lack of computer hardware and software that will support the level of computation required.
- Lack of a user-friendly computational environment.
- Reliability of accuracy of computer-based results.

During the last two decades, there has been a considerable improvement in the computer technology for the study of dynamical systems both in the areas of hardware and software — emergence of inexpensive and powerful computers with user-friendly interfaces and high-resolution computer-graphics connected with high-speed telecommunication networks. Correspondingly, the mathematics underlying the study of dynamical systems has undergone a substantial growth in directions that provide the necessary infrastructure for the development of computational tools that can effectively use the capabilities of such a computational environment.

For the last 25 years, computer technology was mostly in the form of time-shared facilities in teaching and learning environments. The breakthrough came in the last 10 years when decentralized and microcomputer-based computing became widely available. It is this facility which has brought great promise for the use of computer in education [28].

8.1 Motivation for Computation in Dynamical Systems

There are various motivations for a computational analysis of dynamical systems:

- In analyzing dynamical systems, particularly nonlinear dynamical systems, we are interested in their asymptotic solution structures. Theoretical analysis of such systems is usually very difficult due to the extremely complicated dynamics which often fails to give a proper insight into the asymptotic solution behaviour of a given system.
- Often, an explicit formula for the orbits cannot be given. In cases where one exists, a complicated algebraic form of it may not provide much insight.
- In many cases the dynamical system consists of a large number of equations — only a computer can follow the interrelations among so many equations.
- Determining the evolution of orbits requires iterating the corresponding dynamical system — a task well-suited for computers.
- Computation has also suggested new phenomena. For example, the phenomenon of period doubling in the logistic map could not have been readily guessed by a theoretical construction, but it is immediately suggested by a computer experiment. The detailed properties of the system have been established by a conventional proof thereafter.

Computation can benefit dynamical systems education in many ways:

- It makes the subject more *real*.
- It helps *illustrate* mathematical ideas.
- It helps student acquire *skills* required to learn a concept and work on examples.
- *Implement*, not just study, algorithms.
- It helps *exploit* and *improve* geometric intuition which can provide insight in dynamical systems phenomena
- It makes higher mathematics *accessible*.

8.2 Advantages of the WWW in Dynamical Systems Computation

Using the WWW technology for dynamical systems–related computation offers the following advantages:

- **Platform Independence.** Presently, almost all dynamical systems software packages face portability problems — they are designed for a specific architecture with a specific operating system. There is a lack of a uniform environment where computer experiments with dynamical systems could be carried out without concerns about the specifics of installation. Using a WWW browser as a universal operating system interface diverse applications can be carried out in unison.

- **Connections.** Often, classroom and laboratory components in a course are separated. At best, the teacher illustrates a static computer program for algorithms under study. It is difficult for students to make cognitive connections among the basic concepts involved in the algorithms unless they see them in action. WWW combines these components and provides an environment where the teacher can demonstrate real–time computations using an overhead–screen connected to a computer with a WWW browser and students can 'see' the connections between the textbook and corresponding computer implementation by immediately applying the concepts they have been taught. This can lead to better understanding on part of students.

- **Object–Oriented Paradigm.** Object-oriented programming (OOP) style has lot of promise for dynamical systems computing. It can enable a mathematical programmer to access, for example, an attractor as a high–level computational *object* with the same ease with which ordinary numbers are now manipulated. These objects 'know' how to retrieve themselves, how to perform appropriate operations on themselves and how to write themselves to files. This style of programming supports hierarchy and modularity through features of the language, which is important for translating mathematical structures into the computer, since mathematics is itself heavily hierarchical.

  Using OOP style, students working in dynamical systems, be able to communicate their experiments and results using very compact expressions which can be understood by other computers.

  The geometrical objects often have attribution associated with them specifying their dynamical significance. Objects can be grouped in an hierarchy with the entire phase or bifurcation diagrams as the highest–level object associated with an individual system. Asymptotic evolution of dynamical systems produces geometrical objects in the phase space such as: equilibrium points and periodic orbits; invariant manifolds; quasi-periodic invariant sets and chaotic invariant sets.

  Choosing OOP as the programming style means deciding the environment and language of choice to program. WWW is based on an object–oriented design (OOD). For example, HTTP is an object–oriented protocol. As Berners–Lee said at JavaOne conference in 1996: "[...] we now have an excuse to really use object-oriented programming". Among the languages which support object-oriented paradigm and seem suitable for dynamical systems-related computation on the WWW are Java, and to some extent JavaScript. Such programs, need not necessarily be the most (time or space) efficient, however, they suffice for educational purposes. Currently, such dynamical systems programs are scarce.

- **Data Management.** Many problems in dynamical systems require large amounts of data, both input and output, as well as data generated during the course of the computation. Data management deals with efficient storage and retrieval of this data. With the help of the WWW, programs can share the same data files. This is essential in a teaching and learning environment as the output from one student may serve as an input to another, creating a sense of group work among students.

- **Cost.** Many of the widely-used dynamical systems software packages for educational purposes are platform-specific and commercial. WWW-based programs have a broader utility and can be developed at a very modest cost.

8.2.1 Computational Requirements

When developing WWW-based programs, among the other design issues, the following need to be taken into consideration:

- **Efficiency.** Computations are performed quickly enough so that a significant time lapse is not felt at the user's end. Speed is important for calculations requiring large number of iterations.

- **Graphics.** Output results are representable by high-resolution interactive graphics with good colour capabilities. This has been discussed in detail in Chapter 9.

- **User Interaction.** Easy and flexible graphical user interface (GUI) which offers dynamic and animated displays and responds quickly when recomputation is required.

  When carrying out a computational analysis with dynamical systems, many different operations are required, and the user needs an efficient means of interacting with the program in diverse ways. Such interaction can be such as: setting initial conditions and parameter values for computations, starting computations, querying the stored database of information, visualization of output data and manipulation of graphical displays of such data, etc. When the software is being used for teaching or learning, such as for classroom demonstrations, there are forced time constraints. It is important that GUI be such that the operations of changing equations, manipulating parameters, carrying out computations, saving the output can be done in a user-friendly manner and in a short period of time so that the teacher or the student have enough time to analyze the output. It should also be interactive, such as by use of dialog-boxes and animation controls. Experience has shown that a good user-interface design can make the learning experience enjoyable and beneficial [16]. The ease of use allows students to concentrate on the *mathematics*, rather than the mechanics of making the software work. In [39, 16], the existence of a good user-interface has been emphasized as an important requirement of a dynamical systems educational software.

Teachers often do not have time (and sometimes skills) to invest in such software development. Therefore, such task is usually left to professional programmers who may not be aware of the requirements of the educational environment. A collaboration is necessary for a good product.

8.3 Server–Side Computations

8.3.1 CGI Approach

Most WWW servers support CGI. In using CGI scripts for computations, there are two possibilities: developing CGI scripts (known as handlers) which can communicate with external programs (see Sections 13.1 and 13.4 for examples) or developing customized CGI scripts which themselves carry out the computations (see Section 13.5 for an example).

8.4 Client–Side Computations

8.4.1 Browser–Support Applications Approach

Existing dynamical systems software environments can be *integrated* as plug-ins or helper applications. Plug-ins, for example, for Maple, Mathematica, and Matlab are already in existence for various platforms.

8.4.2 Java Applet/JavaScript Script Approach

Java Applets and Dynamical Systems Computation

Java offers various features to dynamical systems program development. There are many advantages of using Java applets for dynamical systems computations for educational purposes, over existing programming languages:

- **Speed.** Applets are accessible on the client-side, hence are faster.
- **User interaction.** A Java applet embedded into a long, complex WWW document can give a student a much-needed graphical illustration. Often, students can change parameters in a Java applet and immediately see the effect the change creates.
- **Platform Independence.** Till now, multi-platform versions of dynamical systems programs are scarce, and have a strong dependency on hardware. Java applets are virtually platform independent — one can run the same applet from any machine: IBM PC, Macintosh, or UNIX (commonly used platforms for dynamical systems computations). Students can access the WWW using a variety of different types of machines

and Java–compliant WWW browsers, but still all see the same information in nearly the same format.

- **Distributed and Network Computing.** Until recently, results from one application usually could not be shared by remote users in real–time. Java API has support (package `java.net`) for writing distributed and network programs.
- **Maintenance.** Usually, the process of writing a dynamical systems program is: main program (which does the computations) + GUI + graphics (which plots the numerically/symbolically obtained results), each of them often in more than one languages, which is difficult to maintain. With the support of Java API, these three components can be integrated into one uniform application.

A detailed discussion of Java applets in mathematics education has been carried out on the WWW at the URL

`http://indy.cs.concordia.ca/www/java/java_applets_edu.shtml`.

Teaching and Learning with Applets

The following are some of the ways that Java applets can be used in dynamical systems education:

- **Illustrating a Concept.** Examples of concepts can be illustrated by a moving schematic, such as of the concept of iteration. See Section 13.5, for an example.
- **Carrying out a Calculation.** This is discussed in the previous section.
- **Assessment.** For example, by designing a quiz applet.

Applets can complement a lecture and sessions with information which is difficult to convey in traditional manner. In order to be useful in teaching, the applets will need to satisfy the following requirements:

- **Speed.** Applets need to be fast, due to time limitations in a classroom environment.
- **Parameter Manipulation.** Simple and illustrative.
- **Starting, Stopping and Resetting Criteria.** Applets should *not* be designed to automatically start as soon as a WWW page embedded with the applet is visited. They need to be a facility (such as a button) for starting the applet so that the GUI features of the applet can be introduced to the students prior to the computations. Applets should be designed to stopped at any stage of the computation so that intermediate results can be explained. Resetting criterion is useful for carrying out computations with default parameter values.

From the point of view of learning, the applets will need to satisfy the following requirements:

- **Help.** The help can, for example, include prerequisites to the concept or usage instructions of the program under study.
- **Experimentation–Oriented.** Applets should facilitate *experimentation* rather than observation of the evolution of a dynamical system like in a movie. For that applets need to have an interactive GUI with carefully chosen parameters that can be manipulated.
- **Robustness.** Students can make errors during a computer-based learning process (such as choosing an out-of-range parameter). Applets should be designed to accomodate those errors and prompt the user with appropriate messages.

8.4.3 Limitations

Inspite of the tremendous potential that Java has to offer towards dynamical system-related computations, there are a few limitations:

- **Efficiency.** Java applets that currently use a WWW browser with a Java interpreter (such as Netscape Navigator 2.0) run relatively slowly compared to the ones which use a WWW browser with a JIT compiler.
- **Application Development Environments.** There is current lack of application development environments for writing Java programs for scientific computation. (Java API provides only 'standard libraries'.) For applets written for educational purposes (which tend to be small), however, this is not a serious limitation. The situation should get better over the time with the emergence of reusable class libraries. One such effort is the Java Numeric Library (JNL) from Visual Numerics.

JavaScript

JavaScript scripts can be designed and implemented to carry out light-weight dynamical systems computations. Such scripts will be platform-independent but can only provide a limited HTML interface.

8.5 Limitations of Computation

It is possible for students to get an impression that the computer can answer all their questions and solve all their problems. The use of computer as an experimental tool in the study of dynamical systems can stimulate and motivate new ideas and problems, but these experiments alone can not provide understanding of *why* the observed phenomena happen. Also, various mathematical results are needed when implementing any algorithms (which should converge) and verifying the correctness of computed results (which should be reliable). This brings us from computer science to the realm of mathematics. An emphasis of this interplay, dynamical systems theory $\longleftrightarrow$ computation, supported with illustrative examples, can help students appreciate the power of mathematics.

8.5.1 Open Programs vs. Black–Boxes in Education

It is quite common to use black–boxes in programming in form of subroutines, classes/objects, libraries, and using other programs. With the use of a GUI also, the underlying program becomes a black–box. Black–boxes are necessary to hide complexities of programming for which students might not be prepared. In mathematics, when we have a black–box then there are mathematical concepts hidden inside. If students can not see the internal workings, then they may not learn those mathematical concepts *instrinsic* to the implementation of the black–box — they may learn 'how' but not 'why'. For proper use of black–boxes in education:

1. Students must have access to the inside of the black–box, if needed, and
2. Students should (be motivated to) understand the theoretical background and functionality of algorithms in computational studies of dynamical systems, wherever necessary.

8.5.2 Reliability

> *Numerical experiments are crucial to the development of insight into the behaviour of dynamical systems. However, [...] it is [also] of crucial importance to understand how much of what we see in computer–generated pictures of chaotic attractors are artifacts, [...] and how much is real.*
>
> — James A. Yorke

In computer experiments with dynamical systems, computer–generated trajectories, and the statistics obtained from these trajectories, are dependent upon the computer's hardware/software responsible for making round–off/truncation decisions. The theoretical and computational solution behaviour of certain dynamical systems, particularly chaotic systems, will almost always differ significantly due to sensitive dependence on initial conditions characterizing such systems and limited precision from the necessarily finite discrete arithmetic on the computer. A schematic of the various stages in the life–cycle of a computer-generated attractor of a dynamical system is given in Figure 11.

From the viewpoint of a classic dialectic, we then face following question: In what sense and to what extent do the results of computer experiments with dynamical systems reflect the true dynamics of the actual system? This classical question stimulates careful mathematical and computational analysis.

During teaching, the distinction between the studies related to 'real' system and its computer model can be made with appropriate reference to pitfalls[1]. In doing so, the need for mathematical analysis of computer–based results is realized. This, once again, can help students draw the connection between mathematics and computer science, and can make

[1] In [5], examples of one–dimensional maps as dynamical systems are given and it is shown that in an effort for computing their orbits how misleading the results can be owing to the fact that virtually all computers do arithmetic in binary. One example is the binary shift map $f(x) = 2x (\mathrm{mod}\, 1)$ shown in Figure 12.

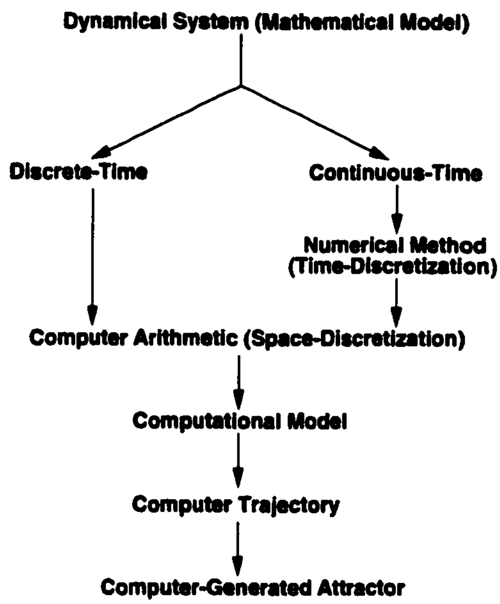


Figure 11: Life–Cycle of a Computer-Generated Attractor.

them experience and realize the power of mathematics. The following example, which can be carried out in a classroom, demonstrates the limitations of using finite arithmetic in computations with systems which have a sensitive dependence on initial conditions, such as the chaotic systems, in two different ways. Two calculators, CASIO $fx - 7000G$ and an HP28S, with 10 and 12 decimal digits accuracy respectively, were used for this purpose.

Example 8.5.1 (The Parable of the Parabola) The Verhulst's equation is given by

$$f(x) = x + ax(1 - x).$$

Case 1. Starting with the initial condition $x = 0.01$ and $a = 3$, the results of first 40 iterations on the two calculators are shown in Table 1.
Case 2. Now, $x + a(1 - x)$ can be equivalently expressed as $(1 + x)a - xa^2$. Starting again with the initial condition $x = 0.01$ and $a = 3$, in both the implementations, the results of first 40 iterations are shown in Table 2.

Tools of Reliability

After the initial cautionary message, it can be shown in a simple setting to students that there exist tools which can guarantee a high degree of optimism towards computer–generated results with dynamical systems. Limitations of these tools [37] should also be discussed.

Geometrical Approach

A tool which has been used frequently while investigating the question of reliability of computer orbits is the *pseudo–orbit shadowing property* (or just shadowing property) of the system. While the pseudo–orbit will diverge exponentially from the true orbit with the

Iteration	CASIO	HP
1	0.0397	0.0397
2	0.15407173	0.15407173
3	0.5450726260	0.54507262626044
4	1.288978001	1.28897800119
5	0.1715191421	0.171519142100
10	0.7229143012	0.722914301711
15	1.270261775	1.27026178116
20	0.5965292447	0.596528770927
25	1.315587846	1.31558435183
30	0.3742092321	0.374647695060
35	0.9233215064	0.908845072341
40	0.0021143643	0.143971503966

Table 1: Verhulst's Equation: Same implementations on Different Calculators give Different Results.

Iteration	$x + a(1 - x)$	$(1 + x)a - xa^2$
1	0.0397	0.0397
2	0.15407173	0.15407173
3	0.5450726260	0.5450726260
4	1.288978001	1.288978001
5	0.1715191421	0.1715191421
10	0.7229143012	0.7229143012
11	1.323841944	1.323841944
12	0.03769529734	0.03769529724
13	0.146518383	0.1465183826
14	0.5216706225	0.5216706212
15	1.270261775	1.270261774
20	0.5965292447	0.5965293261
25	1.315587846	1.315588447
30	0.3742092321	0.3741338572
35	0.9233215064	0.9257966719
40	0.0021143643	0.0144387553

Table 2: Verhulst's Equation: Different Implementations on Same Calculator gives Different Results.

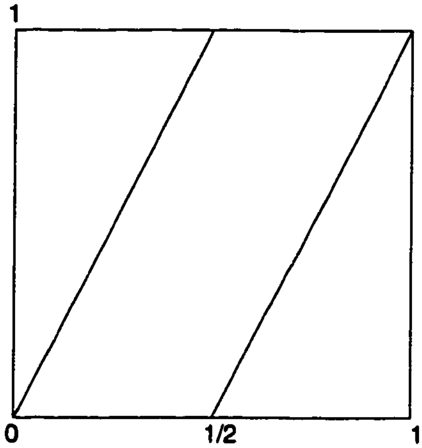


Figure 12: Binary Shift Map.

same initial point, there often exists a *different* true orbit with a slightly different initial point which stays uniformly close to the pseudo–orbit, i.e., *shadows* it for long period of time.

Let (X, d) be a metric space. A sequence $\{\tau^n(x)\}_{n\geq 0}$ is called the (true) *orbit* of the point $x \in X$. For a given $\delta > 0$, a *δ–pseudo–orbit* for a transformation $\tau : X \to X$ is a sequence $\{x_n\}_{n\geq 0}$, $x_n \in X$ for each n, such that

$$d(\tau(x_n), x_{n+1}) \leq \delta,$$

for each n. Given an $\epsilon > 0$, a δ-pseudo-orbit $\{x_n\}_{n\geq 0}$ for τ is *ϵ-shadowed* by an $x \in X$ if,

$$d(\tau^n(x), x_n) \leq \epsilon,$$

for every n. τ is said to have the shadowing property if for every $\epsilon > 0$, there exists a $\delta > 0$ such that each δ—pseudo–orbit for τ is ϵ–shadowed by some point of X.

Figure 13 shows a schematic of the shadowing property. As an example, in [33, page 186], it is shown that the binary shift map (Figure 12) has the shadowing property.

Statistical Approach

In many situations, determining the asymptotic geometric structure of the attractor is not feasible due to extreme complication of the dynamics. Instead of studying the attractor itself, we can then study the *statistical* behaviour of the system on the attractor. Also, since we can not always know the initial conditions exactly, we are interested in the statistical properties of trajectories of dynamical systems that are shared by the entire *classes* of initial conditions and whether the computer orbits of the system 'exhibit' any of those properties. Invariant density is one such property which has been shown to 'survive' during computer discretizations [27].

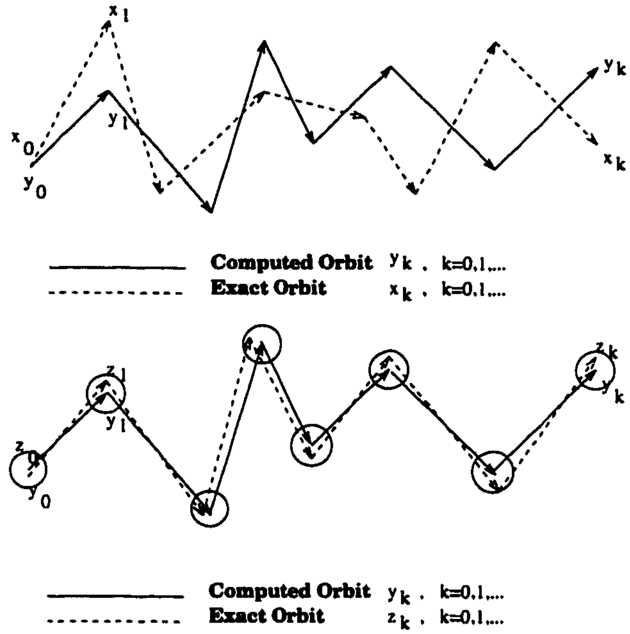


Figure 13: Shadowing Property Illustrated. With x_0 as the starting point, the Exact Orbit is $x_0, x_1, \ldots, x_n$. The Computed Orbit starts within an ϵ–distance at y_0 and Deviates from the Exact Orbit. However, in the ϵ–shadow of the Computed Orbit $y_0, y_1, \ldots, y_n$ there is an Exact Orbit which starts at z_0.

8.5.3 Limits of Computability, Computational Irreducibility and Undecidability

There are geometric structures in dynamical systems which due to time or space complexity may reach their computability limits. Such studies should be indicated to students. For example, in Plant's model of bursting nerve cells (a system of five ODEs), when set–up as a boundary value problem, methods used for computing homoclinic orbits are quite accurate and robust. However, calculations demand much computation time and, in extreme cases, become impossible due to increasing ill–conditioning of the boundary value formulation. As Doedel states [18]: "Although the use of more numerically stable linear solvers may somewhat extend the computation domain, it is likely that the calculations [...] approach the computability limits."

Extensive use of computation in dynamical systems study might lead students to believe that if not by analytical methods, using the computer might solve all their 'dynamical' problems. This leads to the question: Can the computer *always* tell us the asymptotic behaviour in the evolution of a dynamical system?

For dynamical systems which are capable of universal computation (mimic the behaviour of any possible computer), there is no algorithm which can speed–up their process of evolution — they are *computationally irreducible.* There are examples of cellular automata [78] whose ultimate pattern form is a result of an infinite number of steps, corresponding to an infinite computation (see Figure 14). If these cellular automata are also computationally

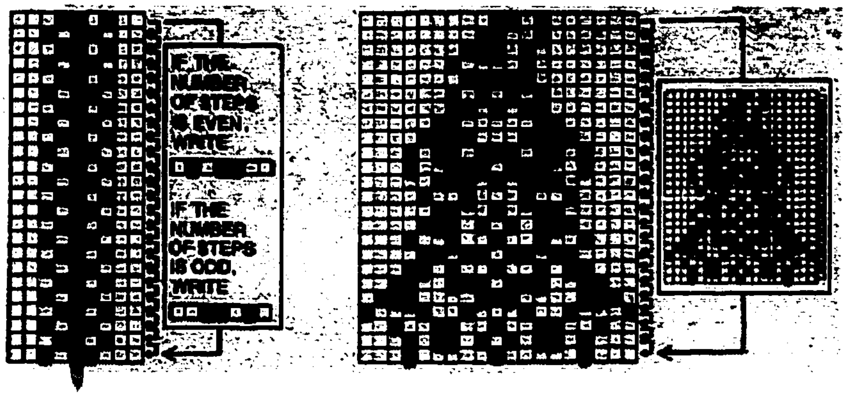


Figure 14: Computational Reducibility and Irreducibility. For the system shown at the left, the formula merely requires that one find the remainder when the number of steps in the evolution is divided by 2. The system is Computationally Reducible. For the system at the right, the behaviour is so complicated that in general no such description of the evolution can be given. The system is Computationally Irreducible.

irreducible, then the asymptotic evolution of their pattern can not be reproduced by any mathematical or computational process. Any questions about the ultimate behaviour of such a system (such as whether a particular pattern will ever die out) must therefore be considered *undecidable*[2]. Computational irreducibility may well be the rule rather than an exception in chaotic systems. This imposes a fundamental limitation on what is and what is not possible in computation of dynamical systems.

8.6 Didactics of Computation in Dynamical Systems

By introducing a computer–based environment in dynamical systems, we are faced with the following pedagogical, epistemological and cognitive considerations [80]:

- Where in the dynamical systems curriculum is computation appropriate, and why?
- How does computation change what students should know?
- Will computation help students learn dynamical systems concepts more deeply, more easily, and more quickly?
- How and to what extent is a student's cognitive behaviour affected?

Didactics of mathematics studies the student's acquisition of mathematical knowledge. The basic object is the *didactical system*: the knowledge, the teacher, the student and the interactions and relationships among them. This can be represented by the triangle [10] which changes with the introduction of a computer as shown in Figure 15. There is a *didactical transposition* as a result: from a dynamical systems knowledge to the dynamical systems knowledge to be taught. We are then interested in how the three 'nodes' in this triangle have changed.

[2]This can be viewed as a manifestation of Gödel's Theorem.

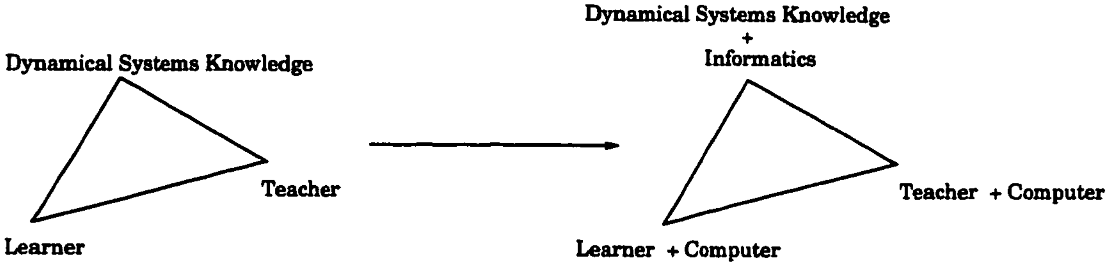


Figure 15: The Didactical System Transformation.

The subject of dynamical systems changes in the following way by the introduction of a computer:

- **Problem Solving.** By numeric and symbolic computation, by use of computer graphics, by use of techniques from artificial intelligence and expert systems for discovering proofs and automatic proving.
- **Concept of Proof.** From an existential to a constructive proof.

The change in the subject leads to the change in the *content of teaching*:

- **Computer as Alternate 'Black–Boards'.** By supplementing lectures with selected computer–based demonstrations prepared in advance to illustrate dynamical systems phenomena (e.g., period–doubling bifurcation) and objects (e.g. attractors) where teacher plays extemporaneous variations of the theme.
- **Computer as Laboratory Extensions of Dynamical Systems Courses.** By setting–up a LAN of (teacher and student) computers individualized or a group instruction can be carried out, and results can be shared.

Hardware and software do not automatically provide knowledge: we need a pedagogical environment to build activities within a pedagogical strategy. This is provided by *didactical engineering*, which consists of techniques and methods to suggest appropriate teaching strategies. Therefore, when evaluating and suggesting a computational environment such as of the WWW for educational purposes, it is crucial to keep in perspective whether it readily supports teaching strategies for efficient use. In particular, that applies to evaluating WWW sites for educational use.

8.6.1 Computation and Learning

Empirical evidence has shown that the use of computer proves more successful in education when it is used to enhance meaning and construction of concepts. The question then is: How does the computer *affect* the *process* of learning a concept?

To answer the above question to a greater extent, we must consider the computer as a part of the mechanism of learning and ask: what knowledge is produced by using a computer?

8.6.2 Computation, Cognitive Development and Cognitive Obstacles

There have been studies of how knowledge is structured within the mind of a student and models of the cognitive structure have been given. Prior to teaching a given concept, students have certain *preconceptions*. All these conceptions (definitions, theorems, examples) are organized in what is known as the *concept image*. The teacher can act upon this concept image and add to it all the conceptions that are necessary for using the concept. A dynamical systems computation environment that keeps this perspective in their software design can then improve the student's concept image of concepts such as iteration, bifurcation, etc.

It is known that computers can assist in cognitive development [53]. Recent research into concept development has led to the hypothesis that learning may be improved by helping students *construct* knowledge in their minds in a context which is designed to aid, or even stimulate, that construction. One way of doing this is through designing computer programs which use a *combination* of mathematical and cognitive principles — building on what students already know in way which is consistent with their cognitive development.

Often, (logical) errors are made by students due to the logical sequence of their knowledge, which leads to a cognitive obstacle. Therefore, a careful study of the different possible ways the knowledge about a concept is structured and of the conceptions which can occur, by analyzing the errors by a student, and designing computer-based exercises accordingly, can allow the teacher to discover a student's knowledge.

8.6.3 Computation, Epistemological Analysis and Epistemological Obstacles

An epistemological study based on Piaget's theory [56] of the use of computer in learning a mathematical concept has been carried out in [19]. It describes which kind of abstraction is necessary at each step of learning, gives the organization of these steps and obtains the genetic decomposition (a tree of knowledge and skills) of mathematical concepts. This decomposition is then used to elaborate didactical situations which will help the student traverse the paths of decomposition. A dynamical systems computation environment which assists students in passing steps in the genetic decomposition of a concept can then improve learning.

Epistemological obstacles [65] are obstacles in learning a mathematical concept which are part of the concept and it is necessary to encounter them in order to acquire the whole concept. They consist of a knowledge, which is sufficient to solve a certain range of problems, but which become insufficient to treat a new type of problem. To overcome them, it is necessary to eliminate the previous knowledge and to build a new knowledge suited to the new problem. Most of these obstacles exist in the history of the construction of the concept. Computer-based activities which have undergone an epistemological study and a subsequent didactical study of the concept, and elaborate situations and problems in which the new knowledge is efficient can then, hopefully, overcome epistemological obstacles.

Chapter 9

Dynamical Systems: Multimedia on the WWW

Multimedia. These are the axes of revolution.

— Tim Berners–Lee

In this Chapter, we ask the following questions: What is multimedia and what type of multimedia environments are there? What is role of multimedia in dynamical systems education?

Multimedia is communication using more that one medium. In our context, it can be a computer–based device with text–display, graphics–display, audio and video capabilities. We shall restrict ourselves to two aspects of multimedia: visualization and sound.

9.1 Dynamical Systems and Visualization

9.1.1 Mathematical Visualization, Computer Graphics and Dynamical Systems

The use of computer graphics is now in the process of altogether changing the role of the eye [...] as an integral part of the very process of thinking, search and discovery.

— Benoit B. Mandelbrot

Dynamical systems evolution gives rise to various patterns. One of the most fruitful ways of realizing these patterns is via *mathematical visualization*. Mathematical visualization is a process of forming images and using such images effectively for mathematical discovery and understanding. In mathematical visualization, the interest is in visualizing a concept or a problem (which means to understand the problem in terms of a diagram or visual image), *not* in visualizing a diagram (which means to form a mental image of the diagram). To achieve this kind of understanding, requires learning how ideas can be represented numerically, symbolically and graphically, and to move back and forth among these modes. It also requires the ability to choose the approach most appropriate for a particular problem, and to understand the limitations of these three dialects of the mathematical language. The most important contribution in the field of visualization of dynamical systems has been via computer graphics.

Reluctance towards Visualization

Visualization, particularly computer graphics-based, as an acceptable tool in studying mathematics was not encouraged initially [21]. There were doubts about the validity of visual proofs and low computing power to support it. Computing now has enough power and resolution capabilities to support visualization with accurate representations of problems and their solutions. This helps restoring the visual side of mathematics and opens new possibilities in mathematics problem-solving. Visualization has been encouraged in mathematics education recently, as reflected by the "Proof without words" section in *College Mathematics Journal*, *Mathematics Magazine* and *American Mathematical Monthly.*

9.1.2 Motivation towards Visualization in Dynamical Systems

Computer graphics can not only aid dynamical systems study, it can also become a *necessary* tool in cases which lack symbolic representations. In cases even where a symbolic representation is available, it might *need* the geometric interpretation, as seen by the following example.

Example 9.1.1 Consider the following first order ODE

$$y\frac{dy}{dx}\sec 2x = 1 - y^2 \tag{1}$$

By the separation of variables method, it yields

$$\frac{dy}{dx} = \cos 2x\frac{(1-y^2)}{y}, \tag{2}$$

which on integration gives

$$-\frac{1}{2}\ln \mid 1 - y^2 \mid = \frac{1}{2}\sin 2x + C \tag{3}$$

Even though we have an analytic solution (3) of of (1), what does equation (3) *mean*? By regarding equation (2) as specifying the direction of the tangent vector (dx, dy) to the solution curve through any point (x, y) results in a direction field of line segments to be drawn in appropriate directions through an array of points in the plane. This gives some solutions as closed loops and others as functions of the form '$y = f(x)$' (see Figure 16). The symbolic solution here is of little value without the graphical representation of its meaning, and the graphical interpretation lacks precision without its symbolic counterpart.

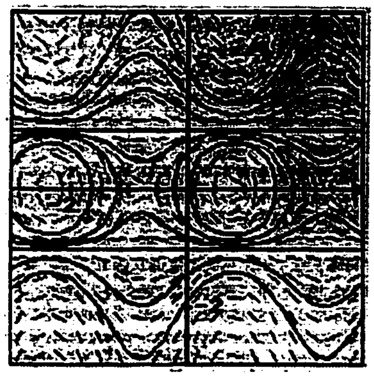


Figure 16: Graphical Solution to a First Order ODE

MacMath [36] is a differential equations software package which emphasizes qualitative approach through computer graphics. Without computer graphics, the authors found that students had difficulty appreciating the notions of existence and uniqueness of solutions. Students faced difficulty in understanding how a solutions could exist if it could not be expressed as a formula. Computer graphics helped them to see the existence as *the ability to draw the solution* — a solution that existed visually. It also made them appreciate the power of a formal mathematical result — that a solution exists and is unique provided that the differential equation specifies a direction to follow at each point.
Specific advantages to visualization in dynamical systems education are:

- Taking advantage of one of our primary physiological organ of sense — eyes.
- Representing pictorially large set of numbers. Data sets produced by multi–dimensional dynamical systems are often very complex and often very large in number such as iterates that result during computation. Such data sets just cannot be printed and/or it is not possible to acquire an understanding of the system by generating entire data set and attempting to analyze the numerical output even if they are printed.
- Ability to focus on specific components and details of complex problems. By magnifying specific areas of a phase and bifurcation diagrams, a better understanding of the systems under study can be achieved. Often, properties such as self–similarity or symmetry can be much easily observed in this diagrams, than in their numerical/symbolical representation.
- Ability to show *dynamics* of systems. Visualization comes very close to making mathematical results visible, thus interpreting *δεικνυμι* — the word used by Euclid for 'to prove'. In contrast to verbal or symbolic presentation, the approach of visualization can often be more subtle to mathematical structures in a learner's imagination.

9.2 Phenomenology of Visualization

What kind and level of visualization is appropriate for the purpose of education? There are three kinds of visualization:

1. **Postprocessing.** Knowledge is complete and the user is creating a display of a finished product.

2. **Tracking.** Knowledge is being developed and the user is watching it being displayed to see its nature.

3. **Steering.** User is in the processing loop and can interact with and manipulate the simulation as it is underway.

Educational studies have shown that students respond better to dynamic rather than static images, i.e., (1) and (2). In particular, (3) gets students involved in development. There are three levels of visualization:

1. **Presentation level.** Show work to the audience.

2. **Peer level.** Communicate to others working at same level.

3. **Personal level.** Understand personal ideas in development.

(1) has too much effort involved to be effective in teaching, and (3) requires more knowledge of what is being displayed than what students have. (2) seems to be the right level for education.

9.3 Forms of Visualization

9.3.1 2D and 3D Graphics

The understanding of dynamical systems often depends on the ability to visualize the geometrical objects associated with it. In two–dimensions (2D), objects can be visualized readily. 2D graphics can be used to illustrate phase and bifurcation diagrams of simple dynamical systems, such as one–dimensional maps. However, many dynamical systems with interesting behaviour are high dimensional (for example, the Hénon map or Chua' equations (4), and have attractors which are three–dimensional. In three–dimensions (3D), visualization is difficult, but some success can be achieved by rotation and 'real–time' motion of the object, for example, of curves and surfaces. Although, 2D projections can be quite useful and lead to insights, it is still a compromise. Also, such projections can be misleading and are not always feasible. In such a case, 3D graphics is necessary. Visualization becomes particularly difficult in higher dimensions. In such a case, 2D projections onto different coordinate planes can provide some insight into higher–dimensional phase spaces. 3D projections of the phase space with interactive panels for generating motion with change of viewpoint can also be useful. Similarly, visualization of the bifurcation diagram is also very significant. For example, by assigning symbols with dynamical meaning, a map of 2 or 3D parameter space respectively can be constructed which displays the regions associated with each type of phase diagram.

9.3.2 Film, Video and Animation

We can use animations to provide students with a sense of how processes behave over time. Sometimes we can even try to animate the *steps* of our solution algorithms so that students can develop an understanding of how each step in the algorithm transforms the initial conditions into a final result. The animation features provided by current software (like Maple's `animate` and `animate3d` routines) allows us to visually simulate additional dimensions of problems.

Video can provide a point of contact with the real–world as a background for computation, or if the images are matched carefully, as a check on the computational accuracy. Video images with suitable computer enhancement can also guide the student through an examination of complex structures.

Dynamical systems, and in particular, bifurcation theory, is the study of how systems change in time, so it is only natural to record these events continuously on film. In 1984, with the help of computer–generated animation, Devaney [15] made a film on the explosion of Julia set of the function $(1+\epsilon i)\sin z$. This and few other films made along similar lines eventually led to the discovery that Julia sets of entire transcendental functions can "explode" while undergoing a series of homoclinic bifurcations. It is crucial here, that without using a film, it would not have been possible to put the record of the order in which all the homoclinic bifurcations occur into perspective.

This, has useful educational implications, as Devaney [15] states:

> Perhaps, the most important consequence of these films will not be the research they inspire. Rather, the films seem to be particularly effective means of communicating the beauty of mathematics and the excitement of mathematical research to younger students. The students seem fascinated by the intricate patterns that they see on the screen [...] the realization that all is not known about sine or cosine [...] or in mathematics in general, is an important message that is easily communicated with film.

9.4 Aspects of Visualization

9.4.1 Colours

Colours in graphics provide invaluable insight in visualizing the local and global behaviour of dynamical systems. Pseudocolouring a function over a domain, a widespread technique in applied mathematics, has revealed various hitherto hidden phenomena in nonlinear dynamical systems.

Use of colour–coded representations in the graphics can serve several purposes:

- Reflecting 'depth' in the display.
- Identifying different orbits.
- Associating dynamical meaning, such as the stability properties of the orbit.
- Showing Variation of a parameter, such as temperature distribution.

The purpose is to use colour in controlled ways that will increase the information content of the diagram. The simplicity of integrating colour allows us to focus on using colour itself to portray additional dimensions of a problem or to highlight critical features.

9.4.2 Interaction

The geometric structures of dynamical system are often intricate and extremely sensitive to change in the system parameters, so that interaction with the calculations and visualization is necessary to refine them and interpret their meaning.

Interactive computing has revolutionized the media of analysis and learning. We can harness human perceptual processes more fully and conduct exploratory as opposed to confirmatory analysis of data — "If a picture is worth a thousand words, this one is worth a million, because I can ask this picture questions." Visualization provides the graphical displays and animation on which users base their observations — all human senses can be employed, vision, hearing and even touch. Advanced interaction techniques enable rapid exploration of data. The techniques available allow different resolution levels, parameter settings, processing options, feature extraction and detailed textual and numeric enquiry when needed.

9.5 Server–Side Visualization

9.5.1 CGI Script Approach

CGI scripts, particularly handlers, can be quite useful in visualizing dynamical systems, though their GUI is limited to that provided by HTML. To associate a visual component in a CGI script carrying out a dynamical systems computation, probably the easiest approach is to pass the numerical results to external programs (such as GNUPLOT/Maple) which return 2D/3D plots as image files that can be rendered on the WWW browser. An example of such as approach is available on the WWW at the URL

`http://www.geog.ubc.ca/numeric/course/labs.cgi`.

9.5.2 Clickable Imagemaps

Clickable imagemaps are a special form of hyperlink involving an in–line image. Clicking on different parts (hot regions) of the image takes the user to different pages. Clickable imagemaps can be used to describe salient features of graphic images generated from dynamical systems computations. For example, in a bifurcation diagram of a system, representative

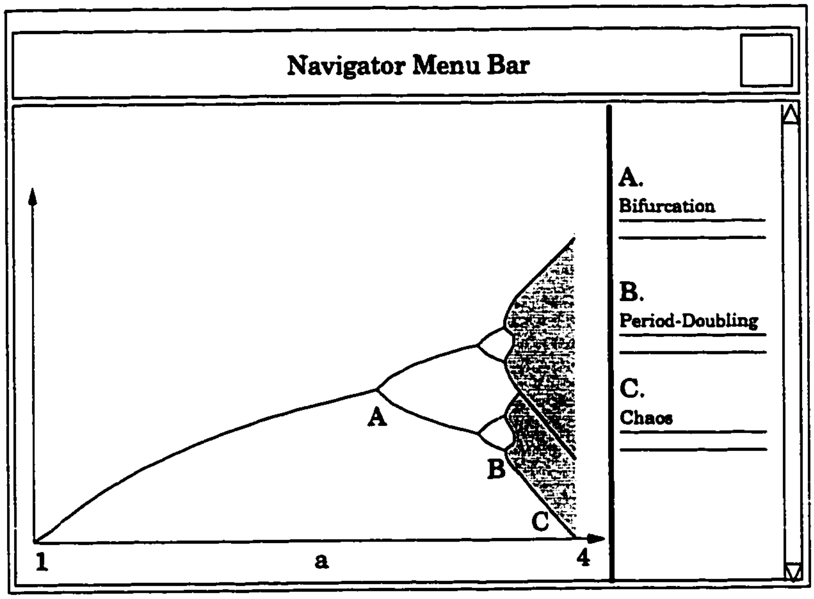


Figure 17: Period–Doubling route to Chaos in the Logistic Map $\tau(x) = ax(1-x)$. Regions A, B, C in the imagemap in the left frame have hyperlinks corresponding to the definitions of Bifurcation, Period–Doubling and Chaos, respectively in the right frame.

areas reflecting key aspects of solutions can be labelled and hyperlinked to appropriate text for descriptions. This can be particularly useful in conjunction with HTML Frames. One such example is depicted in Figure 17.

9.6 Client–Side Visualization

9.6.1 Browser–Support Applications Approach

Images and movies can be developed, made available and can be viewed with the assistance of a WWW browser with the relevant plug–in/helper application. A collection of dynamical systems movies is available on the WWW at the URL

`http://indy.cs.concordia.ca/ds/animation/`.

Visualization using VRML

For visualizing higher dimensional attractor in dynamical systems, 3D graphics is important: 2D projection of a 3D object is a compromise. Also, what is needed is *interactive* graphics. This is where VRML can be very useful. For example, with a VRML viewer one can navigate through complex data, to view complex objects such as attractors from multiple viewpoints. Currently, to view a VRML file on the WWW browser, an appropriate plug–in or helper application is needed. Figure 18 shows the Mandelbrot set using the helper application VRweb.

Figure 18: Mandelbrot set in VRML.

The present limitation of using VRML for dynamical systems–related visualization is the absence of a rich API, which we hope will improve, as the language evolves.

Dynamical systems and VRML resources are available on the WWW at the URL
`http://indy.cs.concordia.ca/www/vrml/ds/`.

9.6.2 Visualization with Java Applets

Java has significant implications towards creating WWW–based multimedia applications for various reasons. Java applets are platform–independent; can interface with various forms of media: text, graphics, animation and sound; interact with other applets and with programs on the server; can work in conjunction with JavaScript and VRML.

Java API (package `java.awt`) supports creating and manipulating images. Animations can also be created in Java by drawing successive frames at a relatively high speed on the screen. Applets that have active elements (buttons, menus, etc.) allow the learner to gain experience with a concept in a way that allows the learner to have control of the experience. In particular, applets with a well–designed GUI give valuable experience with constructive, programming concepts without having to be concerned with the syntax.

9.7 Teaching and Learning to Visualize in Dynamical Systems

Learning to visualize mathematical patterns enlists the gift of sight as an invaluable ally in mathematical education.

— Lynn A. Steen

When discussing the issue of visualization in dynamical systems education, one has to consider the *technical* aspect and the *sociological* aspect.

The task of the teacher is to find opportunities where the use of visualization techniques can provide students with an intuitive grasp of what often, at least initially, may seem to be very difficult and unintuitive concepts. In doing so, the following can be useful:

- **To Visualize or Not to Visualize.** *Which* material is best understood symbolically/visually and their connections should be determined. Example 9.1.1 illustrates an instance where such an interplay is justified.

- **Highlighting Properties that Vary.** Colour or animation can be used to highlight properties that *vary*. Human visual processing is biased towards noticing change, whether it be due to variations in colour, texture, reflectance, or motion. For this reason, we can effectively focus a student's attention by causing change to highlight features of a diagram that we consider to be important.

- **Providing Frame of Reference.** Invariant properties can be used to provide a frame of reference. Particularly when observing an animation, we benefit from the presence of some stable reference points that do not change in the midst of the surrounding motion. Examples of such reference points are using axes to frame an image, drawing an asymptote that a function will approach, or highlighting some point in the solution space that remains unchanged in the presence of surrounding activity.

- **Coordinating Related Concepts.** Simultaneous displays of related concepts should be coordinated. Colour can be used to emphasize how the various parts of the image are related. For example, if simultaneously plotting iterates of a function, the initial conditions, and (attractive and repelling) fixed points, one set of colours can be given to attractive fixed points and their corresponding initial conditions, and another set to repelling fixed points and their corresponding initial conditions to emphasize the relationships. Effort should be made not to show too many different things at the same time. Though it is good to simultaneously display related concepts, too much information is being displayed at the same time may lead to confusion among students.

- **Helping by Visual Cues.** Visual cues can be given to help students *understand* what they are seeing. When a complicated diagram is being generated, some visual cues can be provided to help students understand what they are looking at and how they should interpret it. Informative titles, axes labels and colurs are also important ways of providing useful visual clues that aid the student in interpreting a diagram. In generating 3D plots, hidden–surface renderings or perspective transformations can be used to increase a viewer's sense of depth when looking at the picture.

- **Warning against Fiction.** Visual plots are often based on numerical approximations and are subject to the same potential errors and degenerate behaviours. It should be known what the plot will show and it should be verified that those critical features are indeed present in the result and that no fictional features have been added. If undesirable features are unavoidable, they should be brought to attention of students.

9.8 Limitations towards Visualization

Some of the current limitation towards visualization are:

- **Visual Limitation.** If we wish to analyze two different results which are very close, say at the 10th decimal place, it might not be possible to distinguish their graphical representation.
- **Visual Impairement.** For those who are visually handicapped, description of a concept strongly dependent on graphics is of little or no use.

9.9 Obstacles to Visualization

Inspite of the above mentioned benefits, certain students do encounter visual obstacles in form of geometrical obstacles.

- **Beliefs about the Nature of Mathematics.** Perpetuating the view among students that mathematics it is nonvisual regardless of whether or not a visual representation is at the base of an idea.
- **Technical.** Until recently, many computers with sophisticated visualization capabilities were unaffordable for educational purposes. This has seriously affected the educational visualization software development, and education itself. Cost of visual tools and lack of sufficient software development are two related technical obstacles.

  Another drawback, particularly in 3D graphics, has been the absence of any programming standards. There has been a strong dependency on hardware (e.g., programming in SGI IRIS GL). There are now some 'open' standards available such as openGL but mainly for X Window System.

  Now with the evolution of the WWW, this situation is improving.

- **Reliability — Is Seeing, Believing?** "Truth, in science, lies not in the eye of the beholder, but in objective reality [44]". In many cases, graphical results can be insufficient indicators of the system's behaviour. This may lead to incorrect conclusions as illustrated by the following example.

  The *Mandelbrot set* $\mathcal{M}$ (see Figure 3) arises from the iteration of complex quadratic function

  $$f(z) = z^2 + c,$$

  where $c = x + iy$, and is defined by

  $$\mathcal{M} = \{c : f^n(c) \text{ is bounded as } n \to \infty\}.$$

An interactive facility for *zooming-in* on different parts of the Mandelbrot set is available on the WWW at the URL

`http://www.vis.colostate.edu/~user1209/fractals/explorer/mandel.html`.

Ewing [22] has expressed his skepticism concerning the computer-graphic images of $\mathcal{M}$:

> We don't know the area of the Mandelbrot set within 10% accuracy. How then can we use the computer to zoom in on small pieces of the boundary that are no larger than an hydrogen atom? Are those really pictures of the Mandelbrot set, or are they intricate shadows produced mainly by round-off error in our computer and a badly chosen threshold? So [...] the next time you see [...] pictures of the Mandelbrot set, with swirls and dots and dainty patterns, that claims to represent the fine detail of an amazingly complicated set, I hope you will admire the artistry [...] and question the mathematics.

It has been *proved* by Douady and Hubbard [13] that $\mathcal{M}$ is connected. But pictures can be misleading: it appears from them (see Figure 19) that the Mandelbrot set is *not* connected — that it consists of a main body with an infinite number of 'islands' nearby (and an infinite number of islands near each of these, and so on).

In some cases, however, results are optimistic and seeing can mean believing when the algorithms which generate the pictures corresponding to a dynamical system are carefully analyzed and what it *means* for a picture to be 'correct' is defined. It is shown in [20] that pictures corresponding to Julia sets of certain complex exponential functions are reliable. It is important that visual representation be supplemented with selected tables of numerical results wherever possible for error analysis and that the programs be equipped with tools which can scrutinize the data they produce.

- **Cognitive.** Visual results are prone to optical illusions. Just because something is displayed does not mean students see it or make sense of the properties they attribute to them. It takes cognitive processing to make sense of visual presentations. Lack of necessary geometrical background can lead to difficulty in understanding complex attractor images. A cognitive approach to visualization in design of a computer program could be: program is specially designed to enable user to manipulate generic examples of dynamical systems concepts — the learner is *directed* through a suitable sequence of activities with examples (and non-examples) *towards* the fundamental properties of the concept, where students see it develop graphically in *real-time*. These approaches — numerical/symbolical/visual — should be given a role beyond that of merely another representation of the problem; of being central objects from which information is processed.

- **Sociological.** There appears to be a difference between methods of processing information used by the teacher and what is taught — didactical transposition. There is

Figure 19: A section of the Mandelbrot set after 'Zooming-in'.

a *linear* presentation of knowledge but that is *not* the way knowledge is organized in the teacher's mind. Since prerequisites should be presented explicitly rather than implicitly, it necessarily allows for *sequential* presentations. The fundamental difference between a diagrammatic and sentential representation is that the diagrammatic representation preserves explicitly the information about the topological and geometric relations among the components of the problem, while the sentential representation does not. This leads to the difficulty understanding visual presentations [21]. The importance of the suggestions of Section 9.7 thus becomes more apparent.

9.10 Dynamical Systems and Sound

Mathematical study of music dates back to 5th century B.C. Greece, when the Pythagoreans formulated a scientific approach to music, expressing musical intervals for different scales as numeric proportions and drawing relationships between notes in specific scales.

It seems that a mathematical view towards music significant from a dynamical systems viewpoint was initiated by Russian music theorist Joseph Schillinger in 1920's. He created [62] rhythmic patterns in music by the interference of waves of different periodicities where the resulting wave would carry the effect of each of the original waves — an idea which was used intuitively by many classical composers. Shapes of such patterns turned out to be self-similar, a hallmark of fractal sets.

In the mid 1970's, instead of studying the *structure* of the music composed, Voss and Clarke [59] analyzed the actual audio physical sound of music that was played. They analyzed the audio signals on a PDP-11 computer and found that the presence of the $1/f$-power law behaviour in the spectral density of the signal was *generic*; it existed in completely different kinds of music — from Bach's *First Brandenburg Concerto* to Joplin's *Piano Rags*.

Since then there have been various computer–generated studies of music and its dynamical interpretation and vice versa; use of simple dynamical systems, such as the logistic equation as a source for notes in computer–generated music. For an example, see the composition *The Voyage of the Golah Iota, Wind, Sand, and Sea Voyages: An Application of Granular Synthesis and Chaos to Musical Composition* by Gary Lee Nelson, available on the WWW at the URL

`http://timara.con.oberlin.edu/~gnelson/papers/Gola/gola.htm.`

9.10.1 Educational Implications: Examples

Nonlinear Dynamics in Musical Instruments

The first–order linear theory of musical instruments, is remarkably successful in explaining their acoustic behaviour. For example, simple impulsively excited instruments such as guitars and bells have nearly linear behaviour, with all modes simply decaying exponentially with time.

Nonlinearity in several forms is essential to the operation of all sustained–tone musical instruments, and contributes greatly to the characteristic sounds of the more interesting percussion instruments. Nonlinear phenomena are essential for the production of harmonic sounds from musical instruments with sustained tone through the phenomenon of *mode locking*. See *Nonlinear Dynamics and Chaos in Musical Instruments*, by Neville H. Fletcher, available on the WWW at the URL

`http://www.csu.edu.au/ci/vol1/Neville.Fletcher/paper.html.`

Examples are the bowing mechanism of a violin and the reed driving a clarinet. Activities involving playing such instruments could demonstrate difference between linearity and nonlinearity in real–world situations. This experience can be brought to the classroom using the WWW using pre–recorded sound clips of these instruments. Examples of where the theory of chaotic systems can be applied to electroacoustic and instrumental composition are presented in *Composing with chaos; applications of a new science for Music*, by David Clark Little, available on the WWW at the URL

`http://www.xs4all.nl/~19521952/dcl/COMPwCHAOS.html.`

Fractals in Music

Using dynamical systems concepts such as fractals forms more than just the ideas for music composition. Hearing the way fractals sound, adds another dimension to how we perceive fractals, namely time. When we see a picture of a fractal that is created by an algorithm, often we are unable to see the *order* in which the points were generated. If we were to get a sense for what notes corresponded to what sections on the grid, we could remember the approximate order in which the points are generated by simply memorizing the melody which corresponds to the fractal. This could give us more insight into how these algorithms work. It is also possible that we may be able to perceive some properties held by fractals more easily by *hearing* them then *seeing* them[1]. Fractals are studied in ways which are very visual, such as by computer-generated graphics. Another possible use of such audio programs then could be aiding the visually impaired in learning about fractals. Use of sound in dynamical systems education is still in its infancy.

9.11 Music on the WWW

In the early days of the Internet, the only way to listen to audio was to download `.au` files from FTP sites. In 1994, with the introduction of the Internet Multicasting Service, the first 'broadcasting' on the Internet came into existence. There were two limitations to this approach:

- The (large) size of the files.
- The lack of streaming capability. An hour-long show encoded in `.au` format generates a file 14 megabytes in size. Consequently, only users with high-speed connections or a lot of patience would download these files. Even if one did have a high speed connection that could deliver enough throughput to listen to the broadcast in real time, one still had to wait for the whole file to download before one could start to listen to it.

All that changed in 1995 with the advent of RealAudio from Progressive Networks. Real Audio allows the delivery of audio over a 14.4 kilobit per second Internet connection. The means of communicating sound via the WWW are similar to other multimedia such as graphics and animation: CGI-script approach for server-side sounds in dynamical systems and browser-support applications approach for client-side sounds in dynamical systems.

Chua's Circuit on the WWW

Chua's circuit is one of the simplest electronic devices for which the presence of complex behaviour has been observed experimentally, verified by computer experiments and proven mathematically. It can be modeled by a set of three nonlinear ODEs

$$\begin{aligned} x' &= \alpha[y - h(x)], \\ y' &= x - y + z, \\ z' &= -\beta y, \end{aligned} \tag{4}$$

where $h(x) = m_1 x + \frac{1}{2}(m_0 - m_1)\{|\, x+1 \,| - |\, x-1 \,|\}$, and $\beta = 14.3$, $m_0 = -\frac{1}{7}$, $m_1 = \frac{2}{7}$.

Chua's circuit has been used to generate sound and music. With the help of the WWW, the musical experience of Chua's equations (4) can be brought into the classroom. See *Chua's Oscillator: Applications of Chaos to Sound and Music* on WWW at URL

`http://www.ccsr.uiuc.edu/People/gmk/Papers/ChuaSndRef.html`.

To do that, however, the bandwidth is of significance.

[1]Consider the difference between the sheet of music of a song and an audio recording of the song being played as it is written. Both the music sheet and the recording contain the same information, but in this case it is easier to hear, for example, the regularities in the notes and rhythms by *listening* to the recording rather than *looking* at the music sheet.

A list of resources relating dynamical systems and sound is available on the WWW at the URL

`http://indy.cs.concordia.ca/ds/sound/.`

9.12 Obstacles to Using Sound in Dynamical Systems Education

The present limitations of using sound in dynamical systems education are that the environment is not widespread due to the following reasons: there has been a lack of uniformity in hardware (which are mostly Macintoshes), software for computer-generated sound production is expensive, and until recently audio formats were not standardized. Now that there are various sound players available cheaply for a variety of platforms in form of WWW browser plug-ins or helper applications, this situation may improve in future.

Chapter 10

Dynamical Systems: Databases and Searching on the WWW

Teaching and learning requires the use of resources. Books alone, in dynamical systems study, are often not sufficient because of the nature of information (multimedia) that is generated. The WWW can be viewed as a library of resources. In that view, one educational use of the WWW is to exploit the enormous amount of information that is accessible. In this Chapter, we discuss how dynamical systems–related information be organized (in databases) and accessed (searched) efficiently.

10.1 Criteria for Evaluating an Educational WWW Site

The following criteria can be used for evaluating an educational WWW site for the purposes of creating a local database (Section 10.2.1) or while searching a WWW site (Section 10.3) for educational use:

- **Speed.** Is access to the site fast enough to be used in a classroom?
- **Content.** What is the level of content provided by the site? What type of media is used by the site? Are those supported by the browser being used?
- **Design.** Is the site well–designed to support easy navigation?
- **Access.** Is the information openly accessible or does the site restrict access to certain information?
- **Reliability.** How reliable is the information available? Who is the author? What is the affiliation?
- **Stability.** Is the purpose of the site to provide information for a long period of time or is it an experimental site?

10.2 Dynamical Systems on the WWW: Databases

Databases play a vital role in storing, searching and sharing information in an organized manner. Selected representative dynamical system equations, along with their properties (for easy reference) and corresponding phase and bifurcation diagrams (to facilitate comparison), could be efficiently stored in a database that can be queried. For this purpose, knowledge of taxonomy and phenomenology of dynamical systems can be useful.

Notable collections with dynamical systems–related resources are listed on the WWW at the URL

`http://indy.cs.concordia.ca/ds/sites/`.

10.2.1 Dynamical Systems Databases: Examples

Dynamical Systems URL Database

To facilitate the WWW 'journey', a database of URLs relevant to dynamical systems can be designed and implemented. Such collection may consist of annotated URLs of information related to dynamical systems that is of specific interest.

Advantages of creating such local databases are:

- **Accessibility.** External sites may be restricted or may become inaccessible.
- **Efficiency.** Accessing information available locally is often faster than an external access.
- **Accuracy.** Global search engines often yield irrelevant information. It is easier to obtain desired information it is is accessible locally.

An (object–oriented) URL database has been developed and is available on the WWW at the URL

`http://indy.cs.concordia.ca/ds/`.

Figure 20 illustrates the topics under which the database was organized. URLs among each topic have been categorized into those which are internal and those which are external to the site. There exist Perl scripts which can be used to maintain the database, such as updating URL addresses.

Dynamical Systems Dictionary Database

To be able to comprehend proofs, various definitions should be well understood. An HTML–based database of dynamical systems–related definitions can be quite useful. The main advantage of hypertext from a learner's perspective is that it is able to adapt to his/her needs (for example, by asking for complementary information and 'bookmarking'). HTML Frames can be quite useful here. Figure 21 shows a two–frame HTML document, with one window giving the definition of the concept and the other displaying corresponding multimedia (such as an image or an animation) illustrating the concept. In the definition of a

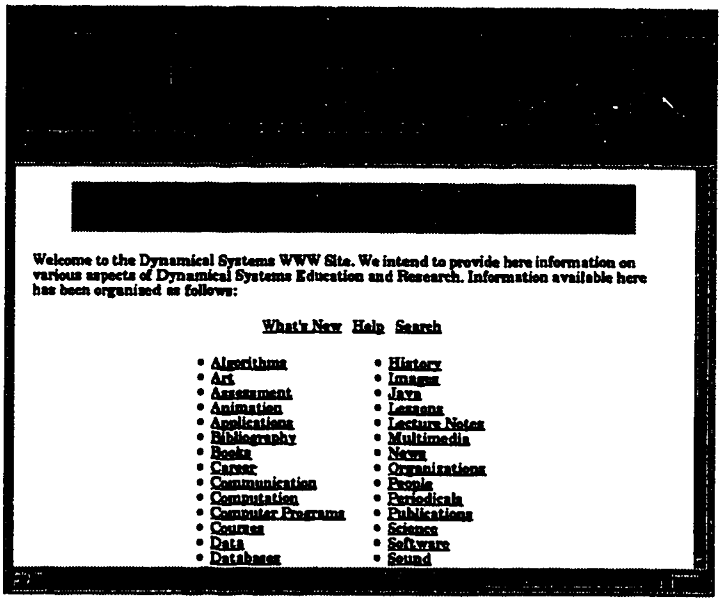


Figure 20: URL Database 'Home Page' on the Dynamical Systems WWW Site at the URL `http://indy.cs.concordia.ca/ds/`.

concept, there can be prerequisite material (other definitions). In mathematical hypermedia any object (definition, equation) can be hyperlinked. Such content on clicking the mouse can be displayed in a separate window (which can be closed by the user when not needed). An example of such hierarchy is attractor → orbit → iteration.

10.3 Dynamical Systems on the WWW: Searching

The WWW is one of the world's largest sources of publicly available information. It provides a myriad of information but still lacks in navigational aids. It is important for teachers and students alike to search relevant information efficiently and quickly on the WWW, particularly in a classroom.

To help us deal with incredible amount of data, a new skill is needed: WWW searching. Given a topic, anyone with a WWW browser and access to the Internet can search the WWW for information on that topic. Searching, however, is not the same thing as finding. There is little organization or consistency on the WWW. A variety of problems can occur while searching through the WWW:

- The search yields no results.
- The search yields too many (most of them irrelevant) results.
- The network connection is slow at the time when results are needed quickly.
- The WWW site on which the relevant information lies is inaccessible.

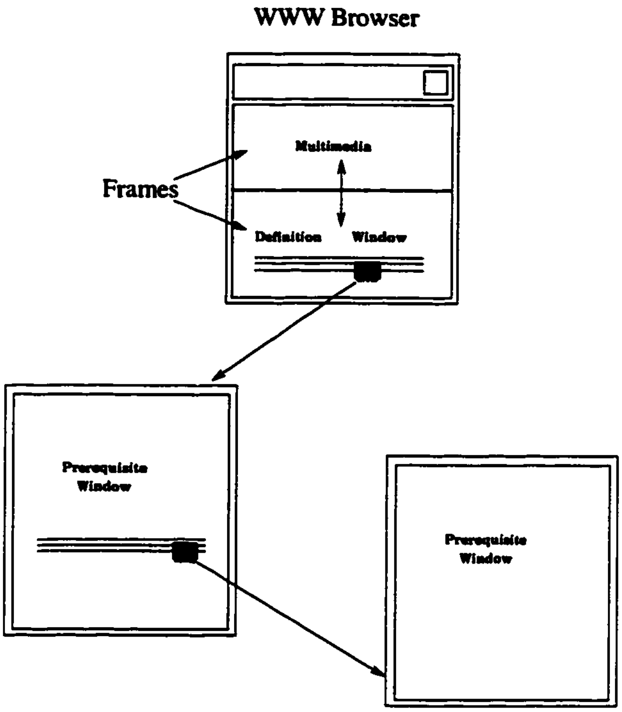


Figure 21: The HTML Document Structure of the Dynamical Systems Dictionary on the WWW.

- The search did yield a result, and the site location did have the relevant information but on a subsequent visit, the document was not found.

A successful search requires a search strategy, which depends on the search object, view of the Internet, search tool, query (if the search tool is a search engine) and search technique.

10.3.1 Views of the Internet

There are a number of possible views of the Internet: it is a large group of *computers*, it is a collection of *programs*, it is a collection of *resources*, it is a *library* it is a large *community*. Figure 22 illustrates these diverse views. A key to effective searching is to be able to *change* one's view of the Internet as needed [2].

We now describe each of these views, along with their advantages and disadvantages in searching.

- **Internet as a Group of Computers.** Every computer on the Internet has a unique name, for example, `indy.cs.concordia.ca`. The advantage of knowing some computer names during searching is speed. The disadvantage of this view is that one has to remember lot of computer names. Fortunately, the naming scheme is standardized (using DNS) and follows an intuitive pattern.

- **Internet as Collection of Programs.** Knowing certain Internet programs well such as an e–mail program (e.g., mh), a WWW browser (e.g., Netscape Navigator), a

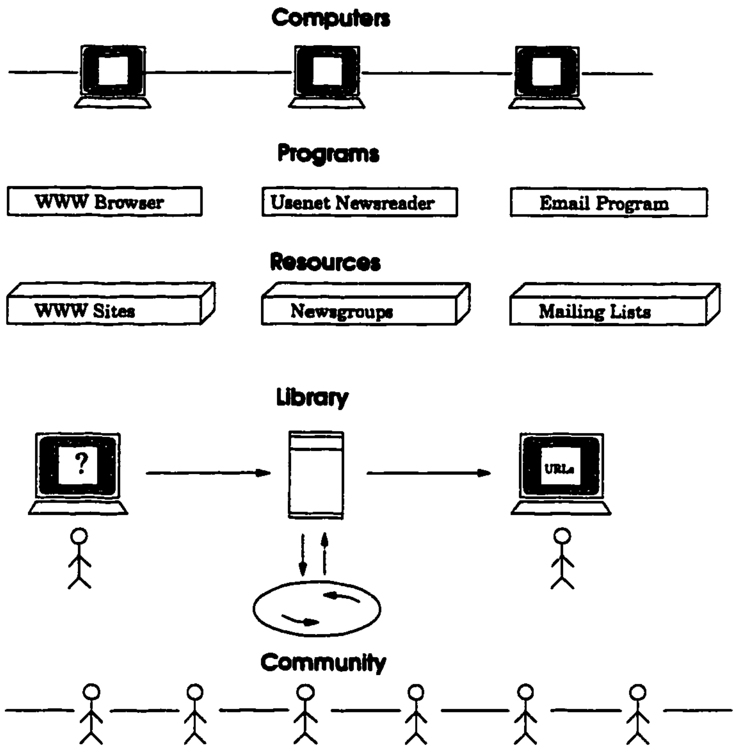


Figure 22: Views of the Internet.

WWW search engine (e.g., AltaVista), a Usenet newsreader (e.g., rn) can speed-up the search. The disadvantage of this view is that it ignores intuition — just because one knows every single feature of a WWW browser does not mean that the search will always be effective.

- **Internet as a Collection of Resources.** Internet could be viewed as "Yellow Pages" — a collection of resources such as WWW sites, newsgroups and mailing lists. The advantage of this is that resources on a topic one is trying to locate may already exist such as a collection of URLs: all one has to do is to look them up. The disadvantages of this view are that such Yellow Pages go out of date, they list only very few resources on each topic and are limited to searching one topic at a time — they are not useful for searching a *combination* of topics.

- **Internet as a Library.** Internet can viewed as a large collection of libraries world-wide. In this view of the Internet, a WWW document is a "book", a URL is the "call number" and a WWW search engine is a "librarian". The obstacles in this view are that each of these libraries, unfortunately, has its own method of organizing and accessing information, there is no master index of all libraries and there contents, there is no 'roadmap' to get from one library to another.

- **Internet as a Community.** The advantage of this view is that there are many knowledgeable people with access to the Internet (in particular, Usenet) who can answer one's question (atleast at an educational level) or help locate a resource. The disadvantage of this view is that such a facility can easily be misused. For example, students can post their assignment problems on the newsgroup and obtain solutions to them without the knowledge of the teacher. Also, asking too many 'general' questions can overwhelm a newsgroup.

10.3.2 Search Engines

Search engines are powerful tools for seaching the WWW. A search engine works as follows. The words given to search to a search engine are called *keywords*. A set of one or more keywords is a *query*. A search engine keeps a catalog of WWW pages and uses it to locate a query. If a search engine finds a WWW page in its catalog that matches a query, the matching page is called a *hit*.

There are three types of search engines: *active*, *passive* and *meta-search*. See Figure 23. An *active* search engine collects WWW page information by itself. An example is AltaVista. It uses a program called a *robot* that travels around the Internet, locates WWW pages and adds entries to the catalog. The advantages of active search engines are that they usually have large catalogs and are updated frequently (without human intervention). The disadvantages are that there are often too many hits, which are not very well organized. A *passive* search engine allows people to register their WWW pages. An example is Yahoo!. Once a page is registered with the search engine, the page can be found by queries. The advantage of a passive search engine is that they tend to be very organized. The disadvantages are that they usually have smaller catalog and items may be organized in unexpected (from the user's viewpoint) places. *Meta-search* engine is a search engine which uses several search engines simultaneously. An example is MetaCrawler. The advantages are that they save effort of searching multiple search engines. The disadvantages are that the search can be slow and that they may summarize the data in their own way or present only partial results from each search engine, possibly hiding relevant information.

Dynamical Systems Local Search Engines

We call a search engine *local* if it searches only the information local to a WWW site. Such a search facility for the Dynamical Systems WWW site at the URL

`http://indy.cs.concordia.ca/ds/`

has been developed. Relevant CGI scripts have been created for that purpose. It is expected that using the HTML form-based interface, teachers and students will be able to carry out search through it efficiently and quickly.

Dynamical Systems Global Search Engines

We call the search engines on the WWW which are not local, as *global*. Figure 24 shows the results of search for the query `fractals` on the WWW global search engine Yahoo!.

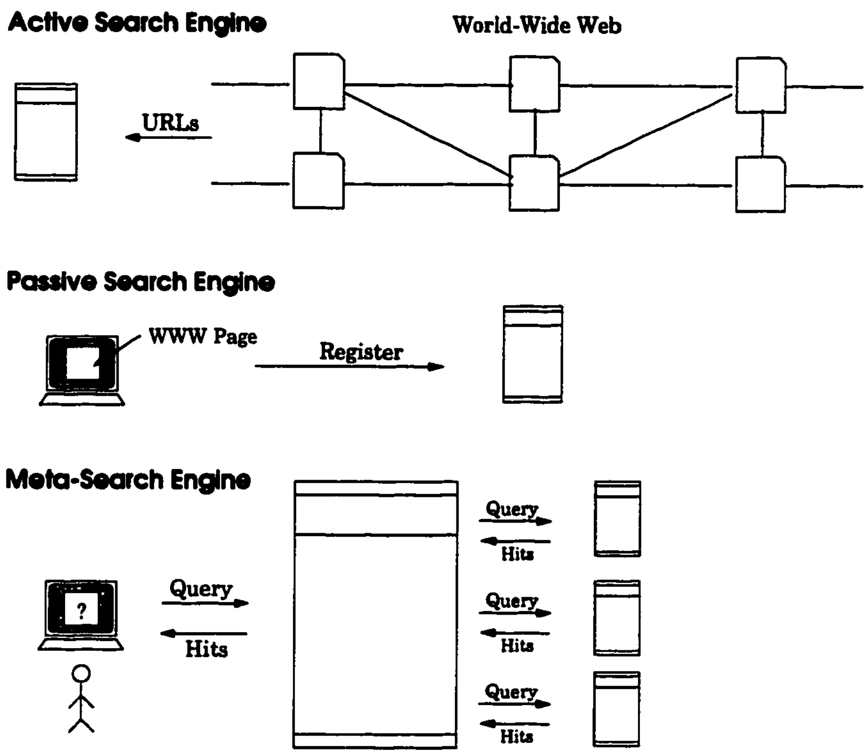


Figure 23: The Three Types of WWW Search Engines.

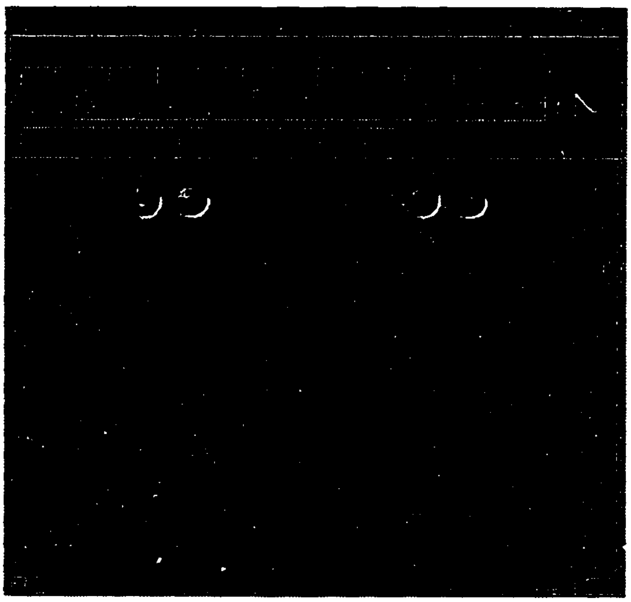


Figure 24: Results of Search of the query **fractals** on the WWW Search Engine Yahoo!.

Limitations of Search Engines

Inspite of the strength of search engines as a search tool, there are certain limitations that one should be aware of in their usage. These are: the way they sort and rate information — often irrelevant information is presented earlier in the list of hits, over–information, and their inability to distinguish word meanings.

10.3.3 Query

The success of WWW search depends heavily on the keywords chosen. It is important to know how are they treated by specific search engines. A *simple query* consists of just the keywords one wishes to search. Usually, simple queries can be too broad, resulting in a response with many unrelated hits. An *advanced query* consists of Boolean operators to include or exclude keywords from the search. The type of operators supported by a search engine define its *query language*. Different search engines usually have different query languages which can be found on their respective WWW sites.

10.3.4 Search Techniques

Using certain search techniques, searching time and effort and be greatly reduced. The techniques can be of the following type [2]:

- **General Search.** A general search uses the query operator `OR` to concatenate keywords, instructing the search engine to locate pages that contain *any* of the keywords. The more keywords one adds the *broader* the search, the *more* hits one is likely to get. But one is also likely to get many irrelevant hits as well.
- **Specific Search.** A general search uses the query operator `AND` to concatenate keywords, instructing the search engine to locate pages that contain *all* of the keywords. The more keywords one adds the *narrower* the search, the *fewer* hits one is likely to get. The chances of getting a hit are low but hits are likely to be relevant.
- **Incremental Search.** Incremental search allows one to narrow and broaden the search progressively until one 'zeroes–in' on the desired information. This process can, however, be time consuming.
- **Substring Search.** Substring search allows one to search for the occurence of a keyword (substring) in any word (string) in which it occurs. This process can simplify query but is likely to produce irrelevant hits.
- **Search–and–Jump.** Search–and–jump search allows one to search using the Find facility (such as the `Find` button in Netscape Navigator or the "/" command in Lynx) of browsers. One can choose a search engine, set number of hits to the maximum number possible, type the query and start the search, use the Find facility to find specific information in the WWW page returned. This process is faster than performing multiple quries but it can, however, be time consuming.

Object	Internet View	Search Tool	Query	Technique
Topic	Library	Search Engines	Advanced	Category
Document	Library	Search Engines	Advanced	Specific Search
Software	Resource	Usenet FAQ	Advanced	Search-and-Jump
People	Community	Search Engines, Usenet	Advanced	Specific Search

Table 3: Dynamical Systems-Related WWW Search Strategies.

- **Category Search.** This allows one to search using the categorized list of topics of subject guides (such as Yahoo!, Lycos). No query is required. The limitation here is that the list may not be organized in a manner one might expect.
- **Search and Rank.** Some search engines (such as AltaVista) search and rank the results. Relevant hits are listed first, however, effective ranking functions are still not known.

Table 10.3.4 summarizes the search strategies for some dynamical systems-related (search) objects. In addition to that, a list of currently available dynamical systems software can be obtained from the FAQ of the Usenet newsgroup `sci.nonlinear`, WWW search engines can locate people's 'home pages', Usenet search engines can locate participants in the newsgroups, and directories such as Combined Membership List (CML) of AMS/MAA/SIAM can also locate people. A list is also available on the WWW at the URL

`http://indy.cs.concordia.ca/ds/people/`.

Chapter 11

Dynamical Systems: Communication on the WWW

If you don't network, you don't work.

— Michael Dertouzos

Communication, verbal or written, between the teacher and the student (and even among students) is important in the learning process. The WWW provides an ideal medium for communication among students and between students and the teacher on several levels. With the help of WWW-based tools such interaction between students and teachers becomes very feasible. In this Chapter, we shall discuss the tools for asynchronous communication (e–mail, Usenet newsgroups, electronic bulletin board system) and synchronous communication (video–conferencing, chat rooms). Apart from communication, these tools can provide improved course coordination in various ways.

11.1 Noninteractive Communication

11.1.1 Electronic Mail

Electronic Mail (or e–mail) is system which uses the Internet to send and receive mail. A WWW browser can be used as an interface for sending and receiving e–mail. Figure 25 shows the Netscape Navigator's interface to the e–mail program.

Students can contact the teacher in a personal way using e–mail. At designated places on the course WWW documents they can find buttons to send e–mail to the teacher. This feature can be used by teachers to give the students personal assistance, to receive their solutions to given problems and to give them personal feedback. The possibility of attaching WWW documents with e–mail is especially useful, since it allows the student to emphasize specific details (by using colour and special fonts) and include (non–ASCII) documents such as image files complementing usual (ASCII) text. Students can use e–mail to communicate

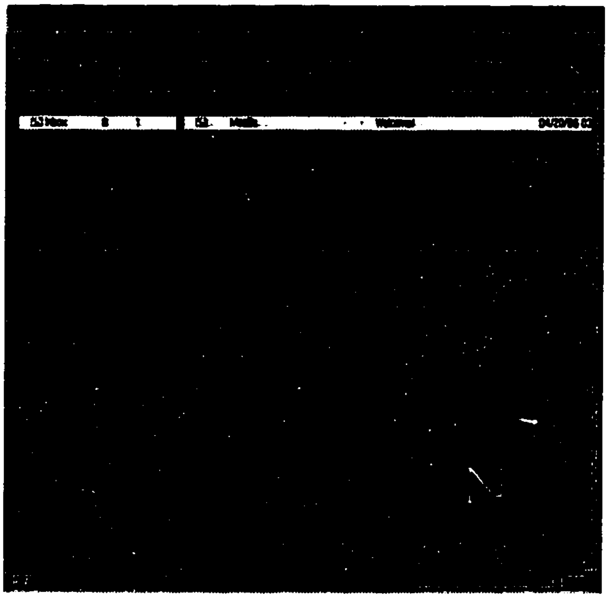


Figure 25: Netscape Navigator Mail Interface.

with other students and with other educators. E–mail messages can be sent to many people simultaneously, which can be useful in making announcements.

11.1.2 Mailing Lists

Mailing lists provide a means for the teachers and students interested in specific topics to communicate with others who share those interests. Anyone with an e–mail address can use a mailing list. One subscribes to a mailing list by sending a subscribe request to the site that runs the mailing list server. Once subscribed, messages sent to the list are distributed to the subscribers via e–mail. This is done by a program (such as listserv or majordomo) administering the mailing list.

Useful dynamical systems–related mailing lists are listed on the WWW at the URL

`http://indy.cs.concordia.ca/ds/comm/`.

A list of mailing lists related to mathematics education is given in [40, page 68].

11.1.3 Usenet Newsgroups

Usenet is an informal group of systems (called as Newsgroups) on the Internet that exchange "news". Usenet newsgroups make it possible to send a message many people can read and respond to. After subscribing to a newsgroup, it can be read using a newsreader program. A WWW browser can be used as an interface to this newsreader. Figure 26 shows the Netscape Navigator's news interface.

Almost all newsgroups have a list of Frequently Asked Questions (FAQs). They are very useful for obtaining answers to basic questions on the topic of the newsgroup. FAQs are routinely updated and posted on the respective newsgroup.

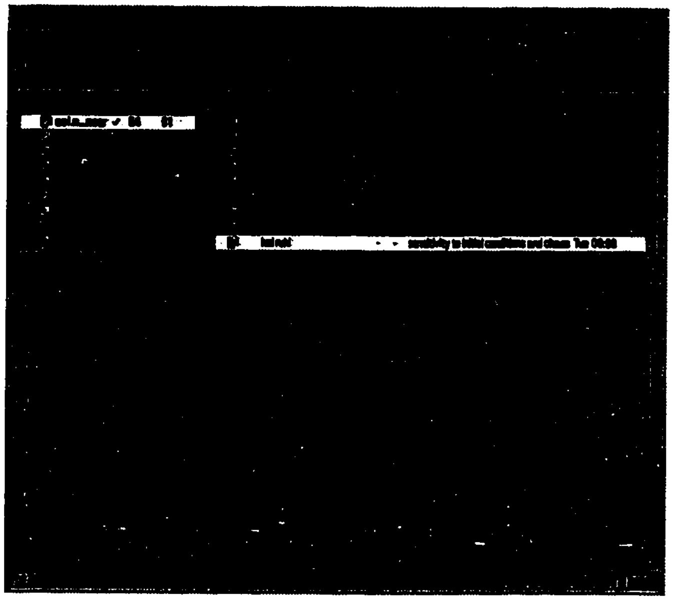


Figure 26: Netscape Navigator News Interface.

A list of dynamical systems–related newsgroups is available on the WWW at the URL
`http://indy.cs.concordia.ca/ds/comm/`.

Local newsgroups can be created within an educational institution for the purpose of supplementing term courses. Unlike the Usenet newsgroups, such newsgroups serve a fixed community, are usually restricted, and expire at the end of the term. They can be used for providing hints/solutions to exercises to large number of interested students simultaneously. Students can use local newsgroups to share questions and ideas. This encourages group participation.

11.1.4 Bulletin Board Systems

Local bulletin board system can be used by teachers for posting notices and making general announcements. Students can use them to share questions and ideas. For each hypermedia course, a bulletin board can be set–up. To avoid server overload, access should be made to a limited number of users at a given time.

11.1.5 Noninteractive Communication Tools: A Comparison

The disadvantage of a mailing list is that one gets e–mail messages one is not interested in. The disadvantage of using a newsgroup is that one has to search for topics of interest (which could done using a Usenet search engine DejaNews available on the WWW at the URL
`http://www.dejanews.com/`
which searches for archived newsgroup messages). Recent newsgroup messages are available on the WWW for a certain time period; however mailing list messages are not archived.

11.2 Interactive Communication

Real-time communication is a vital part of education. UNIX "talk" facility, IRC (inter relay chat) and MUD (Multiple User Dimension) are forms of real-time text-based communication tools that have been around the Internet since early 1980's. Recently the WWW has provided several new forms of real-time communications that may lead to even better real-time tools.

11.2.1 Video-Teleconferencing

Students and their teacher can use video-teleconferencing for their meetings and discussions. Students can use video-conferencing to communicate with other students and with other educators. Such an interaction is relatively fast and more personal. To be used in conjunction with a WWW browser, this needs appropriate software, for example, CoolTalk with Netscape Navigator, on each client.

11.2.2 Chat Rooms

Chat rooms were the first means of interactive communication on the Internet. They could be considered as a real-time bulletin board systems. This facility can be brought to the WWW, for example, with Java in form of a graphical chat applet on specialized topics. Chat sessions can be quite useful in engaging quick discussions world-wide. The advantage here is that one needs only a Java-compliant WWW browser; there is no need for any special software on the client-side.

Chapter 12

Dynamical Systems: Assessment on the WWW

A computer-based learning environment should provide feedback to the students. Interactive WWW-based exercises form such an activity.

12.1 Advantages of Using WWW as a Tool for Assessment

There are following advantages of a WWW-based assessment:

- **Asynchronous Assessment.** Students can test their learning in their own time.
- **Range of Problems.** Broad range of problems can be designed owing to multimedia.
- **Type of Problems.** The WWW provides a suitable environment for designing multiple-choice question quizzes and 'laboratory' exercises where the students run a dynamical systems-related program and returns their conclusions on the WWW browser.

12.2 Interactive Exercises on the WWW

WWW-based interactive exercises can be useful because the response can be immediate. Here the use of HTML Frames could be appropriate: a two-frame structure, with one window panel providing the questions and the other the answers.

12.3 Examples

12.3.1 CGI Approach: Server-Side Processing of Responses

The server-side of interactive exercises (or more generally interactive documents) can be implemented as CGI scripts [29, page 163]. WWW browsers, with appropriate HTML forms/buttons, provide an interface for the exercises. The interaction that takes place is as follows. (See also Figure 1.) In a typical interactive exercise the computer poses a question with multiple choices to choose from (with a radio button next to each of them) and the student responds to the question by choosing a (radio) button and checking the `Submit` button. Once the response (i.e., an HTTP request) is submitted, it is passed on to and checked by the CGI script (by matching the response in its solution database) on the server-side. The answer is assessed and a message (an appropriate HTML or other type of document describing whether the answer is correct or what went wrong) is returned.

In certain exercises, there might be several requests which form a session. This entails that for each session certain state information has to be maintained and special care should be taken to support more than one session at a time. Since, HTTP is stateless, there is no direct support (at the HTTP level) for interactive sessions which consist of several HTTP requests. This could be done by designing appropriate JavaScript cookies.

If the CGI script is being used to call an external program, such as Maple, then certain text formatting capabilities are also necessary. In particular, to present mathematical formulas in the exercises we can use a 2D HTML form, made up of several preformatted lines of characters, which we surround by ASCII characters to make it look like normal Maple syntax.

12.3.2 Java Applet/JavaScript Script Approach: Client–Side Processing of Responses

Java applet or a JavaScript script can be designed to form simple quiz questions that can be processed on the client-side. JavaScript scripts are less useful in this case as the answer to the questions need to be included in the code itself which is embedded in the HTML source of the document and is accessible to the user. Figure 27 shows an HTML Frame document with an example of an interactive quiz.

Some such WWW-based dynamical systems-related exercises are available on the WWW at the URL

`http://indy.cs.concordia.ca/ds/assess/`.

12.4 Noninteractive Exercises on the WWW

There are exercises which can be designed where the student submits the answer on the WWW browser, but does not obtain the response immediately; rather the response is passed on to the teacher for assessment.

12.4.1 CGI Approach

In this approach, a CGI script can be designed as follows. The student carries out a specified computation and records necessary observations. Once completed, he/she accesses the 'response page' and submits his/her name, ID# and conclusions from the experiment using a 2D HTML form. If any of the entries are empty, the form is returned (with an

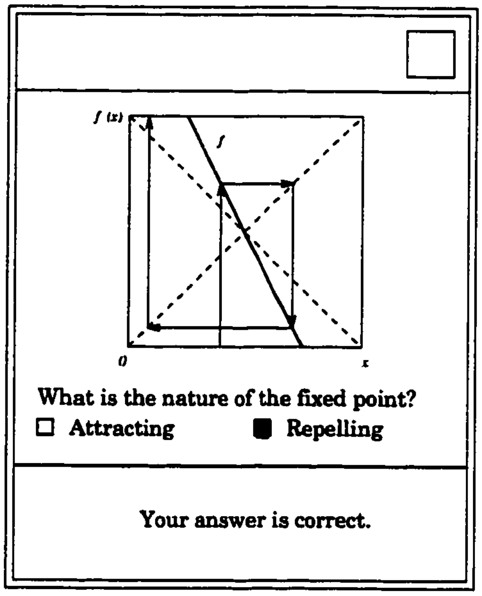


Figure 27: An Example of an Interactive Quiz with HTML Frames.

error message) to be completed. The information submitted by the student is attached (concatenated) with a 'time stamp' to an HTML file. This file is accessible only by the teacher. Such an example is illustrated in Figure 28.

12.5 Scope and Limitations

Within the realms of a classroom, such exercises could be used for actual examinations. However, certain factors need to be taken into consideration: access speed, (if used) quality of graphics, etc. Outside the classroom, such exercises can be quite useful for personal assessment. If used for an actual examination *outside* the classroom (without the supervision of the teacher), there are possibilities that may sacrifice fairness: students obtaining external assistance, getting to know the problems beforehand. This requires special design considerations, particularly involving WWW security. Use of JavaScript cookies can be made which can restrict the number of times a WWW page has been accessed, impose limitations on total access time, and obtain information about the user.

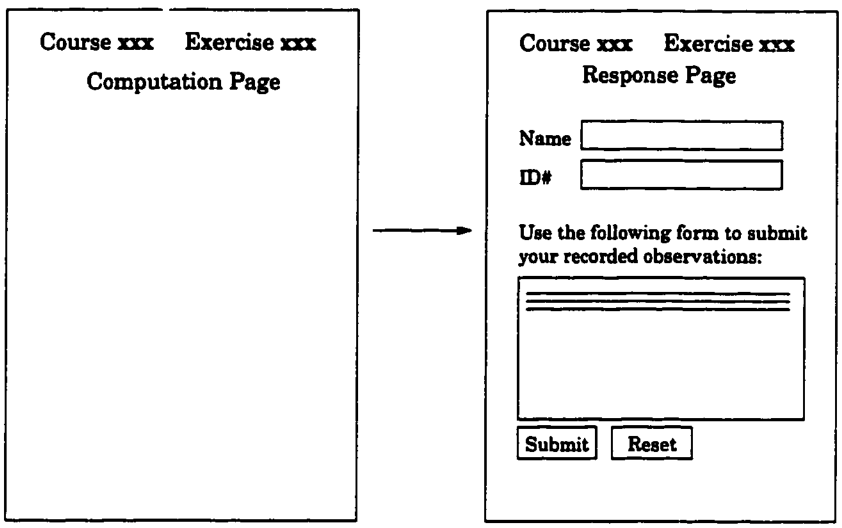


Figure 28: An Example of a CGI–Based Computational Exercise.

Part III

Dynamical Systems Educational Environment on the WWW: Applications

Chapter 13

Teaching Mathematics from a Dynamical Systems Viewpoint

Reform of undergraduate mathematics is the key to revitalizing mathematics education.

— NCTM

It has been emphasized recently [50] that the students should be introduced to simple nonlinear dynamical systems early in their mathematical education. This would greatly enrich the students' intuition about nonlinearity and make them realize that simple nonlinear systems do not necessarily possess simple dynamical properties.

One problem that is faced during curriculum development to incorporate new courses is *saturation* — curriculums at pre-university and university undergraduate levels are already crowded. Instead of replacing the traditional topics, the study of dynamical systems gives *new* approaches to develop fundamental concepts, thus strengthening the traditional curriculum while enriching it.

How can traditional topics in mathematics be taught from a dynamical systems viewpoint? What is the role of WWW in it?

In this Chapter, we take topics from Number Theory, Calculus, Geometry, Differential Equations, Numerical Analysis, Complex Analysis, and Probability, and suggest how they could be taught from a dynamical systems viewpoint. We also discuss how the environment of WWW can assist in teaching and learning these topics. Instances where these topics are intrinsically interrelated are also pointed out.

13.1 Number Theory

There is an inherent relationship between basic number–theoretic concepts such as Pascal's triangle or Fibonacci numbers and discrete dynamical systems. In this section, we explore this connection with the help of the WWW.

13.1.1 Pascal's Triangle, Cellular Automata and Fractals

Arrays of numbers defined by simple rules can give rise to fascinating geometrical patterns. Consider the following construction given by the following steps:

1. Start with the number 1, and make it the apex of what will become a triangle of numbers.

2. In the second row, write two 1s.

3. For each subsequent line, add together adjacent numbers of the previous row and write the sums in the new row. Then place 1s at both ends of the line.

Here is what we get for the first eight rows:

```
              1
            1   1
          1   2   1
        1   3   3   1
      1   4   6   4   1
    1   5  10  10   5   1
  1   6  15  20  15   6   1
1   7  21  35  35  21   7   1
```

This array of numbers is widely known as *Pascal's triangle*.

The most important mathematical interpretation of Pascal's triangle is the rows of the triangle represent binomial coefficients, i.e., the coefficients of the polynomials:

$$(1+x)^n = a_0^{(n)} + a_1^{(n)}x + \cdots + a_n^{(n)}x^n, \qquad n = 0, 1, 2, \ldots \tag{5}$$

where

$$a_k^{(n)} = C(n,k) = \frac{n!}{(n-k)!k!}, \qquad 0 \le k \le n \tag{6}$$

Thus, one approach to the patterns in Pascal's triangle could be to understand the divisibility property of binomial coefficients. However, computing the numbers a_n in equation (6) does not lead very far since the factorials grow very rapidly as n increases. Using the recursion relation,

$$C(n+1,k) = C(n,k-1) + C(n,k) \tag{7}$$

avoids the computation of large factorials as well as for computing actual numerical values of the binomial coefficients when testing for divisibility — for example, $C(n+1,k)$ is odd provided $C(n,k-1)$ is odd and $C(n,k)$ is even, and vice versa.

The numbers along the diagonals follow certain patterns. For example, the second diagonal running from 1 to 7, consists of consecutive whole numbers. The numbers along the third diagonal are known as *triangular numbers*.

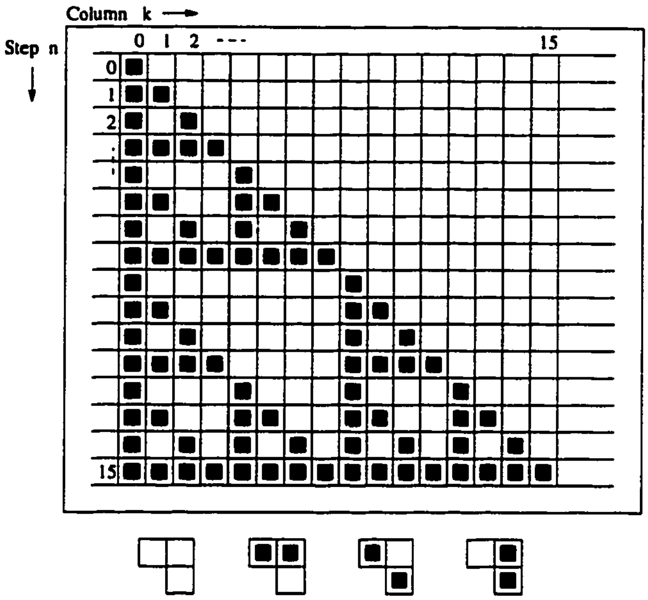


Figure 29: Pascal's Triangle in a Cellular Automata Evolution.

It is possible to convert this triangle into geometric forms. For example, one can replace the odd coefficients with 1 and even coefficients with 0 to get the following array in the above construction:

1

1 1

1 0 1

1 1 1 1

1 0 0 0 1

1 1 0 0 1 1

1 0 1 0 1 0 1

1 1 1 1 1 1 1 1

Continuing the pattern for many more rows results in a collection of triangles, of varying size, within the initial triangle. In fact, the pattern represents as a fractal — the even coefficients occupy triangles much like the 'holes' in the Sierpinski gasket. In other words, the pattern inside any triangle of 1s is similar in design to that of any subtriangle of 1s, though larger in size.

The above procedure allows the generation of a geometrical pattern but *does not* explain *why* do we begin to see the Sierpinski gasket when colouring the odd entries in Pascal's triangle. This has an inherent connection with one–dimensional cellular automata and modulo 2 arithmetic.

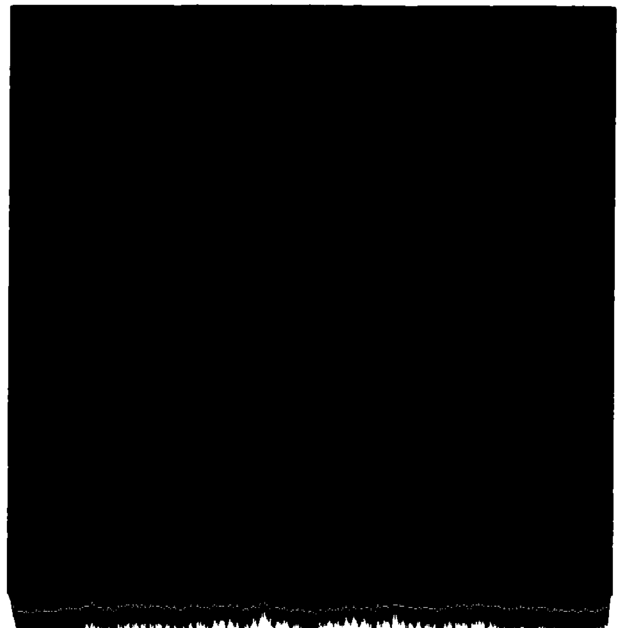


Figure 30: Pascal's Triangle Modulo 2 (32 rows).

Let $a_k(n)$ denote the state of cell number k of the nth–layer of a one–dimensional cellular automata. Starting with $a_0(0) = 1$ and $a_k(0) = 0$ for $k \neq 0$, the rule

$$a_k(n) = a_{k-1}(n-1) + a_k(n-1)$$

generates the coefficients of $(1+x)^n$ and is just an alternate way of expressing equation (7). Now, lets look at the divisibility properties of $a_k(n)$ with respect to an integer p. To test for even or odd binomial coefficients $C(n,k)$ or cells $a_k(n)$ is simply to check whether $a_k(n) \equiv 0 (\mathrm{mod} 2)$ or $a_k(n) \equiv 1 (\mathrm{mod} 2)$ respectively. Figure 29 illustrates this cellular automata evolution.

Pascal' Triangle on the WWW

Using the *Pascal's Triangle Interface* on the WWW at the URL
`http://www.cecm.sfu.ca/organics/papers/granville/support/pascalform.html`, a CGI script, one can specify the number of rows (up to 100), the modulus (from 2 to 16), and the image size to get a colorful rendering of the requested HTML form. It can be a useful learning experience where students can 'visualize' binomial coefficients and explore the fractal side of Pascal's triangle. Figure 30 shows Pascal's triangle using 2 as the modulus for 32 rows.

If we extend Pascal's triangle to infinitely many rows, and reduce the scale of our picture in half each time that we *double* the number of rows, then the resulting design is *self-similar.* The pattern becomes more evident if the numbers are put in cells and the cells coloured, say black or white, according to whether the number is 1 or 0, respectively. Figure 31 shows Pascal's triangle using 2 as the modulus for 64 rows. Such an approach to Pascal's triangle is taken in [77]; and many examples of self–similarity have been investigated in [48].

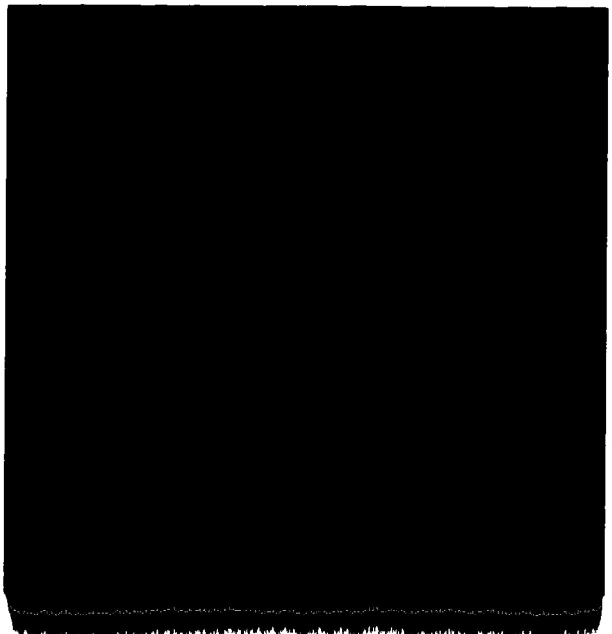


Figure 31: Pascal's Triangle Modulo 2 (64 rows).

Similar, though more complicated designs appear if one replaces each number of the triangle with the remainder after dividing that number by 3. Thus, one gets:

```
              1
            1   1
          1   2   1
        1   0   0   1
      1   1   0   1   1
    1   2   1   1   2   1
  1   0   0   2   0   0   1
1   1   0   2   2   0   1   1
```

This time, one would need three different colors to reveal the patterns of triangles embedded in the array. We can also use other prime numbers as the divisor (or modulus), again writing down only the remainders in each position. Figure 32 shows an example using 3 as the modulus.

A *Pascal's triangle Maple interface*, a CGI script which uses Maple as an external program to carry out computations and produce 2D and 3D graphics, is available on the WWW at the URL

http://www.cecm.sfu.ca/organics/papers/granville/support/pasform.html.

A glimpse of Pascal's biography can be shown in the classroom; it is available on the WWW at the URL

http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Pascal.html.

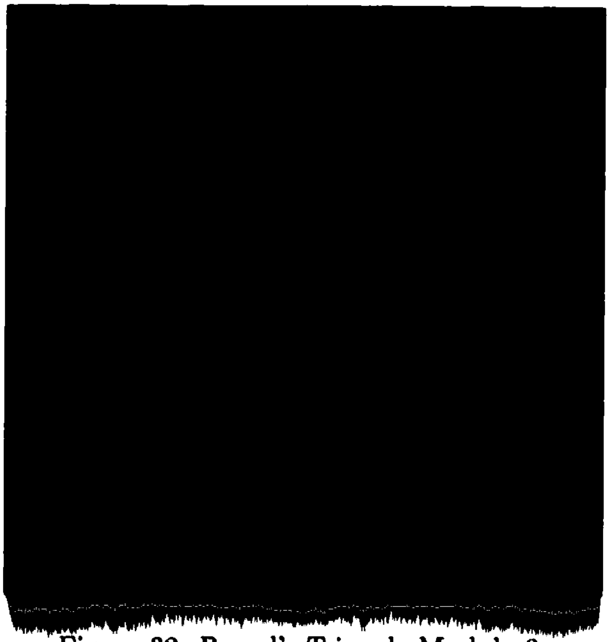


Figure 32: Pascal's Triangle Modulo 3.

13.2 Calculus

Calculus is one of the areas of mathematics that forms a mandatory part of most mathematics curricula and remains the archetype of higher mathematics. The quality of calculus instruction is a measure of mathematics education [11, page 52]. Among the courses in need for revision and reform, calculus courses are of utmost importance. The study of dynamical systems offers students an opportunity to use and build upon their knowledge of calculus and at the same time to see in a concrete setting many of the important topics from basic analysis — thus it serves developing a bridge between the two.

13.2.1 Limit and Iteration of Functions

The study of calculus often begins with an introduction of the concept of a function. Iteration and graphical analysis fits naturally into a discussion of functions and graphing techniques. Simple examples of iteration of functions can be shown to the student when the concept of composition of functions is introduced. Where an algebraic definition of infinity may seem rather abstract (and may lack motivation), a geometrical viewpoint can be taken and simple 2D fractals such as Sierpinski's triangle that arises as an iterated function system (IFS), can be introduced to teach the concept. Concepts of arithmetic and geometric sequences and series, and of their limits arise while studying the asymptotic behaviour of iteration of linear functions. It might help students develop notion of limit of a sequence better with the introduction of attractive fixed points of functions. We pursue this in Chapter 14.

13.2.2 Continuity and Shift Map

When the notion of continuity is being taught, students often have the impression that a function is continuous if its graph doesn't 'break'. Shift map is an example of a continuous function whose continuity is not as intuitively obvious. So while introducing the notion of continuity, the shift map can be used to illustrate the importance of the '$\epsilon - \delta$ definition'. In the process of showing the continuity of the shift map, there will be a need to define a suitable 'distance' (metric) between two points in the domain S (which is also the range) of the shift map which is a space of sequences: $S = \{s_0s_1s_2\ldots \mid s_i = 0 \text{ or } 1, i = 0, 1, 2, \ldots\}$ — the usual (here the 'absolute value') distance is not sufficient in this case. This can help students view the notion of distance in more than one way and help them realize the *need* for abstract metric spaces. Students who have a prior analysis background begin to appreciate the necessity of such an abstraction.

Teaching shift map has other advantages. Often, during the analysis of a one-dimensional maps (such as the tent map) as a dynamical system, its topologically equivalent (for a large set of parameters) shift map is studied instead since the dynamics of the shift map are more easily understood [75]. The shift map is of immense importance in the study of *symbolic dynamics* and provides a good base for encounter in more advanced topics such as Markov partitions.

13.2.3 Derivative and Fractals

Mandelbrot recommends [1] the introduction of fractals in teaching the concept of derivative:

> The principal place where fractals should be introduced is with the presentation of the derivative. I have the very strong feeling that many [students] would understand the derivative better if they are told at the very outset that it does not have to exist. [Since] these things are essential in natural science, [it is] easy to tell a student that, for example, if you take the motion of a particle along a coastline there is no tangent, there is no derivative.

By use of computer-graphics, the abstract notion (as a symbolic expression) of continuous non-differentiable functions can easily be introduced.

As *applications* of basic concepts from calculus, some fundamental results from dynamical systems can be shown. This can give the idea to the students that powerful results in mathematics can be established using very basic tools and can be accessible at an elementary level.

13.3 Geometry

Teaching and learning geometry is important for two main reasons:

- Geometry is the study of mathematical patterns in form of shapes. Some shapes are visual like snowflakes, others are highly abstract like 4-dimensional manifolds. The role that geometry plays, for example, in dynamical systems is reflected from the fact that the 3 volumes by Abraham and Shaw[1] are dedicated entirely to illustrations of geometric forms arising in evolution of various dynamical systems. For students to understand the geometrical shapes that arise in dynamical systems study, a solid background in geometry is necessary.

- To use and understand computer-graphical results in mathematics. The following example [63] can be illuminating. Galileo's discovery of mountains and craters on the moon helped change the way the universe is viewed. Thomas Harriot was an English astronomer who had been looking through the telescope at the moon at the same time that Galileo made his discovery. Harriot's sketches show, however, that "strange spottedness of the moon" did not look like mountains and craters to him. How could it have happenened that Galileo and Harriot looking at the same object through the similar telescopes, did not "see" the same thing? It has been suggested in [63] that, "Unlike Harriot, Galileo was a trained artist, skilled in the use of perspective and chiaroscuro, the rendering of light and shadow."

  The educated "beholder's share" is just as essential today to make sense of computer–generated graphics which are meaningful only if the viewer has *prior* experience with geometric structures. Therefore, the students should be taught the geometry, particularly coordinate geometry *underlying* computer graphics.

The role of geometry is a perennial issue in mathematics education at all levels from elementary school to graduate school. NCTM 1987 yearbook *Learning and Teaching Geometry* suggests some of the many questions involved. One of them is that geometry related courses are *compromises* between many different goals. What is needed is not just a better compromise for geometry, but a new and coherent mathematics curriculum that *integrates* it.

Such an amendment, however, will have to address many important pedagogical questions: What geometrical concepts should be taught? Will teaching traditional topics from a dynamical systems viewpoint enhance learning? What teaching and learning environments are needed?

In the study of shapes we need to teach how to discover similarities and differences among objects, to analyze the components of form, and to recognize shapes in different representations. The principal tools to do that are:

- **Identification and Classification.** This can done by properties of congruence, self–similarity, combinatorics and topology.

- **Analysis.** Besides recognizing similarities and differences in patterns, we also need to analyze them. Some of the criteria are: symmetry, lattices, combinatorics, dissection (dividing a region into compartment of various shapes).

- **Representation and Visualization.** The purpose is to understand scale models, shadows, sections and projections, reconstruct shapes from their images, to draw

[1] R. H. Abraham, C. D. Shaw, The Geometry of Behavior, Volume I, II, III, Aerial Press, 1980.

accurately, to use computer graphics. An extensive classification of representations is given in [24].

Examples from dynamical systems can be used in teaching these tools: Koch snowflake (congruence), Mandelbrot set (self-similarity, topology), cellular automata (symmetry), coupled map lattices (lattices), partitions in numerical solutions of differential equations (dissection), etc. Furthermore, as seen in Chapter 9, these examples can be brought to life using the environment of the WWW.

13.4 Differential Equations

Dynamical systems in form of differential equations had their inception in the time of Newton, when he formulated the basic laws of mechanics.

13.4.1 Teaching Qualitative Theory of Differential Equations

Until almost a century ago, dynamical systems studies mainly concentrated on describing various physical processes via differential equations and finding *solutions* to those equations. At the beginning of this century there came a turning point in these studies, when Poincaré devised methods to analyze long-term solution *behaviour* of dynamical systems.

Often, many courses on differential equations appear as a collection of algebraic methods for finding explicit solutions, thus failing to convey both the spirit and reality of the subject in its current state.

Even for 'simple looking' differential equations, often obtaining explicit expression of formulas for solutions is an arduous task — and even if we can find the solution (with respect to a fixed value of a parameter), that solution may not say much about the *overall* behaviour of the differential equation — which is our ultimate goal. This approach also obscures the central question: how do solutions *behave*? Therefore, instead of looking for explicit formulas for solutions, the approach should be a qualitative study of the differential equation.

Qualitative methods which involve graphing the field of slopes, enable one to draw approximate solutions following the slopes and study these solutions *all* at once. These methods may give a rough graph of the behaviour of solutions more quickly than the traditional methods of finding finding formulas, particularly the asymptotic behaviour. Also, these methods can yield numerical information about the solution such as location of singularities, asymptotes and zeroes.

To understand basic ideas of qualitative theory, it is necessary to include concepts such as equilibrium points and their stability, singular points, bifurcation of equilibria, etc — a topic in elementary bifurcation theory. This topic is rarely included in traditional differential equations courses, yet it is of crucial importance in many scientific and engineering applications.

Visualization in Differential Equations

Computer graphics has revolutionized how we 'see' solutions of differential equations —

as graphs, as phase and bifurcation diagrams, animations, and as time series. The use of present technology of computer graphics on powerful computers can be of significant assistance in qualitative approach [36]. But experimental insight gained from visualization should be turned into *statements*, not equations, about mathematics.

Differential Equations and Discrete Dynamical Systems

Poincaré was the first mathematician who established and exploited the connection between iteration and differential equations. By iteration of the *first return map* on a *Poincaré section* of the differential equation, if one exists, can give a lot of useful qualitative information. This continuous–discrete dichotomy suggests one way in which iteration of functions is related to differential equations. Introduction of discrete dynamical systems can provide a good foundation for a course in differential equations [61]. That could also be useful towards computer–based study of differential equations as, for example, numerical integration of the differential equation is the iteration of the appropriate map.

With the intention of communicating some of this excitement to undergraduates, new courses on ordinary differential equations [39] as well as reform of the existing ones [6], has been suggested. A consortium focussing on reform of existing curriculum of ordinary differential equations has been established. For its activities, see Boston University Differential Equations Project which can be accessed on the WWW via the URL

`http://math.bu.edu/odes/`.

Differential Equations on the WWW

Differential equations–related resources are available on the WWW at the URL

`http://indy.cs.concordia.ca/ds/topics/odes/`.

Differential Equations CGI

A Lorenz equations CGI is available on the WWW at the URL

`http://www.geog.ubc.ca/numeric/labs/lab6/cgi-bin/lorentz.cgi`.

It calls Maple as an external program to generate the graphics.

Differential Equations Java Applet

Java applets for Lorenz equations are available on the WWW at the URL

`http://indy.cs.concordia.ca/www/java/ds/`.

13.5 Numerical Analysis

13.5.1 Newton's Method

There are various numerical methods, which are based on iteration and can be expressed as discrete dynamical systems. One such method is the Newton's method for root–finding. The formula for Newton's method is given by:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$n = 0, 1, 2, \ldots$. It begins with an initial guess x_0, and by *iteration*, we get *improved* estimates $x_1, x_2, \ldots$ of the initial guess to the root x^*. The iteration is repeated till we reach a desired accuracy. The procedure is shown in Figure 33. The question from a dynamical viewpoint

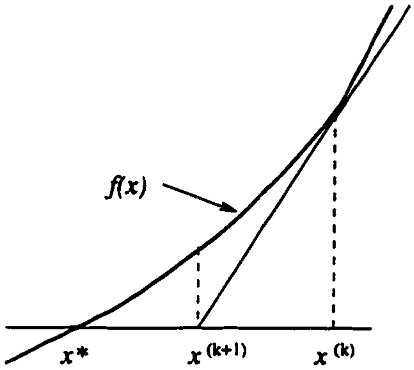


Figure 33: Newton's Method Illustrated.

then is: How does the choice of initial guess determine the future behaviour of the process? This leads to the concept of the ***basin of attraction*** of each root — the set of starting points from which Newton's method converges to that root. In the complex plane, these basins are often fractals.

Newton's method for $x^2 + 1 = 0$ in search for i is a good example of seeing how mathematical connections arise between numerical analysis, complex analysis and iteration of functions. It is an example of iteration, divergence, periodicity and chaos [69].

Even though the Newton's method has a long history as a numerical method [79], a 'chaotic side' of it has been revealed only in the 15 years. In 1879, British mathematician Arthur Cayley studied Newton's method to find roots of a one-parameter family of cubic polynomials after varying the parameter but abandoned his search because he found the result to be very complicated. Now, with the help of computer graphics, we know that Newton's method for cubic polynomials even with real coefficients has nontrivial and rich dynamical behavior [74]. Computer experiments and the theorems suggested by the observations lead to a deeper understanding of Newton's method and its *limitations.*

Numerical Analysis on the WWW

A Numerical Analysis WWW site has been developed and is accessible at the URL

`http://indy.cs.concordia.ca/na/.`

Newton's Method CGI

A CGI script for Newton's method has been developed and is available on the WWW at the URL

`http://indy.cs.concordia.ca/na/prog/newton/.`

As input it obtains the expression from the user in an HTML form, which is passed to a lexical analyzer and a parser to check for the syntax, and then to the Newton's method 'engine' which carries out the computations and returns the root of convergence and number of iterations to the root as output.

Newton's Method Java Applet

Java Applets which implement Newton's Method for certain predefined, prototypical functions can be useful. An example is available on the WWW at the URL

`http://www.cs.utah.edu/~zachary/isp/applets/Root/Newton.html` which, given an initial guess, illustrates the real-time evolution of the Newton's method for various one-dimensional nonlinear functions.

While teaching Newton's method, a short biography of Newton can be shown in the classroom.

13.6 Complex Analysis

Complex numbers are often encounter by students in courses on advanced algebra. Iterations in the complex plane give rise to one of the most intriguing dynamical systems.

13.6.1 Complex Quadratic Functions, Julia Sets and Fractals

The historical motivation for invention of complex numbers stems from the need to solve (quadratic) algebraic equations such as $x^2 + 1 = 0$, which have no real solution.

The simplest example of a nonlinear iteration procedure in the complex numbers is given by the transformation

$$f_0(z) = z^2. \tag{8}$$

Geometrically, the squaring of a complex number means that the corresponding length of z is squared in the usual sense and the corresponding angle $arg(z)$ of z is doubled $(\mathrm{mod}\, 2\pi)$. The following dynamical behaviour is observed on iteration $z_{n+1} = f_0(z_n) = z_n^2$, $n = 0, 1, 2, \ldots$: depending on the location of the initial point z_0. z_0 *inside* the unit circle leads to a sequence which converges to 0, z_0 *exactly* on the circle leads to a sequence which stays there forever, and z_0 *outside* the unit circle leads to a sequence which escapes to ∞. Thus, the complex plane of initial values is subdivided into two subsets: the *escape set* E_0, which consists of points which escape on iteration and the *prisoner set*, P_0 which consists of points which remain in a bounded region on iteration. The interior of P_0 can be interpreted as the *basins of attraction*, the attractor being the point 0; E_0 can be also interpreted as the basins of attraction, the attractor being the point ∞. The *boundary* between these basin of attractions is known as the *Julia set* J of f_0. Thus, P_0 is the disk around 0 with radius 1; J_0 is the unit circle and E_0 is the exterior of the disk.

This can be used as a background to introduce students to the iteration of

$$f_0(z) = z^2 + c, \tag{9}$$

where c is some complex parameter. Let $E_c = \{z_0 :\mid z_n \mid\rightarrow \infty, n \rightarrow \infty\}$ and $P_c = \{z_0 : z_0 \notin E_c\}$ be the escape set and prisoner set for the parameter c, respectively. The Julia set J_c of f_c is then the boundary of E_c. In this case, J_c, for a given c, turns out to be geometrically nontrival (unlike the previous case), and is actually a fractal set.

Finding iterates of equation (9) by hand becomes very complicated or even infeasible after a few steps. However, such an iteration can be readily carried out on a computer.

When computing P_c (or E_c) how large must an iterate be so that we can decide that the orbit will definitely escape to ∞? The answer to this question turns out to be a simple

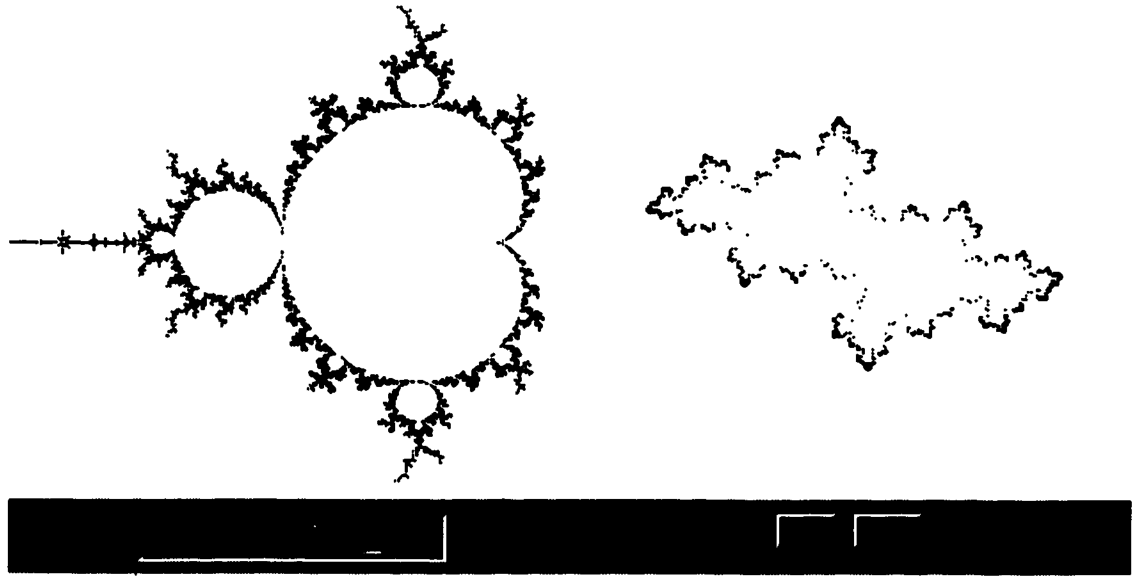


Figure 34: Julia Set Java Applet Illustrated.

proposition [33, page 378] which is fundamental to the implementation of the program which computes P_c.

For some values of c, J_c appears to be in one piece (mathematically, connected) and for others it appears to be a dust of points (mathematically, totally disconnected). We then ask: for a given c, what is the structure of J_c? The answer to this question turns out to be nontrivial and has led to the discovery of the Mandelbrot set $\mathcal{M} = \{c \in \mathbf{C} : J_c \text{ is connected}\}$.

Julia Set CGI

Julia set can be generated (on the server-side) using a CGI. An example is available on the WWW at the URL

`http://aleph0.clarku.edu/~djoyce/julia/explorer.html.`

Julia Set Java Applet

Julia set can be generated (on the client-side) by a Java applet. An example is presented in Figure 34 shows the Julia sets of f_c for $c = -0.648 + 0.442i$. It is available on the WWW at the URL

`http://indy.cs.concordia.ca/www/java/ds/.`

Given a complex number that is chosen from an image of the Mandelbrot Set, it generates the corresponding Julia Set.

13.7 Probability

Statistical approach to chaotic dynamical systems has received lot of attention in the last decade [42]. We then ask: What can students taking basic statistics courses be taught of this exciting field?

13.7.1 Probability Density Functions

The concept of a probability density function is fundamental in introductory statistics courses. There exists an inherent connection between chaotic dynamical systems and probability density functions.

Let $\tau : [0,1] \to [0,i]$ be a transformation. Instead of studying the evolution $\{\tau^n(x)\}$ of a single point x in the phase space $[0,1]$ by the transformation τ, we study the evolution of a density f of a collection (or ensemble) of points in $[0,1]$:

$$\begin{array}{cccccccc} x_1 & x_2 & x_3 & \cdots & x_n & & & f \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & & \downarrow \\ \tau(x_1) & \tau(x_2) & \tau(x_3) & \cdots & \tau(x_n) & & & ? \end{array}$$

The question we ask is: How does the probability density function f evolve? We answer the above question using a computer program. We choose an initial point x_0 and iterate it under the map, say m times. We then observe the parts of the interval $[0,1]$ that are visited by the orbit $x_0, \ldots, x_m$ and their frequency of visitation.

The algorithm consists of the following steps:

Step 1. Start with a 'typical' real number x_0 in $I = [0,1]$. Such a number is usually chosen by a random–number generator.

Step 2. Using this number generate a sequence of N numbers $x_1, x_2, \ldots, x_N$.

Step 3. Choose a bin size h and divide the interval I into n bins such that $nh = 1$.

Step 4. Count the x values in each bin.

Step 5. Construct a histogram.

The histogram displays the frequency with which states along a trajectory fall into given regions of the state space. As $N \to \infty$, the histogram can be replaced (as a consequence of Birkhoff Ergodic Theorem) by a probability density function $f(x)$. Then

$$\int_0^1 f(x)dx = 1$$

and $f(x)dx$ is the probability that an arbitrary number in the sequence lies between x and $x + dx$.

The Invariant Density of the Logistic Map Java Applet

For the logistic map $\tau(x) = 4x(1-x)$, the invariant density obtained by an implementation of the above method is shown in Figure 35. The snapshot was taken from a Java applet available on the WWW at the URL

`http://indy.cs.concordia.ca/www/java/ds/`.

We used 10,000 points, 1000 bins and 500 iterations. Numerically computed invariant density is in a good agreement with the theoretical invariant density given by

$$f^*(x) = \frac{1}{\pi\sqrt{x(1-x)}}$$

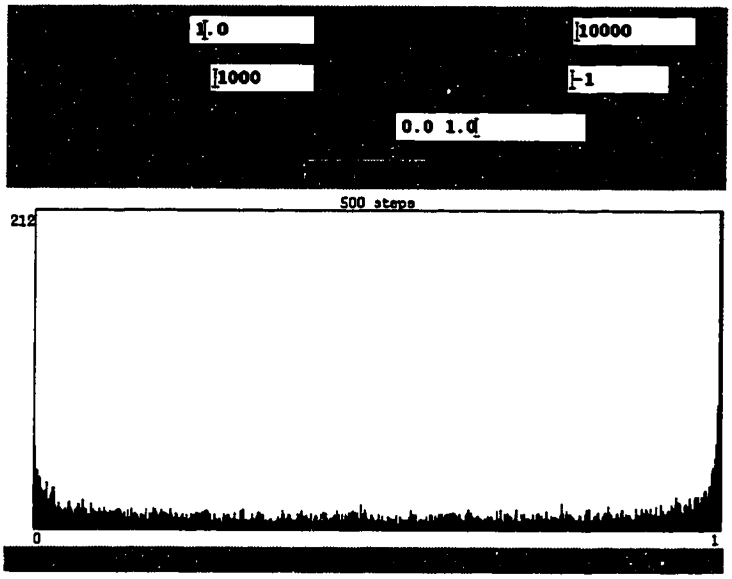


Figure 35: The Invariant Density f^* of the Logistic Map $\tau(x) = 4x(1-x)$ obtained by Computation of the Histogram of a Numerical Orbit.

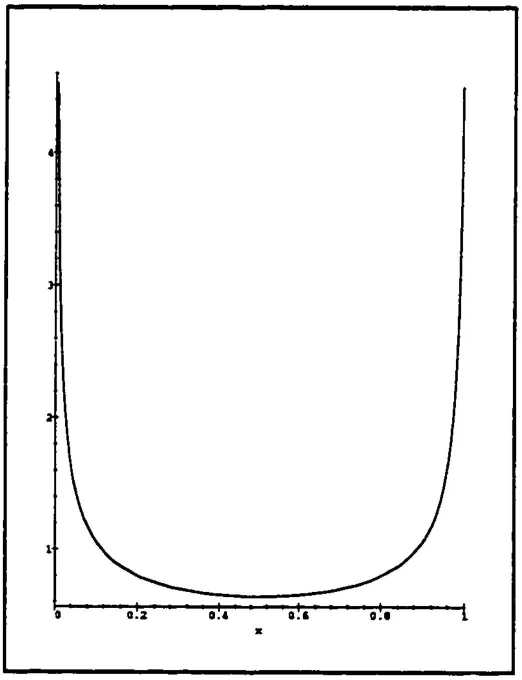


Figure 36: The invariant density f^* for the Logistic Map $\tau(x) = 4x(1-x)$.

and shown in (the Maple generated) Figure 36.

We now ask: How does the structure of the probability density function f change as the system parameters are changed? In this case, a computer program with the facility for interactively manipulating the parameter r, and computing and displaying the corresponding invariant density, can be quite useful for encouraging the spirit of computer experimentation among students.

A word of caution can be brought to students' attention. Calculation of an invariant density of a chaotic system on a computer by computation of the histogram of a sufficiently long numerical orbit can depend strongly on the *precision* of the arithmetic due to the presence of roundoff/truncation errors (which may disappear as the precision is increased). It has been shown in [33, Section 10.3] that the results obtained by use of single precision arithmetic (which can be spurious) can be drastically different from that obtained from the use of double precision.

Chapter 14

Teaching and Learning Iteration of Functions and Fixed Points

The study of iterations [...] is central to the applications of mathematics.

— Gilbert Strang

Iteration is a fundamental area in dynamical systems theory which studies the behaviour of trajectories of points under functions. Apart from its significance, the topic of iteration of functions, is also *accessible* even at the level of high school. It is therefore crucial to consider *how* a practical teaching and learning environment for it can be realized.

In this Chapter, we present lesson plans from two contrasting viewpoints: constructivist theory [56] and socio–cultural theory [73]. that can be carried out while teaching a class the topic of iteration of one–dimensional functions from a closed interval into itself, and attracting and repelling fixed points. In each of the these pedagogical perpectives, we introduce the topic from algebraic, geometric and computational approaches, in that order, expressing the necessity and motivating the transition from one to another. Our purpose is to show how various pedagogical suggestions made in the previous chapters can be realized in a real–environment of a classroom. Pedagogical strategies and suggestions made here could eventually be incorporated into a suitable instructional design.
The lesson plan model used here is presented in Table 4 and is adapted from [4] and [64]. A variant of this model will be used, as appropriate.

14.1 The Constructivist Approach: Pedagogical and Epistemological Implications

The lesson is based on the following assumptions underlying the constructivist approach to teaching and learning:

Pedagogical and Epistemological Implications
Mathematical Content
Identifying mathematical objects and operations in the topic
Sequencing the topic in a hierarchy of topics
Learning Objectives
Learning Resources
Preassessment Strategy
Identifying prerequisite Mathematical Content
Assessing student readiness to learn the topic
Teaching/Learning Strategy
Organizing the session
Conducting the session
Managing the learning environment

Table 4: The Lesson Plan Model.

- A constructivist perspective of teaching mathematics focusses on the student *rather* than the teacher. When one takes the approach of constructivism towards teaching, one rejects the assumption that one can simply pass on information to the student and expect that understanding will result [34]. When teaching concepts as a form of communication, the teacher must form an adequate model of the student's way of viewing an idea and then must assist the student in restructuring those views to be more adequate from the student's, and from the teacher's perspective.

- The teacher should determine what kind of knowledge can be used as a foundation for the building of the intended concept and ascertain that such a basis is present in the student. Thus an evaluation — whether each step in the proposed construction is accessible to the student — is necessary.

- The constructivist approach raises the following question: how can the student be guided to construct of his/her mathematical schemas on the basis of his/her existing knowledge? A modest version of this could be achieved in principle by understanding what epistemological or cognitive obstacles do students encounter in understanding a mathematical concept. Therefore the teacher has to continually ask the following questions: in teaching a certain concept, what skills serve as prerequisites? What misconceptions and alternate conceptions can arise and how can they be corrected?

14.2 Mathematical Content

14.2.1 Identifying the Mathematical Objects and Operations in the Topic

- Mathematical *objects* being explored: functions, fixed points.

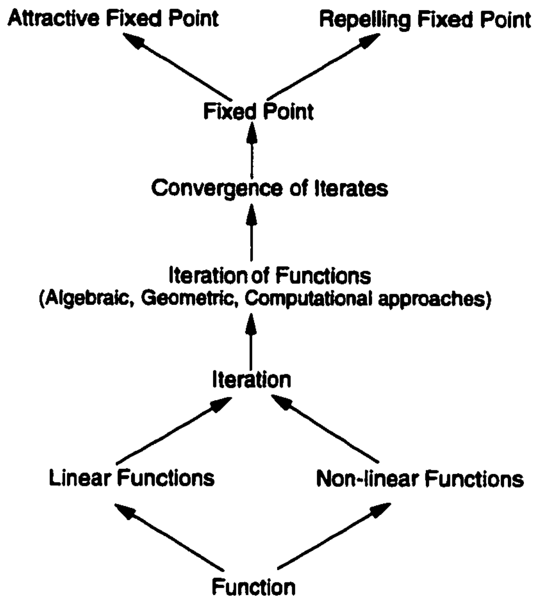


Figure 37: Constructivist Approach: Hierarchy of Topics in the Lesson.

- Mathematical *operations* being performed: composition, iteration, differentiation.

14.2.2 Sequencing the Topic in a Hierarchy of Topics

The main topic can be divided into sub–topics which are illustrated in Figure 37 according to their hierarchical dependence. Teaching should follow this order.

14.3 Learning Objectives

The lesson has the following learning objectives:

- Students realize the significance of the role of the teacher in guiding them towards verification of their own constructs of concepts involved in the topic.
- Students *experience the power* and *realize the necessity* of computer technology in the study of iteration of functions. In doing so, they can realize that the role of exploration via experimentation is important in the process of learning the topic and verification of their understanding of concepts involved.

14.4 Learning Resources

- **Literature.** There are various resources available for the topic such as [16], [33] which are accessible at a pre–university level. One of the above can be used as a reference textbook. Apart from that, prepare class–notes prior to the lecture.

- **Computer Systems.** Computers connected to a LAN. The type of platforms required depends on the software available and chosen (described below) and vice versa.
- **Computer Softwares.** Several dynamical systems softwares for educational purposes are available [9]. For a micro-computer environment, there are, for example: PHASER and DYNAMICS on a IBM-PC (compatible) computer and MacMath on an Apple Macintosh computer. Choose one of these according to your computing environment. Use the following criteria when making a choice: capabilities, good user-interface, on-line help facility, graphical options, cost, number of nodes allowed for a site license. This lesson describes computational experiments using PHASER.
- **Visual Equipment.** Overhead-screen and projector in the classroom for transperencies.

It is assumed that the classroom is equipped with a black/white board. Make sure that the various technical equipment is in order before the class begins.

14.5 Preassessment Strategy

14.5.1 Identifying Prerequisite Mathematical Content

Identify the prerequisites at each step of instruction. Before introducing any new concept, check whether the students have the requisite background and preparation for pursuing it. If needed, review the desired prerequisite.

Mathematical Connections

It has been observed that students understand a new concept better if it has connections to notions *already familiar* to them. It can be mentioned in the class that topics in mathematics to which iteration of functions makes connections are: *linear and non-linear (quadratic) functions, composition of functions, numerical patterns, geometric sequences and series, convergence, divergence, infinity, notion of limit, slope, graphing, geometrical patterns and their visualization, calculator/computer (computation), algorithm, mathematical software.*

Ask the students if they have already come across these notions before. We shall denote the prerequisite content by the symbol **[P]**.

14.5.2 Assessing Student Readiness to Learn the Topic

Conduct a short survey in the class — due to time constraint, an oral survey is recommended. The survey can include questions on their background regarding the notions mentioned above. It will be useful while conducting the lesson. Ask also their preferences in introduction of a topic: mathematical equations, figures, computer simulations.

14.6 Teaching/Learning Strategy

14.6.1 Organizing the Session

Social Context of the Session
Prior to the lecture, the following need to be assessed:

- **Level of instruction.** This lesson is designed for students at a pre–university level.
- **Number of students in the classroom.** To realize a close teacher–student contact, it is suggestive to have a small number of students.
- **Types of classroom communication.** Lecture by teacher (supplemented with black/white board, literature, projector for transperencies, computers with necessary softwares) and between student $\longleftrightarrow$ computer via the software GUI.

14.6.2 Conducting the Session

Functions [P]
Since we are discussing iteration of *functions*, the notion of what a function and its notation are, should be clear to the student. The notion, from the viewpoint of iteration, can be developed from an intuitive perpective as a rule or function applied to each input or first coordinate x to produce a corresponding output or second coordinate $f(x)$.

Iteration

Motivation — Calculating $\sqrt{2}$
It is important for the students to know *why* they have to learn the topic of iteration of functions. Start the topic by first motivating the students towards it. The following can be used as a guideline.

When we are computing π or the roots of a polynomial, we encounter iterations. Ask students if they have come across any such instances. They can be reminded of the well-known *quadratic equation*:

$$ax^2 + bx + c = 0, \quad a, b, c \in \mathbf{R}.$$

For example, consider $f(x) = x^2 - 2$. Its roots are $\pm\sqrt{2}$. Since $\sqrt{2}$ (being irrational) can not be represented exactly on the Cartesian plane $(x, f(x))$ in practice, we look for its *approximate* or *numerical* value. In other words, we ask: Can we find a numerical value where the curve $f(x)$ intersects the x–axis? Mention to students that we shall come back to this problem later.

Introduce the formal definition of iteration after the above background.

Definition 14.6.1 (Iteration) The repeated composition of same function on itself is called *iteration*.

Once the concept of iteration has been introduced, mention the *purpose* of iteration. Ask and answer the following questions: what is the *result* of iteration? What are we looking for while carrying out an iteration of a function — *asymptotic* or *long–term* behaviours.

The ways of introducing the concept of iteration of functions fall broadly into three categories — algebraic, geometrical, computational — teaching should proceed in a manner which shows that they are *linked* to each other. Each of these methods can lead to various misconceptions and alternate conceptions (see Chapters 8, 9, [64, §IV]) of which the teacher has to be aware. Also, among these approaches there are different ways to introduce the concept of iteration of functions. These different approaches altogether identify the same topic. Another reason for their introduction in the class is that different students may have different (or a combination thereof) preferences (algebraic, visual, algorithmic) towards understanding.

The Algebraic Approach

Composition of Functions [P]

The concept of composition of functions can be difficult to understand for some. To find $g(f(x))$, use f first and then g. Start with a domain value x and use it to obtain the corresponding range value, $f(x)$. Then use $f(x)$ as the next domain value and use it to obtain the corresponding range value, $g(f(x))$. It should be pointed out that composition of functions is not always commutative.

Iteration as a Composition of Functions

The process of iteration is developed from an intuitive perspective that connects it to the notion of composition of functions — it connects the concept of the function to iteration.

The iterative process arises when repeatedly forming the composition of a function with itself. This can be illustrated in class by Figure 38 in which an argument is repeatedly passed through the same function $f(x)$ in a cyclic manner, resulting in the following sequence of functional values:

$$x_0 \rightarrow f(x_0) \rightarrow f(f(x_0)) \rightarrow f(f(f(x_0))) \rightarrow \cdots \qquad (10)$$

Often, the notation f^n is used for the n^{th}–iterate which can be confused with n^{th}–power of f. Also, the repeated application of f soon leads to the notational problem with too many parentheses. Ask the students what is their understanding. To check, ask whether $f^2(x)$ is equivalent to $f(f(x))$ or $f(x) \cdot f(x)$. It might be useful to introduce an alternate notation for n^{th}–iterate of f:

$$f^{\circ n} = \underbrace{f \circ f \circ \cdots \circ f}_{n-\text{times}}. \qquad (11)$$

Ask the students if this notation is more comprehensible.

Iteration as a Feedback Loop

The process of iteration is often referred as a feedback loop. The result of each evaluation of function is plugged back into the function. This can be expressed algebraically by equation (10) and illustrated in the class by a flow diagram (see Figure 38) or by a calculator repeating the same button — say, of $sin(x)$, for some initial value — over and over again (iterating).

A convenient way of representing iteration is to consider some initial argument x_0, and use it to obtain the functional value $f(x_0) = x_1$. Then use x_1 as the next argument and

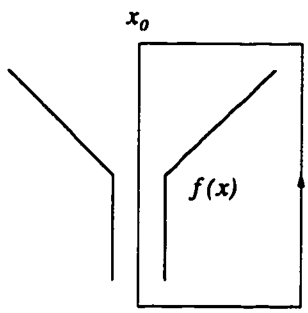


Figure 38: Composition of Functions I.

Iteration	Numerator	Denominator
1	$x+3$	2
2	$x+3+6$	4
3	$x+3+6+12$	8
4	$x+3+6+12+24$	16
5	$x+3+6+12+24+48$	32

Table 5: The First Five Iterates of $f_{\frac{1}{2},\frac{3}{2}}(x) = \frac{1}{2}x + \frac{3}{2}$.

obtain the next functional value $f(x_1) = x_2$, and so on. This results the following *sequence of iterates* or *arguments*:

$$x_0 \to x_1 \to x_2 \to x_3 \to \cdots \tag{12}$$

See Figure 39. This is also shown in Figure 40 as a one–step feedback process.

Sequences and Series **[P]**

Geometric sequences and series arise during iteration of functions: the consecutive term in each sequence is generated by repeatedly *multiplying* the previous term by a constant. A review of the definition, relevant formulas for the n^{th}–term, sum of first n terms and of infinite number of terms of a geometric series can be useful here.

After the introduction of the concept of iteration of functions, state examples of iteration of *linear* functions which are used for their consistent behaviour.

Example 14.6.1 [38] We iterate the function

$$f_{\frac{1}{2},\frac{3}{2}}(x) = \frac{1}{2}x + \frac{3}{2}, \tag{13}$$

where $x \in [0,1]$. Table 5 shows the result of the first five iterates of $f_{\frac{1}{2},\frac{3}{2}}$. The procedure for finding higher iterates soon becomes cumbersome and it might be better to have a *formula* to find the iterates.

Ask the students if they see any numerical pattern in Table 5. The numerator is x plus a geometric sequence and the n^{th}–term of denominator is 2^n. Therefore, using the formula

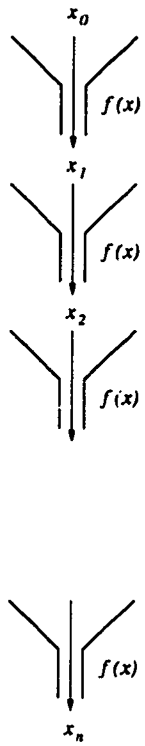


Figure 39: Composition of Functions II.

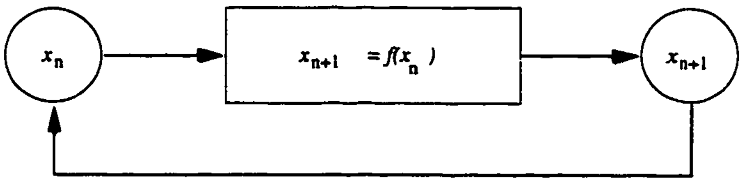


Figure 40: Composition of Functions III.

for the sum of first n terms of a geometric series, we have

$$f^{\circ(n)}_{\frac{1}{2},\frac{3}{2}}(x) = \frac{x + 3 \cdot 2^n - 3}{2^n} = 3 + \frac{x}{2^n} - \frac{3}{2^n} \tag{14}$$

Introduce some other examples as well. There is a possibility that the students might be led to believe that given any function, one can *always* find its iterates. To avoid any such misconception, introduce the following example:

$$f(x) = -\sqrt{x}$$

is defined only for $x > 0$. Starting with any $x_0 > 0$, we get $f^{\circ(2)}(x_0) < 0$. Hence, we can not calculate $f^{\circ(3)}(x_0)$.

Convergence/Divergence of Iterates

We are looking for *long-term* behaviours in iteration of functions: what happens to $f^{(n)}(x)$ for some initial value x, as $n \to \infty$? For example, ask the students what is the result of iteration of $f^{(n)}_{\frac{1}{2},\frac{3}{2}}(x_0)$ for some initial value x_0 as $n \to \infty$? A warning is in order: it is possible for some students to conclude that this could be done by finding the sum of first *infinite* number of terms of the geometric series mentioned above in (14) which is infeasible (since the common ratio is greater than 1) in this case. We find whether the sequence of iterates $\{f^{\circ(n)}_{\frac{1}{2},\frac{3}{2}}(x)\}_{n \geq 1}$ is convergent, i.e., by finding the *limit* of the sequence:

$$\lim_{n\to\infty} f^{\circ(n)}_{\frac{1}{2},\frac{3}{2}}(x) = \lim_{n\to\infty} \left(3 + \frac{x}{2^n} - \frac{3}{2^n}\right) = 3. \tag{15}$$

Ask the students whether we obtain the same behaviour (i.e., of convergence) for the iterates of $f_{\frac{3}{2},\frac{1}{2}}$ and explain if needed that in this case we have divergence. Mention that convergence of iterates brings an important property of interest — the concept of a fixed point.

Fixed point

Mention to students that we are searching for *patterns* in iterative behaviours. One such pattern is the *fixed point.* Introduce the following:

Definition 14.6.2 (Algebraic) A fixed point of a function is a point such that $f(x^*) = x^*$ for some initial value x^*.

Also point out that the concept of a fixed point is the result of the *long*-term iterative behaviour of a function and may not always appear immediately. To avoid any danger of (incorrect) generalization (that the existence of a fixed point is a trivial property) on part of students, ask them whether a given function will always have a fixed point. Give an example of a function with *no* fixed points.

Fixed points can have one of several different kinds of behaviours with respect to iteration, when we consider initial values close to the fixed point. Their nature can be *attracting* or *repelling.* These behaviours depend on the sign of the derivative of f at the fixed point, as shown in Table 6.

Example 14.6.1 has a fixed point at $x = 3$. Use some initial values to illustrate to students that this indeed is the case. Finally, since $f'(x) = \frac{1}{2} < 1$, it is an *attractive* fixed

Derivative at x^*	Nature of the Fixed Point
$f'(x^*) < -1$	Repelling
$-1 < f'(x^*) < 0$	Attracting
$0 < f'(x^*) < 1$	Attracting
$f'(x^*) > 1$	Repelling

Table 6: The Relation between the Sign and Magnitude of the Derivative at and the Nature of the Fixed Point x^*.

point. Ask the students whether changing the initial value will make any difference on the nature of the fixed point.

Mention that we can use the method of iteration of functions to calculate roots of polynomials; in the class, the case of quadratic polynomials $x^2 - a$, $a > 0$ can be introduced.

Example 14.6.2 The equation $x^2 = a$ can be written as

$$x = \frac{1}{2}\left(x + \frac{a}{x}\right).$$

Therefore, $(\pm\sqrt{a}, \pm\sqrt{a})$ is an *attractive* fixed point of

$$f_{\sqrt{a}}(x) = \frac{1}{2}\left(x + \frac{a}{x}\right). \quad (16)$$

Ask the students to verify that.

The function, its iterates and its derivative, which decides the nature of a fixed point of a function, each have a geometrical intrepretation. The difficulty in being able to visualize the behaviour of higher iterates specifically of non–linear functions such as in equation (16) is one of the motivations for resorting to the geometrical approach to iteration of functions.

The Geometrical Approach

Have the figures ready on transperencies for display with the overhead–screen and projector. Prepare visual descriptions of key aspects of dynamics illustrated by the graphics. They can be shared with students during the class. It will be useful to have coloured graphics to distinguish axes, initial starting points of the system being iterated, paths and the fixed point(s). Use of pointer should be made to explain the figures.

Graphs [P]

Students should be familiar with what a graph is, how to draw a graph of linear and non–linear (in our case, quadratic) functions and their shapes (straight line, parabola).

Iteration as a Graphical Process

A very useful method employed in understanding the iteration is *graphical analysis* or *graphical iteration.* It allows a student to follow simple orbits of certain functions using *only* the

begin {Evaluate} Draw a vertical line from an input value a on the x-axis up to the graph of the function. Mark the intersection point as A. A has a y-coordinate $f(a)$. {Transfer} From A, draw a horizontal line to the intersection point B on the diagonal. {Reflect} From B, draw a vertical line up to the top. The x-coordinate at this point is the output value $f(a)$. **end**

Table 7: The Algorithm for Graphical Iteration.

graph of the function instead of the iterates. This is a departure from the algebraic to the geometrical approach.

Iteration can be presented as a simple algorithmic process of drawing horizontal and vertical segments, first to the graph of the function under study and then to the diagonal line $y = x$, which reflects it back to the graph again. This is represented in form of an algorithm given in Table 7. Carry out this process for Example 14.6.2.

Fixed points can be identified by intersections of the graph of the function with the diagonal. After indicating this, the following definition can be introduced:

Definition 14.6.3 (Geometrical) A fixed point of a function is a point where the graph of the function and the diagonal line $y = x$ intersect. Thus, if the x-coordinate of the intersection point is x^*, then the y-coordinate of the intersection point is x^* as well.

Mention why a fixed point is called as such[1].

The respective property of attracting or repelling of a fixed point can be determined by examining graphical iteration in the neighbourhood of the fixed point. Mention that the graphical approach has a specific advantage here — it shows us two types of attracting and two types of repelling points depending on the path to the fixed point which can be either a staircase or a spiral. These behaviours are closely related to the sign of the slope of the graph of the function at these fixed points, as shown in Table 8.

Attractive Fixed Points

There are two types of attractive fixed points. For the first type (see Figure 41), we observe a *staircase in* behaviour toward the fixed point for all initial values close to the fixed point.

[1]The graphical iteration process can not proceed because moving the point left or right to the diagonal would not move the point at all — it is *already* on the diagonal. Hence the name 'fixed' point.

Slope at x^*	Nature of Path	Nature of the Fixed Point
$m = f'(x^*) < -1$	Spiral out	Repelling
$-1 < m = f'(x^*) < 0$	Spiral in	Attracting
$0 < m = f'(x^*) < 1$	Staircase in	Attracting
$m = f'(x^*) > 1$	Staircase out	Repelling

Table 8: The Relation between the Sign and Magnitude of the Slope at, Nature of Path and the Nature of the Fixed Point x^* during Graphical Iteration.

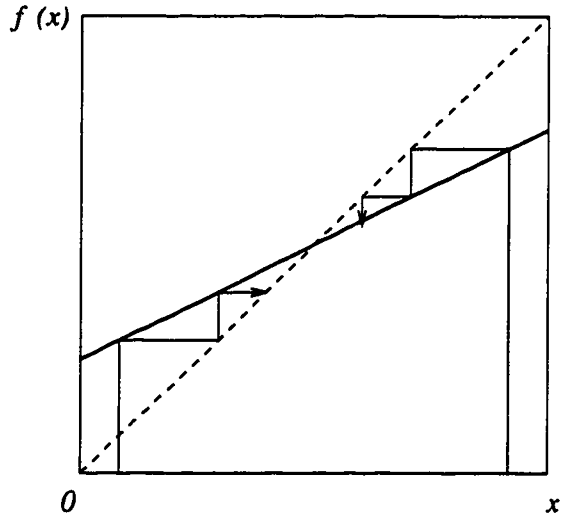


Figure 41: The 'Staircase–In' Behaviour towards an Attractive Fixed Point in case of a Linear Function.

The slope of the function at the fixed point is *positive*. For the second type (see Figure 42), we observe a *spiral in* behaviour towards the fixed point for all initial values close to the fixed point. The slope of the function at the fixed point is *negative*. Explain both the cases using corresponding figures, indicating the initial starting point, the path, ... asking the students what do they *see*.

Repelling Fixed Points

There are two types of repelling fixed points. For the first type (see Figure 43), we observe a *staircase out* behaviour toward the fixed point for all initial values close to the fixed point. The slope of the function at the fixed point is *positive*. For the second type (see Figure 44), we observe a *spiral out* behaviour towards the fixed point for all initial values close to the fixed point. The slope of the function at the fixed point is *negative*. Again, explain both the cases using corresponding figures, pointing the initial starting point, the path taken by it, ... again asking the students what do they *see*.

Ask (and if needed, explain) the students if they see any connection between slope of the graph of the function and the nature of fixed point indicated in Table 6, and between Table 6 and Figures 41–44.

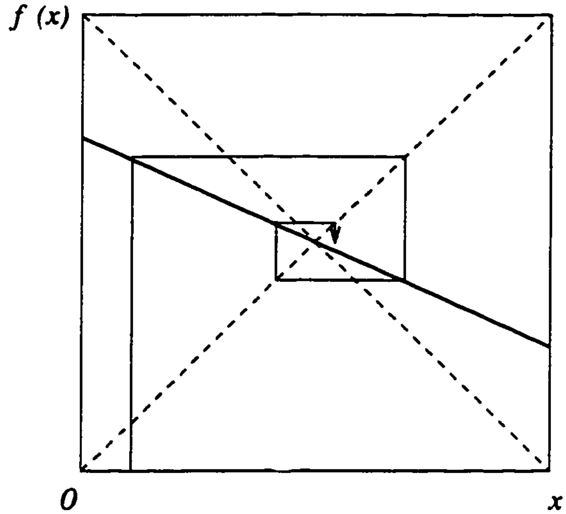


Figure 42: The 'Spiral–In' Behaviour towards an Attractive Fixed Point in case of a Linear Function.

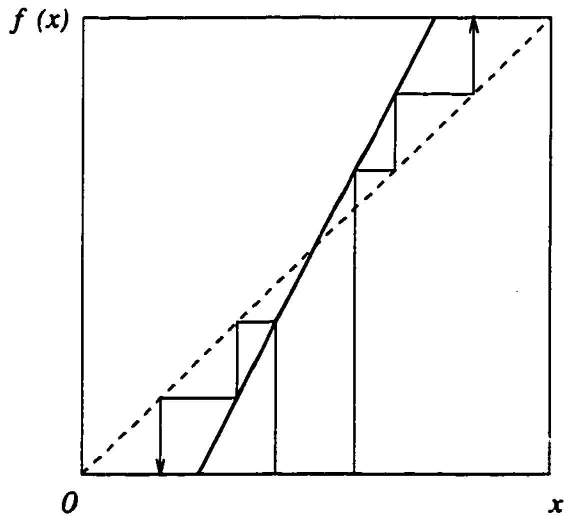


Figure 43: The 'Staircase–Out' Behaviour from a Repelling Fixed Point in case of a Linear Function.

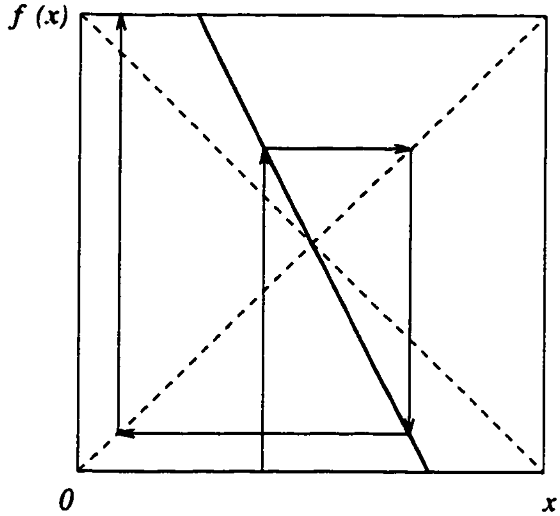


Figure 44: The 'Spiral-Out' Behaviour from a Repelling Fixed Point in case of a Linear Function.

Stress the importance of the *connection* between the geometric process of graphical iteration and the algebraic process of composition of functions. Several steps of graphical iteration can be performed and the corresponding behaviour observed and explained as they relate to the region around the fixed point. Use completed pictures of staircase and spiral behaviour to discuss convergence to or divergence from a fixed point. Stress the visual impact of iteration graphs and seek verbal descriptions of comparison and contrast. Opportunities for practicing the process of graphical iteration on both linear and non-linear functions should be provided to students. For Example 14.6.1, Figure 45 shows the graphical convergence via 'staircase in' to the attractive fixed point $(3,3)$ from an initial starting point $x = 8.3$.

Quadratic functions such as

$$f_a(x) = ax(1-x), \;\; 1 < \mathrm{a} < 3, \;\; x \in [0,1] \tag{17}$$

can be studied as prototypes of non-linear functions. For a particular value of a and x, calculate a few iterates of f_a. This calculation of iterates of quadratic functions soon becomes tedious. In general, often after a few steps, finding iterates of non-linear functions can become a cumbersome task. Have the students realize this. Also, constructing the phase diagram (representing initial values on x-axis and the iterates on the y-axis) in many cases may be very time consuming or even non-trivial. In such cases, computer technology can be useful. Then the numerical iteration and graphical analysis is usually done with the help of a computer. It can be indicated to students that the use of computer in the iteration of functions creates links between the algebraic and geometrical approaches (see Figure 46) in one setting: by generating computer graphics of the numerical data obtained from iteration. This above can be used as a motivation for our next approach.

The Computational Approach

```
begin
    {Initialize: a number k acting as a counter and
    an initial value x_k from the interval [0, 1]}
        input k = 0
        input x_k
        apply f to x_k to obtain f(x_k)
    {test if x_k is a fixed point}
    if f(x_k) = x_k, then
        x_k is a fixed point
        set m = f'(x_k)
        {test for m to decide the nature of x_k}
        if m < -1 then
            x_k is repelling; the path spirals out
        else if -1 < m < 0 then
            x_k is attracting; the path spirals in
        else if 0 < m < 1 then
            x_k is attracting; the path staircases in
        else if m > 1 then
            x_k is repelling; the path staircases out
    else
        set k = k + 1
    {repeat the above steps}
end
```

Table 9: The Iteration Algorithm Pseudo-Code for finding Fixed Points of Functions and their Nature.

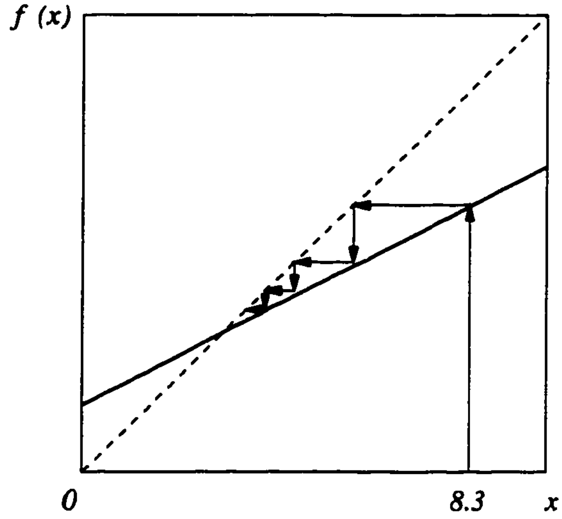


Figure 45: The 'Staircase–In' Behaviour to the Attractive Fixed Point (3, 3) of $f(x) = \frac{x+3}{2}$ starting at an Initial Value of $x = 8.3$.

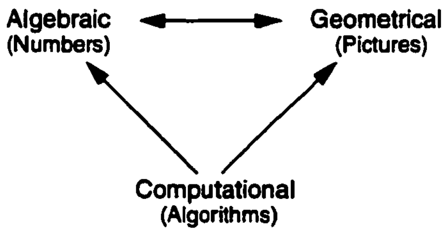


Figure 46: Links among the Approaches of Study of Iterations of Functions.

Computer Skills P

Students need familiarity with the basics of a computer — how to use the keyboard, pointing device such as a mouse and work with the GUI. Ask the students if they do. This should also be reflected in the oral survey taken at the beginning of the class.

Iteration using Computer Technology

Using an over–head display screen and projector the algorithm (pseudo–code) given in Table 9 should be introduced so that students can become familiar with the 'mechanics' of the software used later. Ask the students if they see the connections between Tables 8 and 9.

Iteration of Linear Functions using Computer

This part of the lesson can be carried in a computer laboratory. A previously chosen dynamical systems software can be used to iterate some linear and non–linear functions. For illustrative purposes, we shall use PHASER. Check the necessary set–up for each exercise prior to the lecture.

Each student is assigned one computer to work on. Have the relevant software set–up before the class. Carry out an example of each type on the computer to illustrate to students the usage of the software in the process of calculating iterates, and eventually fixed points.

The usage of the software can be shown to students. They should be introduced to

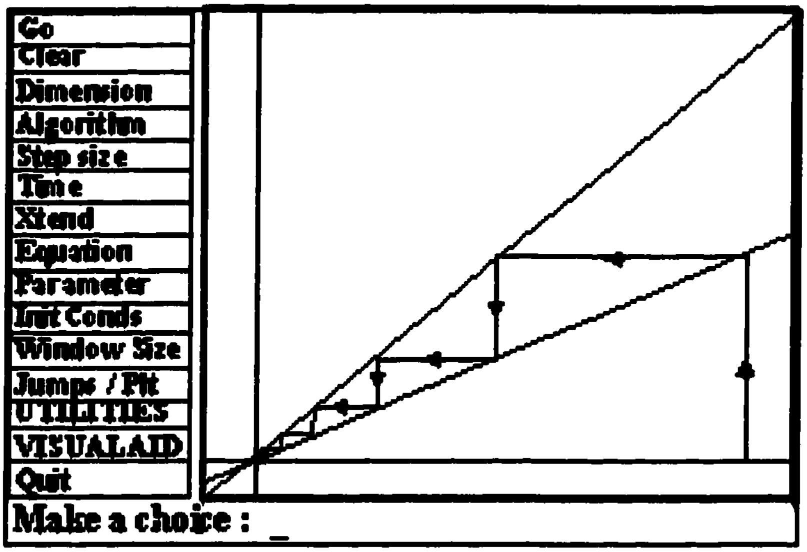


Figure 47: Convergence of $f(x) = 0.5x$ to the Attractive Fixed Point 0.

the PHASER Main Menu and the necessary buttons in it. Example 14.6.1 can be used in the demonstration since at this stage students would already be familiar with it. By using the `dislin1d` button on the PHASER GUI, iteration of any linear function $f_{a,b} = ax + b$ can be carried out. Explain the other necessary buttons on the GUI; `InitConds` button for giving an initial starting point and `Go` button for running the iteration program. Ask the students if they understand the usage instructions, if the graphics generated by the software are clear and comprehensible, and whether they find the graphics better than the ones shown earlier. Also, ask them if they have any difficulties using the computer or the software.

For the function

$$f(x) = 0.5x,$$

starting at 5.0, the iterates are given by the following sequence of numbers

$$5.0, 2.5, 1.25, 0.625, 0.3125, \cdots.$$

The graphical illustration (Figure 47) using PHASER GUI shows the convergence of the sequence to the attractive fixed point 0.

Students can then be given the task of verifying the results related to Example 14.6.1 obtained by algebraic and geometrical approaches.

Iteration of Non–linear Functions using Computer

In this part of the lesson, two specific cases can be illuminating: calculating the square–root of a positive integer, such as $\sqrt{2}$ (used in motivation of the topic) and some examples of the logistic function (finding iterates of which posed difficulty earlier).

I. Calculating $\sqrt{a}$ — Revisited

PHASER uses Newton's method as the underlying engine for finding roots of polynomials.

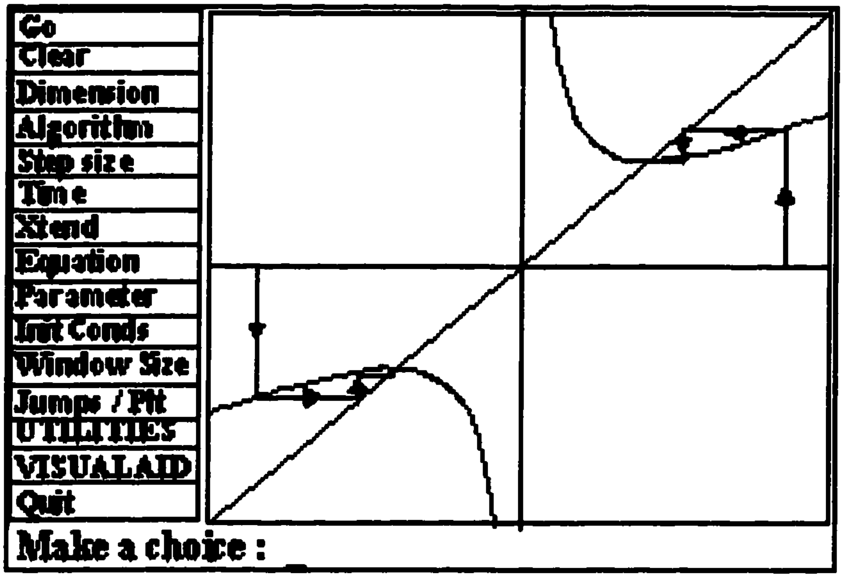


Figure 48: Calculating Roots of $f(x) = x^2 - 2$ by Graphical Iteration.

This part of the lesson can be carried out, by using the `newton` button on the PHASER GUI. The equation has two fixed points $x = \sqrt{a}$ and $x = -\sqrt{a}$. For positive initial conditions, we get a convergence to $\sqrt{a}$ and for negative initial conditions, we get a convergence to $-\sqrt{a}$. For example, for $a = 2$ and $x_0 = 3.0$, we obtain the following sequence of iterates which converges to $\sqrt{2}$: $x_0 = 3.00000$, $x_1 = 1.83333$, $x_2 = 1.46212$, $x_3 = 1.41499$, $x_4 = 1.41421$, $x_5 = 1.41421$, ..., and so we have reached a fixed point. See Figure 48 for a graphical illustration.

Check if the students got these results. Ask the students if this fixed point is attracting or repelling.

II. Iteration of the Logistic Function $f_a(x) = ax(1-x)$

This can be carried out by using the `logistic` button on the PHASER GUI. Once again, demonstrate to the students the other necessary buttons on the GUI, e.g., `Parameter` button for changing the parameter a.

Ask the students to carry out the iteration of the function $f_{2.7}(x) = 2.7x(1-x)$ on the computer. Give them an initial value for a start for which the convergence to the fixed point is known in advance. For most values in the closed interval $[0, 1]$, the iteration produces a long–term behaviour always leading to the same single value, 0.62962..., i.e., we have reached a fixed point. Ask the students if this fixed point is attracting or repelling. Also, ask them to change the parameter 2.7 in $f_{2.7}$ slightly to see if there is any difference in behaviour they observe[2]. For example, changing the parameter by 0.1 to 2.8 leads to the fixed point 0.6429.... Figure 49 shows that.

It can be pointed out to students that in case of non–linear functions, the same function

[2]The convergence to a single value occurs for all functions f_a with a parameter value of $1 < a < 3$.

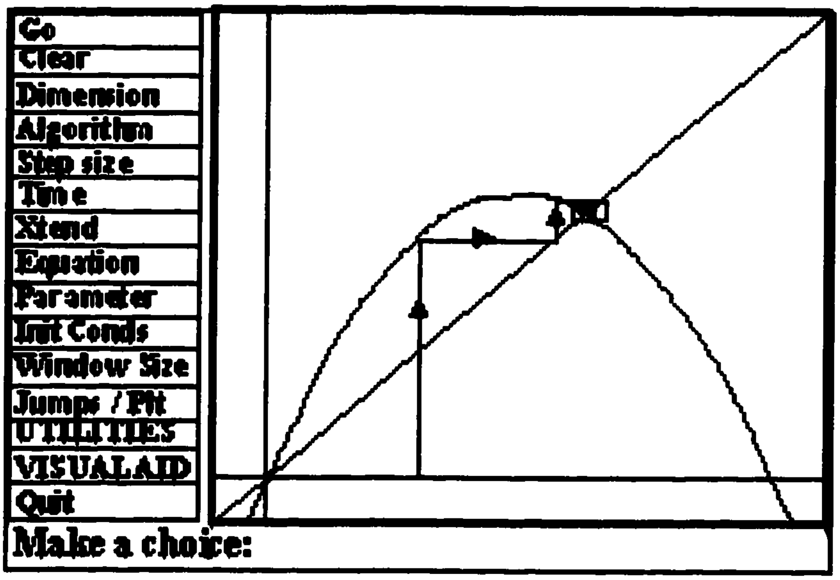


Figure 49: Convergence of $f(x) = 2.7x(1-x)$ to the Attractive Fixed Point.

can have *both* attractive as well as repelling fixed points. As an example, ask them to check this for the function $f_{1.6}(x) = 1.6x(1-x)$ for an initial value, say, $x_0 = 0.2$. f exhibits an attractive fixed point $(\frac{3}{8}, \frac{3}{8})$ and a repelling fixed point $(0,0)$ under graphical iteration.

Using the computer provides a quick way to graphically iterate a *family* of functions and allows for a convenient study of the short– and long–term behaviour of iterations. Besides that, iterates of non–linear functions can be easily computed. Ask them if they indeed find the computer as a *powerful* tool in exploring graphical iteration of functions.

On a Cautionary Side

Even though computer is a powerful tool in carrying out iteration of functions, it has its trade–offs which may not be immediate to students. It may mislead them to conclude that the computational approach is all that is important and studying other approaches (algebraic, geometrical) is not all that necessary. Mention that other approaches, besides being significant by themselves, are needed to *justify* computer–generated results. It should be emphasized that there are cases where graphical results may even be computer artifacts. For a given initial value, ask the students to find the behaviour of the function $f(x) = 2x(\bmod 1)$ by each approach. The computational approach (incorrectly) yields 0 as a fixed point [5]. With the help of two different calculators, Example 8.5.1 can also be illustrated in class. Explain the anomalous behaviour in each case briefly.

14.6.3 Managing the Learning Environment

Monitor if the students are following the topic such as problems in comprehending figures, computer and software usage. Make sure the technical equipment is working properly. Make efforts to give individual attention to each student as much as time permits.

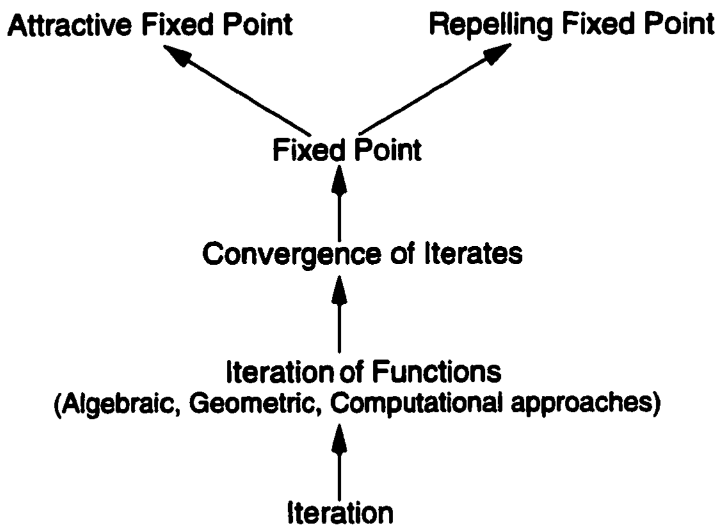


Figure 50: Socio–Cultural Theoriest Approach: Hierarchy of Topics in the Lesson.

14.7 The Socio–Cultural Theoriest Approach: Pedagogical and Epistemological Implications

The lesson is based on the following assumptions underlying the approach to teaching and learning of socio–cultural theory:

- In socio–cultural theory [73], the process of learning is *not* separated from the process of teaching. The social relation between teacher and student has to be considered in this process, as learning is *not* a relation between individuals and knowledge — it is the individual's introduction into an existing culture.
- The mathematics introduced to students in a classroom was developed *outside* the class by mathematicians together with a system of values and acceptable behaviours. Therefore, it is necessary to consider the learning process as a collective social and cultural process, along with its historical context and adjust the teaching accordingly.

14.8 Mathematical Content

14.8.1 Identifying the Mathematical Objects and Operations in the Topic

As in Section 14.2.1.

14.8.2 Sequencing the Topic in a Hierarchy of Topics

The main topic can be divided into sub–topics as illustrated in Figure 50. Teaching should follow the indicated hierarchical order.

14.9 Learning Objectives

Specifically, the learning objectives of the lesson are:

- Students realize that the topic is a 'living' subject with rich history and culture and appreciate the contributions of past mathematicians towards the topic.
- Students realize the significance of group–work in understanding the topic.
- Students *experience* the power and *realize* the necessity of computer technology in the study of iteration of functions.
- Students realize the usefulness of the technology of Internet and the WWW in learning the topic.

14.10 Learning Resources

- **Literature.** As in Section 14.4.
- **Computer Systems.** The teacher and student computer platforms virtually do not have any restrictions (e.g., they could be IBM–PCs (compatibles) or Apple Macintoshes). All computers should be connected by a LAN and to the Internet. Make sure that there are enough computers so that the class can be comfortably divided into small groups and that each group can be assigned a computer to work with.
- **Computer Softwares.** One of the computers on the LAN with a WWW server installed and running, WWW browser such as Netscape Navigator, locally accessible WWW–based dynamical systems programs (similar to `dislin1d`, `newton` and `logistic` of PHASER). Figure 51 illustrates the required computational environment for the lesson.
- **Visual Equipment.** As in Section 14.4.

It is assumed that the classroom is equipped with a black/white board. Make sure that the various technical equipment is in order before the class begins.

14.11 Teaching/Learning Strategy

14.11.1 Organizing the session

Social context of the session
For the lecture, assess the following:

- **Level of Instruction.** This lesson is designed for students at a pre–university level.
- **Number of Students in the Classroom.** It is suggestive to have a prior knowledge of the number of students in the class. This facilitates formation of groups for participation in classroom activities.

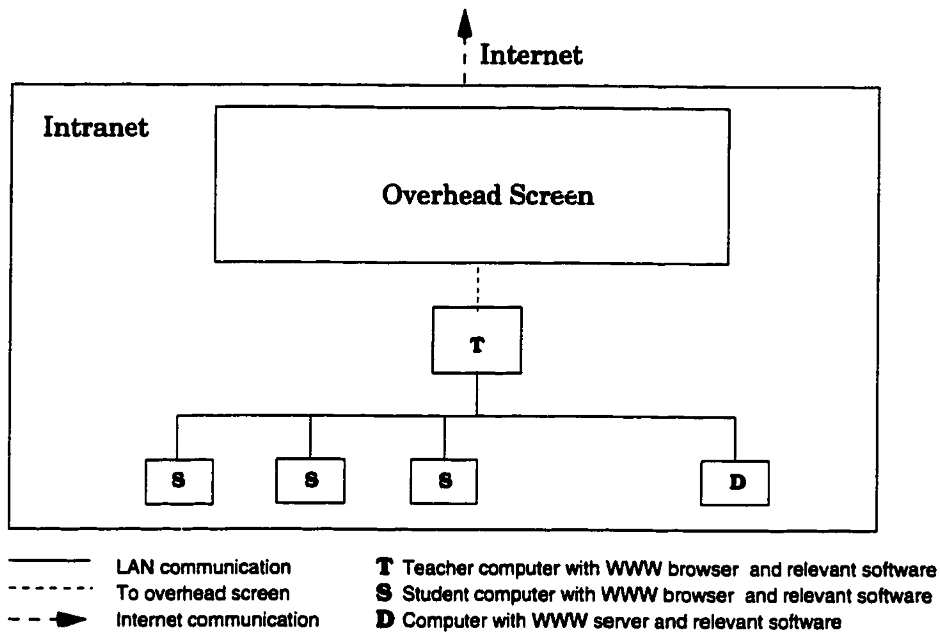


Figure 51: Schematic of the Computational Environment for the Lesson.

- **Student Groups.** Small groups for computer sessions. Larger groups for classroom activities.

- **Types of Classroom Communication.** Lecture by teacher, personal and communication over the computer network between student–teacher and student–student (supplemented with black/white board, literature, overhead–screen and projector with transperencies, computers with necessary softwares).

14.11.2 Conducting the Session

Have the "teacher computer" **T** along with necessary softwares and connected to the overhead screen/projector ready for the class.

Iteration

Motivation — History, Culture and Applicability of Iteration of Functions

History of the topic can be used to provide motivation for studying the subject (see Section 6.1). It should be pointed out to students that iteration of functions is a 'living' topic under constant evolution with many unsolved problems. Students can be told that the topic of iterations of functions has a long and interesting history and have been studied in various cultural contexts. Short biographies of past and present mathematicians who have been part of the evolution of the subject can be presented in the class. It can be mentioned that the topic has been pursued and left from time to time by mathematicians due to various difficulties (for example, absence of necessary computing power). Using an interactive map on the WWW, the part of the world from which these mathematicians came from can be shown.

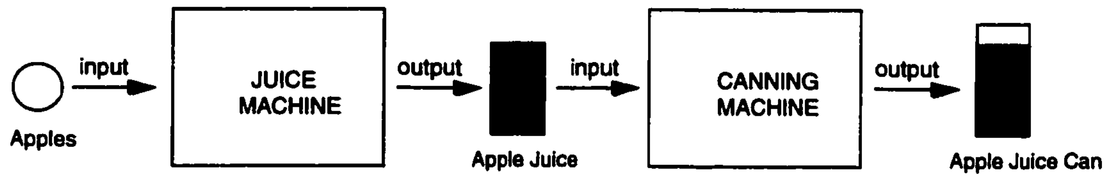


Figure 52: The Function Machine.

Mention that the *need* for iteration of functions occurs in many contexts in mathematics, and other applied subjects such as mathematical biology, e.g., the logistic equation population model to predict population growth, and mathematical economics, e.g., mortgage and compound interest rate problems.

Throughout the lesson, wherever possible, interesting historical instances and contexts of applicability of the functions under study should be shown.

Introduce the formal Definition 14.6.1 of iteration after the above background.

There are several ways of introducing the concept of iteration of functions and they fall broadly into three categories — algebraic, geometrical, computational.

The Algebraic Approach

Iteration as a Composition of Functions

An intuitive approach to the concept of composition of functions can be achieved through the notion of a *function machine.* The function machine accepts a raw material or input (domain) and produces a product or output (range). The next machine accepts the output from the previous machine as an input and produces a new output, and so on. The graphics of such a machine can help communicate the idea of this input–output relationship between x and $f(x)$. When combined, they illustrate the composition of functions in such a way as to make the ordering of functions visible. The following analogy can be introduced in the class. Consider the process of canning apple juice. The first machine accepts apples (input) and produces apple juice (output) by squeezing the apples. The second machine accepts the apple juice (input) and cans it, producing canned apple juice (output). A schematic representation of this is shown in Figure 52.

We can now move towards mathematization of the concept. The iterative process arises when repeatedly forming the composition of a function with itself. This is illustrated in Figure 38 in which an argument is repeatedly passed through the same function $f(x)$ in a cyclic manner, resulting in the sequence of functional values as in Equation (10).

For clarity, use the notation of Equation (11) for n^{th}–iterate of f.

A convenient way of representing iteration is to consider some initial argument x_0, and use it to obtain the functional value $f(x_0) = x_1$. Then use x_1 as the next argument and obtain the next functional value $f(x_1) = x_2$, and so on. This results in the *sequence of iterates* given by Equation (12). See Figure 39. Illustrate this by arranging the following activity:

Student	Amount
S_1	$x_1 = m$
S_2	$x_2 = m + \frac{r}{100}$
$\vdots$	$\vdots$
S_n	$x_n = m \cdot \frac{1-(\frac{r}{100})^n}{1-(\frac{r}{100})}$

Table 10: The Sequence of Iterates in the Finance Game.

Example 14.11.1 (The Finance Game) Choose a group of, say, n students in the class. Arrange the students in a sequence $S_1, S_2, S_3, \ldots, S_n$. An amount of \$m is given to S_1 to be passed onto S_2, who in turn will pass it onto S_3 and so on. The condition is: every student passes an amount m plus (interest of) $r\%$ of the amount he/she received (with the exception of S_1), to another. We then ask: How much the amount of money be when it reaches S_n? This problem gives rise to a sequence of iterates as shown in Table 10.

Iteration as a Recursive Process
Iteration can be introduced as a *recursive* process. It can be viewed as the composition of a function with itself, where the output at one step in the process becomes the input at the next step of the very same process, repeatedly. This could be viewed as a *function machine* (see Figure 52) using the idea of composition of functions.

Convergence/Divergence of Iterates
We are looking for *long-term* behaviours in iteration of functions: what happens to $f^{\circ(n)}(x)$ for some initial value x, as $n \to \infty$? Introduce this by carrying out the next example *with* students.

Example 14.11.2 (Iteration of Linear Functions) Let $f_{a,b}(x) = ax + b$ and consider the problem of finding the iterates of $f_{a,b}$. Form two group of students in the class: G_1, which is given the task of finding the iterates when $| a |< 1$ and G_2, which is given the task of finding the iterates when $| a |> 1$. Specific values of a can be given. Now, the general case can be considered:

$$f_{a,b}^{\circ(n)}(x) = a^n x + b \cdot \frac{(1-a^n)}{1-a}.$$

Then if $| a |< 1$, the first term $\to 0$ and the second term $\to \frac{b}{1-a}$, as $n \to \infty$. So we have a convergence of iterates of $f_{a,b}$. If $| a |> 1$, the first term $\to \infty$ and the second term $\to \frac{b}{1-a}$ as $n \to \infty$, and so we have a divergence. Inquire from each group whether they obtain convergence (and to what number) or not. Ask if all members in the same group got identical results. Mention to students the convergence of iterates leads us to an interesting property — that of a fixed point.

Fixed Point
Mention to students that we are searching for *patterns* in iterative behaviours. One such

pattern is the *fixed point*. Introduce the formal Definition 14.6.2 of a fixed point after the above background. Also mention that, the concept of a fixed point is the result of the *long*-term iterative behaviour of a function and may not always appear as an immediate result of the iterative process. Show the value obtained from group G_1 is the fixed point for the corresponding linear equation. After that it can be mentioned that $f_{a,b}$ has a fixed point at $x = \frac{b}{1-a}$. Have the group G_2 verify that.

Fixed points can have one of several different kinds of behaviours with respect to iteration, when we consider initial values close to the fixed point. Their nature can be *attracting* or *repelling*. These behaviours depend on the sign of the derivative of f at the fixed point, as shown in Table 6. Assign the groups G_1 (for attracting) and G_2 (for repelling) different specific values of a and b for which the existen :e of a fixed point is known apriori and engage them in the activity of finding the nature of the fixed point.

Mention that we can use the method of iteration of functions to calculate roots of polynomials; in the class the case of quadratic polynomials $x^2 - a$, $a > 0$ can be discussed as given in Example 14.6.2.

Students can be told that the study of this Equation (16) goes at least as far back as *Babylonians* who were familiar with this method (now known as *divide-and-average-process*), of using iterations of functions for computing $\sqrt{a}$, $a > 0$. Present calculators use this technique to compute the square root of a number. It can be pointed out that there are numerical methods now for computers (such as the Newton's method) which are a generalization of the Babylonian method. Ask the two groups to verify that giving them different values of a.

The function, the iterates and the derivative, which decides the nature of a fixed point of a function, all have a geometrical intrepretation. Mention that the geometrical approach to iteration of functions provides an alternate way of introducing the concepts discussed earlier.

The Geometrical Approach

Have the figures ready on slides/transperencies for display with the overhead screen and projector.

Iteration as a Graphical Process

The method of iteration is a departure from the algebraic to the geometrical approach. All the figures can be displayed in the class on slides/transperencies using a projector. Use of pointer should be made to explain the figures.

Iteration can be presented as a simple algorithmic process of drawing horizontal and vertical segments, first to the graph of the function under study and then to the diagonal line $y = x$, which reflects it back to the graph again. By repeating this process, a continuous path of alternating horizontal and vertical segments is generated. Establish the function transformation as a *physical* process of moving up from x to the graph of the function (evaluation), left or right to the diagonal (transfer), and then out as $f(x)$ (reflect). Stacked together, one on top of another, a visual picture of the composition of functions appears. In evaluating $g(f(x))$, the first output $f(x)$ becomes the input for $g(x)$. The composition can be traced through from function to function. The final part of the activity introduces this process of composition with a *single* repeating function and then shows how the step-by-step process can be combined onto a single graph. The two-step geometric iteration process is vertical (evaluate) and horizontal (feedback). This process is summarized in Table 7. Have three students come up to the blackboard and direct them to carry out the three steps of the algorithm.

Fixed Point

A fixed point can be identified by intersections of the graph of the function with the diagonal. After indicating this, the geometrical Definition 14.6.3 of a fixed point can be introduced.

The respective property of attracting or repelling of the fixed point can be determined by examining the graphical iteration in the neighbourhood of the fixed point. Mention that the graphical approach has a specific advantage here — it shows us two types of attracting and two types repelling points depending on the path to the fixed point which can be either a staircase or a spiral. These behaviours are closely related to the sign of the slope of the graph of the function at these fixed points, as shown in Table 8.

Attractive and Repelling Fixed Points

Give the graphical definition of attractive and repelling fixed points given in Section 14.6.2. Explain both the cases using corresponding figures, indicating the initial starting point, the path, ... moving along with the assistance of student input. Completed pictures of staircase and spiral behaviour can be used to discuss convergence to or divergence from a fixed point. Activities for practicing the process of graphical iteration on both linear and non-linear functions should be provided to students. For Example 14.6.1, Figure 45 shows the graphical convergence via 'staircase in' to the attractive fixed point $(3, 3)$ from an initial starting point $x = 8.3$. For values of a, b given earlier in Example 14.11.2 for the function $f_{a,b}$, have a member of each group G_1 and G_2 come up to the black (white) board and assist them in carrying out the graphical iteration.

Quadratic (logistic) functions given by equation (17), can be studied as prototypes of non-linear functions. It can be indicated to students that the logistic function has been long known to ecologists[3]. For a particular value of a and x, calculate a few iterates of f_a. This calculation of iterates of quadratic functions soon becomes tedious. Mention to the class that the difficulty in iterating non-linear functions in general was one of the reasons that this area remained dormant for a long period of time and that the advent of modern high-speed computers has motivated its revival.

[3]About two decades ago, Robert May, a mathematical biologist brought the interesting behaviour (on iteration) of this function to the attention of mathematicians.

Iteration using Computer Technology and WWW Tools

This part of the lesson can be carried in a computer laboratory by dividing the students in the class into equally-sized groups and assigning each group a computer to work on. Have the relevant software set-up before the lecture. This includes having the list of relevant URLs saved previously to be used in the laboratory. Test these URLs before the class to check if they are functional (and not out-of-date).

I. The 'Calculating Iterates and Fixed Points of $f_{a,b}(x) = ax + b$ Activity' on the WWW

We repeat the activity of finding the iterates of the linear function $f_{a,b}$ considered in Example 14.11.2 using a computer. The groups can be rearranged from the previous ones. Form three groups: G_1, G_2 and G_3. Groups G_1 and group G_2 are given the task of finding the iterates for specific values of $a, b, \mid a \mid < 1$. Give an initial starting point to each group and inquire from each group whether they obtain convergence (and to what number) or not. G_3 can be given the task of finding the nature of the fixed point, relating the results obtained to that from the algebraic approach, and path to the fixed point *from* the graphics illustrated.

II. The 'Calculating the $\sqrt{2}$ Activity' on the WWW

The goal is to compute $\sqrt{2}$, i.e, the fixed point of equation (16) for $a = 2$. The student (or group) who discovers the fixed point has to announce it to the rest of the class. Arrange the students (or group of students if more than one is assigned a computer to work on) in a sequence: S_0, S_1, S_2 and so on. Give an initial starting point, say, $x_0 = 3.00000$ to S_0, who pass(es) it on to S_1, who compute(s) the first iterate x_1 and pass(es) on the results to S_2, who compute(s) the second iterate x_2 and pass(es) on the results to S_3, and so on.

This activity encourages group work among students where the students see their own results as well as of their peers'. It can keep their interest since the final result (an approximate value of $\sqrt{2}$) can be known to students from their previous experiences (or, e.g., can be computed using a calculator) but the iterate at which the convergence to it shall take place is not known apriori.

III. The 'Calculating Fixed Points of $f_a(x) = ax(1-x)$ Activity' on the WWW

Consider the case where the values for 'a' and 'x' where attractive and repelling fixed points are known apriori. Divide the class into two groups and give them the respective values and some other 'near-by' values of x. The groups are to check the respective behaviour of the fixed points themselves. A short class discussion can then be carried out where each group explains *why* they obtained such results to the other.

14.11.3 Managing the Learning Environment

It need to be observed that class discussions are carried out in decorum and are time efficient, computer-time is properly utilized by students (e.g., no group takes too much time). It is also necessary that any visual equipment is used and placed in a way that is accessible to all students (e.g., proper lighting) and that the computer-network is functioning properly (e.g., is fast enough for computer activities) during the laboratory.

Chapter 15

Conclusion

The patterns of change in nature and in mathematics are unconstrained by conventional categories of thought. In order to make progress, our own patterns of thought must themselves change. To study [the mathematics of change] the scientist of the future will need to combine, in a single integrated world view, aspects of traditional mathematics, modern mathematics, experimentation, and computation.

— Ian Stewart

The field of dynamical systems offers advantages from various viewpoints: pedagogical, epistemological, social, aesthetical, applications to subjects within and outside mathematics. The WWW plays a central role in teaching and learning this view of dynamical systems. In the process of design and development of a WWW–based educational tool for dynamical systems education, it is important to realize throughout, that in the client–server environment of the WWW, the teacher is the "server" and the student is the "client".

It may also be worthwhile to pursue some of the problems in using WWW in education outlined in Section 2.12.2. Apart from that, following are some future directions that might make the use of WWW more fruitful in dynamical systems education, and may be worth pursuing:

- It is important to bring the power of established dynamical systems software environments to the WWW. Such integration is possible by creating CGI gateways to these external programs.

- There is a current shortage of plug–ins for file formats created by various dynamical systems programs, that can be used in conjunction of a browser such as Netscape Navigator; their development is important to support integration of dynamical system program files in WWW documents. The development kit for programming such modules is available on the WWW at the URL
  `ftp://ftp.netscape.com/pub/sdk/plugin/`.

- As it was indicated in Section 8.4.3, for Java to become useful as a programming environment in dynamical systems education, reusable class libraries are needed. Such a class, for example, can be of a generic 3–dimensional nonlinear system of ODEs from which subclasses of say Lorenz and Rössler system can be derived.

- WWW browsers are designed for generic use and in their current state have limitations towards specialized environments. For example, the GUI menu–bar of any of the current browsers does not provide support to any mathematically–specific information (relevant sites, search facilities, etc.). One solution to this is developing customized–browsers. An example is Mathbrowser, which is a WWW browser designed for viewing MathCad documents.

Appendix A

Dynamical Systems: Courses on the WWW

A *home page* is the entry point to a collection of WWW pages on a specialized topic. In this Appendix, we present an outline of a dynamical systems course home page that could be used as a template.

Course #, Title and Term

Announcements

Course–related announcements.

Introduction and Outline

Schedule

Lecture and tutorial schedules. Office hours.

Course Administrators

Name and contact address of course administrators (coordinator, instructor, tutor, marker, secretary).

Resources

Textbook and reference books information (title, publisher, etc.) Course notes (downloadable format). List and URLs of WWW–based lessons.

Computer Programs

List and URLs of available WWW–based programs.

Assessment

Assignments, hints, solutions. Project titles, description, hints. Past/sample examinations, solutions. Examination marks. Final grades.

Search

Local search engine.

Counter

Measures the number of visitations (and thereby judges the utility) to the page.

Authorship

Name, affiliation, e–mail address of the author (of the page).

Bibliography

[1] A. Barcellos. The fractal geometry of Mandelbrot. *College Math. J.*, **15**(98-119), 1984.

[2] Daniel J. Barrett. *NetResearch: Finding Information Online.* O'Reilly & Associates, 1997.

[3] E. T. Bell. *Men of Mathematics.* Simon & Schuster, 1986.

[4] F. H. Bell. *Teaching and Learning Mathematics.* Wm. C. Brown Co., 1978 .

[5] L. Blaine. Theory vs. computation in some very simple dynamical systems. *College Math. J.*, **22**(1):42–44, 1991.

[6] P. Blanchard. Teaching differential equations with a dynamical systems viewpoint. *College Math. J.*, **25**(5):385–393, 1994.

[7] F. E. Browder and S. Mac Lane. The relevance of mathematics. In L. A. Steen, editor, *Mathematics Today: Twelve Informal Essays*, pages 323–350. Springer-Verlag, 1978.

[8] Rikk Carey and Gavin Bell. *The Annotated VRML 2.0 Reference Manual.* Addison Wesley, 1997.

[9] J. Choate. Chaos and fractal software. *College Math. J.*, **22**(1):65–69, 1991.

[10] B. Cornu. Didactical aspects of the use of computers for teaching and learning mathematics. In T. F. Banchoff, et al., editor, *ECM/87–Educational Computing in Mathematics*, pages 187–195. Elsevier, 1988.

[11] National Research Council. *Everybody Counts: A report to the nation on the future of mathemtics education.* National Academy Press, 1989.

[12] National Research Council. *Moving Beyond Myths: Revitalizing Undergraduate Education.* National Academy Press, 1991.

[13] A. Daoudy and J. Hubbard. Itération des polynômes quadratiques complexes. *Comptes Rendus Académie des Sciences, Paris*, **294**:123–126, 1982.

[14] P. J. Davis. Visual theorems. *ESM*, **24**, 1993.

[15] R. L. Devaney. Film and video as a tool in mathematical research. *Math. Intelligencer*, **11**(2):33–38, 1989.

[16] R. L. Devaney. *A First Course in Chaotic Dynamical Sytems.* Addison-Wesley, 1990.

[17] R. L. Devaney. Putting chaos into the classroom. In M. J. Kenny and C. R. Hirsch, editors, *Discrete Mathematics across the curriculum, K-12*, chapter 23. NCTM, 1991.

[18] E. J. Doedel. Nonlinear numerics. Preprint, 1996.

[19] E. Dubinsky. Using computer experiences to implement a Piagetian theory of learning mathematical concepts. Preprint, Purdue University, 1987.

[20] M. B. Durkin. The accuracy of computer algorithms in dynamical systems. *International J. of Bifurcation and Chaos*, 1(3):625–639, 1991.

[21] T. Eisenberg and T. Dreyfuss. On the Reluctance to Visualize in Mathematics. In W. Zimmermann and S. Cunningham, editors, *Visualization in Teaching and Learning Mathematics*, pages 25–37. Math. Assoc. Amer., 1991.

[22] J. Ewing. Can we see the Mandelbrot set? *College Math. J.*, **26**(2):90–99, 1995.

[23] David Flanagan. *Java in a Nutshell.* O'Reilly & Associates, 1996.

[24] G. E. Lohse, et al. A classification of visual representations. *Comm. ACM*, **37**(12), 1994.

[25] Simson Garfinkel and Gene Spafford. *Web Security & Commerce.* O'Reilly & Associates, 1997.

[26] L. Glass and M. C. Mackey. *From Clocks to Chaos: The Rhythms of Life.* Princeton University Press, 1988.

[27] P. Góra and A. Boyarsky. Why computers like Lebesgue measure. *Comput. Math. Applic.*, **16**(4):321–329, 1988.

[28] D. A. Grouws, editor. *Handbook of Research on Mathematics Teaching and Learning*, chapter Technology and Mathematics Education. Macmillan, 1992.

[29] Shishir Gundavaram. *CGI Programming on the World Wide Web.* O'Reilly & Associates, 1996.

[30] H. Jürgens, H.-O. Peitgen and D. Saupe. *The Beauty of Fractals.* Springer-Verlag, 1986.

[31] H.-O. Peitgen, et al. *Fractals for the Classroom: Strategic Activities*, volume **1**. Springer-Verlag, 1991.

[32] H.-O. Peitgen, et al. *Fractals for the Classroom: Strategic Activities*, volume **2**. Springer-Verlag, 1992.

[33] H.-O. Peitgen, H. Jürgens and D. Saupe. *Fractals for the Classroom: Complex Systems and Mandelbrot Set*, volume **2**. Springer-Verlag, 1992.

[34] N. Herscovics and J.C. Bergeron. A constructivist vs. a formalist approach in the teaching of mathematics. In B. Southwell et al., editor, *Eighth International Conference for the Psychology of Mathematics education*, pages 190–196. Sydney, 1984.

[35] P. J. Holmes. Poincaré, celestial mechanics, dynamical systems theory and "chaos". *Physics Reports*, **193**(3), 1990.

[36] J. H. Hubbard and B. H. West. *Differential Equations: A Dynamical Systems Approach–Part I: Ordinary Differential Equations*, volume **5** of *Texts in Applied Mathematics.* Springer-Verlag, 1991.

[37] P. K. Kamthan. Computer Orbits of Chaotic Systems. Master's thesis, Concordia University, 1991.

[38] M. J. Kenny and C. R. Hirsch, editors. *Discrete Mathematics across the curriculum, K-12*, chapter 22 Making connections through iteration. NCTM, 1991.

[39] H. Kocak. Computer experiments in differential equations. In T. F. Banchoff, et al., editor, *ECM/87–Educational Computing in Mathematics*, pages 75–91. Elsevier, 1988.

[40] Ferid Serim & Melissa Koch. *NetLearning: Why Teachers Use the Internet.* O'Reilly & Associates, 1996.

[41] Larry Wall, Randall L. Schwartz and Tom Christiansen. *Programming Perl.* O'Reilly & Associates, 1996.

[42] A. Lasota and M. C. Mackey. *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics.* Springer-Verlag, 1994.

[43] Ben Laurie & Peter Laurie. *Apache: The Definitive Guide.* O'Reilly & Associates, 1997.

[44] M. Atiyah, et al. Responses to "Theoretical mathematics: Toward a culture synthesis of mathematics and theoretical physics", by A. Jaffe and F. Quinn. *Bull. Amer. Math. Soc.*, **30**(2):178–211, 1994.

[45] M. Blomhoj. Modelling of dynamical systems at O-level. In J. de Lange, et al., editor, *Innovation in Maths education by Modelling and Applications*, pages 257–268. Ellis Horwood, 1993.

[46] M. Markus, et al. *Dynamic Pattern formation in Chemistry and Mathematics: Aesthetics in the Sciences.* Max-Planck Institute, 1988.

[47] J. Malkevitch. Tommorrow's geometry. In L. A. Steen, editor, *Heeding the Call for Change: Suggestions for Curricular Action*, pages 47–80. Math. Assoc. Amer., 1992.

[48] B. Mandelbrot. *The Fractal Geometry of Nature.* Freeman & Co., 1982.

[49] R. M. May. Simple mathematical models with very complicated dynamics. *Nature*, **261**:459–467, 1976.

[50] W. Metzler. Another real world approach to complex dynamics. In et al. W. Blum, editor, *Applications and Modelling in Learning and Teaching Mathematics*, pages 405–410. Ellis Horwood, 1989.

[51] Chuck Muschiano and Bill Kennedy. *HTML: The Definitive Guide.* O'Reilly & Associates, 1996.

[52] National Council of Teachers of Mathematics. *Curriculum and Evaluation Standards for School Mathematics.* NCTM, 1989.

[53] S. Papert. *Mindstorms.* Basic Books, 1980.

[54] R. Parris. The root-finding route to chaos. *College Math. J.*, **22**(1):48–55, 1991.

[55] I. Peterson. Searching for a new mathematics. *SIAM Review*, **33**(1), 1991.

[56] J. Piaget and R. Garcia. *Psychogenesis and the History of Science.* Columbia University Press, 1989.

[57] C. Pickover. *Computers, Patterns, Chaos and Beauty.* St. Martin's Press, 1990.

[58] C. Pickover. Mathematics and beauty: several short classroom experiments. *Notices Amer. Math. Soc.*, **38**(3):192–196, 1991.

[59] Richard F. Voss and John Clarke. '1/f noise' in music: Music from 1/f noise. *J. Acoust. Soc. Am.*, 63(1):258–261, 1978.

[60] Laura Parker Roerden. *NetLessons: Web-Based Lessons for your Classroom.* O'Reilly & Associates, 1997.

[61] J. T. Sandefur. Discrete dynamical modeling. *College Math. J.*, **22**(1):13–22, 1991.

[62] J. Schillinger. *The Mathematical Basis of Art.* Philosophical Library, New York, 1948.

[63] M. Senechal. Shape. In L. A. Steen, editor, *On the Shoulder of Giants: New Approaches to Numeracy*, pages 139–181. National Academy of Sciences, 1990.

[64] A. Sierpinska. On 15-17 years old students' conceptions of functions, iterations of functions and attractive fixed points. Preprint 454, Institute of Mathematics, Polish Acad. Sci., Warsaw, Poland, 1989.

[65] A. Sierpinska. *Understanding in Mathematics*, volume **2** of *Studies in Mathematics Education.* The Falmer Press, 1994.

[66] D. Stander. The use of the history of mathematics in teaching. In P. Ernest, editor, *Mathematics Teaching: The State of the Art*, pages 241–246. The Falmer Press, 1989.

[67] L. A. Steen. Pattern. In L. A. Steen, editor, *On the Shoulder of Giants: New Appraoches to Numeracy*, pages 1–10. National Academy of Sciences, 1990.

[68] Lincoln Stein. *How to Set Up and Maintain a Web Site*. Addison Wesley, 1997.

[69] G. Strang. A chaotic search for i. *College Math. J.*, **22**(1):3–12, 1991.

[70] W. P. Thurston. Mathematical education. Preprint, Princeton University, 1991.

[71] W. P. Thurston. On proof and progress in mathematics. *Bull. Amer. Math. Soc.*, **30**(2):161–177, 1994.

[72] T. Vicsek. *Fractal Growth Phenomena*. World Scientific Publishers, 1989.

[73] L. S. Vygotskii. *Mind in society : the development of higher psychological processes*. Harvard University Press, 1978.

[74] J. A. Walsh. The dynamics of Newton's method for cubic polynomials. *College Math. J.*, **26**(1):22-28, 1995.

[75] M. T. Weiss. An early introduction to dynamics. *Amer. Math. Monthly*, pages 635–641, August 1991.

[76] S. Willson. Teaching about fractals. *College Math. J.*, **22**(1):56–59, 1991.

[77] S. Wolfram. Geometry of binomial coefficients. *Amer. Math. Monthly*, 91:566–571, 1984.

[78] S. Wolfram. *Theory and Applications of Cellular Automata*. World Sci. Publ., 1986.

[79] T. J. Ypma. Historical development of the Newton–Raphson method. *SIAM Review*, **37**(4):531-551, 1995.

[80] P. Zorn. Computing in undergraduate mathematics. In L. A. Steen, editor, *Calculus for a New Century: A Pump not a Filter*, pages 233–239. Math. Assoc. Amer., 1988.