BULLETIN OF MATHEMATICS AND STATISTICS RESEARCH
A Peer Reviewed International Journal,
Contents available on www.bomsr.com
Vol.4. S1.2016; ISSN: 2348-0580
Email:editorbomsr@gmail.com
RADMAS– 2016

MATHEMATICAL MODELLING IN COMPUTERS

FIROZE BAGUM
Department of Computer Science,
St. Joseph's College for Women (A), Visakhapatnam

ABSTRACT
This paper presentation throws some insights into the involvement of Numerical analysis to the problems faced in animation Industry and in Google's optimal page searching. The numerical analysis done over a sample of data is modelled into a mathematical problem and is solved. These mathematical solutions are helpful in solving the real world problems in a simpler way thereby reducing the complexity of the problem.

Introduction

Numerical methods play an important role in modern science. Scientific exploration is often conducted on computers rather than laboratory equipment. While it is rarely meant to completely replace work in the scientific laboratory, computer simulation often complements this work.

For example, the aerodynamic simulation of two **NASCAR** autos pictured in **figure 1.1(a)** requires the numerical solution of **P**artial **D**ifferential **E**quations **(PDEs)** that model the flow of air past the car.

An auto body must be smooth and sleek, so it is often modelled using cubic (or higher order) spines. Similar computations are done in designing aircraft.

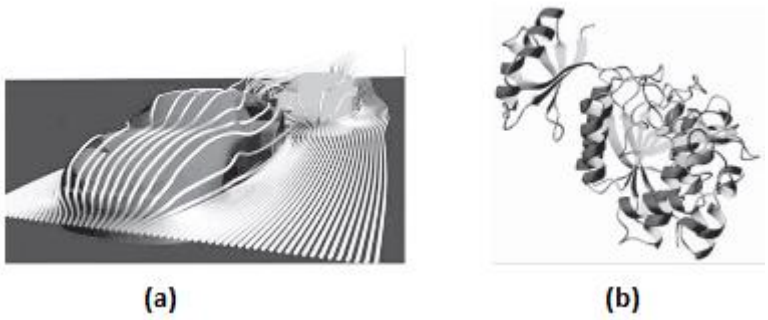

Figure 1.1. (a) A simulation of two NASCAR autos depicts the streamlines of air produced as a car drafts and is about to pass another. (Simulation performed with STAR-CCM+.) **(b)** Solving protein-folding models utilizes numerical optimization.

The mathematical formulation usually represents only a *model* of the actual physical situation, and it is often important for the numerical analyst or computational scientist to know something about the origin of the model; in fact, numerical analysts sometimes work directly with scientists and engineers in devising the mathematical model. This interaction is important for a number of reasons.

First,

Many algorithms do not produce the exact solution but only an approximate one. An understanding of the origin of the problem is necessary to determine what constitutes an acceptably good "approximate" solution: an error of a few centimetres might be acceptable in locating an enemy

BULLETIN OF MATHEMATICS AND STATISTICS RESEARCH
A Peer Reviewed International Journal,
Contents available on www.bomsr.com
Vol.4. S1.2016; ISSN: 2348-0580
Email:editorbomsr@gmail.com
RADMAS– 2016

tank, but it would not be acceptable in locating a tumour for laser surgery!

Second,

Even if the algorithm theoretically produces the exact solution, when implemented on a computer using finite-precision arithmetic, the results produced will most likely be inexact. Part of numerical analysis is the understanding of the impact of finite-precision computations on the accuracy of results.

Here we present two amongst many applications that involve numerical computation and come from the mathematical modelling of various processes in the industry of Computers.

Modelling in Computer Animation

Many of the computer generated graphics that dominate the silver screen are produced with **dynamic simulation**; that is, a model is created, often using the laws of physics, and numerical methods are then used to compute the results of that model. In this section, we will look at the role of numeric's in animation that appeared in the 2002 film *Star Wars: Episode II Attack of the Clones*.

In particular, we will take a careful look at some of the special effects used to digitally create the character of Yoda, a Jedi master who first appeared as a puppet in the Star Wars saga in the 1980 film, *The Empire Strikes Back*. In the 2002 film, Yoda was digitally created, which required heavy use of numerical algorithms.

A key aspect of creating a digital Yoda involves producing believable movement of the character. The movement of Yoda's body is described using **key-frame animation**, in which a pose is specified at particular points in time and the computer automatically determines the poses in the intervening frames through ***interpolation***.

Animators have many controls over such movement, with the ability to specify, for instance, velocities and tangents of motion. While animators indicate the movement of Yoda's body, the computer must determine the resulting flow of his robe.

A Model Robe

Figure 1.2. Stages of simulation in the animation of the digitally created Yoda in the fight scene with Count Dooku in *Star Wars Episode II*. Two layers of Yoda's clothing, seen in (b) and (c), were computed separately. A process known as collision detection ensured that the inner layer of clothing did not intersect the outer robe and become visible.

Referring to **figure 1.2**, we see that the robe is represented with triangles and the motion of each vertex of the robe must be determined. Each vertex is modelled as a particle, which in this context is a point like object that has mass, position, and velocity, and responds to forces, but has no size.

BULLETIN OF MATHEMATICS AND STATISTICS RESEARCH
A Peer Reviewed International Journal,
Contents available on www.bomsr.com
Vol.4. S1.2016; ISSN: 2348-0580
Email:editorbomsr@gmail.com
RADMAS– 2016

The motion of a particle is governed by Newton's second law, which is expressed mathematically by the equation

$$\mathbf{F} = m\mathbf{a} = m(d^2\mathbf{y}/dt^2) \quad (1.1)$$

where **y** is a distance function of time *t*. Note that the equations involve vector valued functions since our computations are performed in three dimensions. Since a particle has mass (*m not Equal to 0*), equation (1.1) can be rewritten as the second-order **O**rdinary **D**ifferential **E**quation **(ODE)**

$$\frac{d^2\mathbf{y}}{dt^2} = \frac{\mathbf{F}}{m} \quad (1.2)$$

This **ODE** is part of an initial value problem since the state of the particle at some initial time is given. In the case of a movie, this is where the scene (which may be several seconds or a fraction of a second in duration) begins.

To keep the shape of the robe, pairs of neighbouring particles are attached to each other using a spring force. Hooke's law states that a spring exerts a force *Fs* that is proportional to its displacement from its rest length x_0. This is expressed mathematically as

$$Fs = -k(x - x_0),$$

where *x* denotes the current position of the spring and *k* is the spring constant. For simplicity, we have stated the one-dimensional formulation of Hooke's law, but to model the Jedi's robe a three-dimensional version is used. Many other forces are computed to animate Yoda's robe, including gravity,wind forces, collision forces, friction, and even completely made-up forces that are invented solely to achieve the motion that the director requests. In the simplest of cases, an ***analytic solution*** of the model may exist.

In computer animation, however, the forces acting on the particle constantly change and finding an analytic solution for each frame—if even possible—would be impractical. Instead, numerical methods are used to find approximate solutions by simulating the motion of the particles over discrete time steps.

2. Modelling a Web Surfer and Google

Submitting a query to a search engine is a common method of information retrieval. Companies compete to be listed high in the rankings returned by a search engine. In fact, some companies' business is to help raise the rankings of a paying customer's web page. This is done by exploiting knowledge of the algorithms used by search engines. While a certain amount of information about search engine algorithms is publicly available, there is a certain amount that is proprietary. Here we discuss how one can rank web pages based on content and is introductory in nature.

Here we will consider a simple vector space model for performing a search. This method does not take into account the hyperlink structure of the World Wide Web, and so the rankings from such a model might be aggregated with the results of an algorithm that does consider the Web's hyperlink structure.

The Vector Space Model

The vector space model consists of two main parts:

1. a list of documents and
2. a dictionary.

BULLETIN OF MATHEMATICS AND STATISTICS RESEARCH
A Peer Reviewed International Journal,
Contents available on www.bomsr.com
Vol.4. S1.2016; ISSN: 2348-0580
Email:editorbomsr@gmail.com
RADMAS– 2016

The list of documents consists of those documents on which searches are to be conducted, and the dictionary of terms is a database of keywords. While the dictionary of terms could be all the terms in every document, this may not be desirable or computationally practical. Words not in the dictionary return empty searches.

With a list of *n* documents and *m* keywords, the vector space model constructs **an *m* by *n document matrix A*** with

$$a_{ij} = 1, \text{ if document } j \text{ is relevant to term } i,$$
$$a_{ij} = 0 \text{ otherwise} \qquad (1.4)$$

When a query is issued, a query vector $\mathbf{q} = (q_1, \ldots, q_m)^T$ is then formed, with $q_i = 1$ if the query includes term *i* and $q_i = 0$ otherwise.

The "closeness" of the query to each document is then measured by the cosine of the angle between the query vector **q** and the column of *A* representing that document.

If $\boldsymbol{a}_j$ denotes the j^{th} column of *A*(and if it is not entirely zero due to document *j* containing no keywords), then the angle ϑ_j between **q** and $\boldsymbol{a}_j$ satisfies

$$\cos(\vartheta j) = \frac{\mathbf{a}^T_j \mathbf{q}}{||\mathbf{a}_j||2||\mathbf{q}||2} \qquad (1.5)$$

The dictionary of terms and documents used in a three-dimensional example.

Table 1.1

Dictionary	**Documents**	
electric	I	the art of war
fencing	II	the fencing master
foil	III	fencing techniques of foil, epee, and sabre
	IV	hot tips on building electric fencing

Since a larger value of the cosine implies a smaller angle between the vectors, documents are ranked in order of relevance to the query by arranging the cosines in descending order.

A major difficulty in forming the document matrix is determining whether a document is relevant to a term.

There are a variety of ways of doing this.

1. For instance, every document can be searched for each keyword. A document is deemed relevant to those keywords that are contained within it. This requires considerable computational expense when one faces a large number of documents and keywords.
2. An alternative way that is employed by some search engines is to read and perform analysis on only a portion of a document's text.

For example, Google and Yahoo! pull only around 100K and 500K bytes, respectively, of a web page text. Other choices that must be made include whether to require exact matching of terms or to allow synonyms, and whether or not to count word order. For example, do we treat queries of boat show and show boat as the same or different? All of these choices impact which documents are deemed most relevant. To illustrate the method, we give a simple example, in which a document is deemed relevant if its title contains the keyword exactly.

BULLETIN OF MATHEMATICS AND STATISTICS RESEARCH
A Peer Reviewed International Journal,
Contents available on www.bomsr.com
RADMAS– 2016
Vol.4. S1.2016; ISSN: 2348-0580
Email:editorbomsr@gmail.com

Searching in a Tiny Space

Suppose that our dictionary contains three keywords “electric”, “fencing”, and “foil” and that there are four documents entitled “the art of war”, “the fencing master”, “fencing techniques of foil, epee, and sabre”, and “hot tips on building electric fencing”. We have written the document titles and dictionary terms in lower case to avoid questions about matching in the presence of such a factor. The information is listed in **table 1.1** for easy reference.

To form the document matrix *A* where rows 1, 2, and 3 correspond to the terms “electric”, “fencing”, and “foil”, respectively, while columns 1, 2, 3 and 4 correspond to documents I, II, III, and IV, respectively we use **(1.4)** to obtain

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

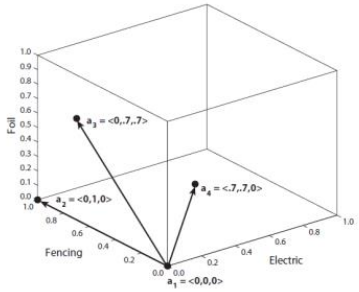

Figure 1.10 depicts the columns of *A* (each except the first normalized to have length 1) as vectors in three-space.

Suppose our query is fencing. Then the query vector is $\mathbf{q} = (0, 1, 0)^T$, and we can see by inspection that it is identical to column 2 of *A*. Thus, document II would be deemed most relevant to this query. To determine the rankings of the remaining documents, we compute the cosine of the angle between **q** and every other nonzero column of *A* using **(1.5)** to find

$$\cos(\theta_3) = \frac{\mathbf{a}_3^T \mathbf{q}}{\|\mathbf{a}_3\|_2 \|\mathbf{q}\|_2} = \frac{1}{\sqrt{2}},$$

$$\cos(\theta_4) = \frac{\mathbf{a}_4^T \mathbf{q}}{\|\mathbf{a}_4\|_2 \|\mathbf{q}\|_2} = \frac{1}{\sqrt{2}}.$$

Thus documents III and IV would be ranked equally, behind document II.

Google’s Page Rank

Real-world search engines deal with some of the largest mathematical and computer science problems in today’s computational world. Interestingly, “Google” is a play on the word “**googol**”, the number $\mathbf{10^{100}}$, reflecting the company’s goal of organizing all information on the World Wide Web.

When you submit a query, such as numerical analysis, to Google, how does the search engine

BULLETIN OF MATHEMATICS AND STATISTICS RESEARCH
A Peer Reviewed International Journal,
Contents available on www.bomsr.com
Vol.4. S1.2016; ISSN: 2348-0580
Email:editorbomsr@gmail.com
RADMAS– 2016

distinguish between the web page listed first and the one listed, say, 100th? There are various factors that play into this decision, one of which is the web page's relevance to your query, as discussed previously.

Another important component, however, is the "quality" or "importance" of the page; "important" pages, or pages that are pointed to by many other pages, are more likely to be of interest. An algorithm called ***Page Rank*** is used to measure the importance of a web page. This algorithm relies on a model of web-surfing behaviour.

As with any model of reality, Google's model of web-surfing behaviour is **an approximation**. An important feature of **Page Rank** is its assumption about the percentage of time that a surfer follows a link on the current web page. The exact assumptions that are made are proprietary, but it is believed that the *Page Rank algorithm* used by Google assumes that a surfer follows a link on a web page about 85% of the time, with any of the links being equally likely. The other 15% of the time the surfer will enter the URL for another web page, possibly the same one that is currently being visited.

When you submit a query to a search engine, an ordered list of web pages is returned. The pages are ranked by their relevance to your query and also by the quality of the page. Consider the small network of web pages in **figure 1.12.**

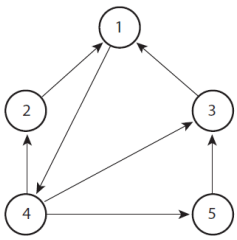

Figure 1.12. Small network of web pages

We will assume this is the set of web pages indexed by our search engine. Each vertex in the graph is a web age.A directed link is drawn from web page *i* to web page *j* if web page *i* has a link to web page *j.*

So, we can see, for example, that web page 1 links to web page4. The *Page Rank algorithm*, as proposed by Larry Page and Sergey Brin, assumes that a surfer follows a link on a web page 85% of the time, with any of the links being equally likely. The other 15% of the time the surfer will enter the URL for another web page, possibly the same one that is currently being visited, again with all pages being equally likely.

Let *Xi* (*t*) denote the probability of being at web page *i* after the surfer takes *t* steps through the network. We will assume the surfer starts at web page 1,so that *X*1(0) = 1 and *Xi* (0) = 0 for *i* = 2, 3, 4, 5.

(a) Find *Xi* (1) for $1 \le i \le 5$.

(b) Find *Xi* (2) for $1 \le i \le 5$.

As a measure of the quality of a page, **Page Rank** approximates $\lim_{t\to\infty} Xi\,(t)$ for all *i* .

These assumptions about surfing behaviour affect not only the accuracy of the model but also the efficiency of numerical techniques used to solve the model. Keep in mind that Google indexes billions of web pages, making this one of the largest computational problems ever solved!

RADMAS– 2016

BULLETIN OF MATHEMATICS AND STATISTICS RESEARCH
A Peer Reviewed International Journal,
Contents available on www.bomsr.com
Vol.4. S1.2016; ISSN: 2348-0580
Email:editorbomsr@gmail.com

There are certainly many other purposes to which computers can be put besides the numerical solutions of mathematical problems . Computers and Mathematics are mutually helpful in solving many complex REAL WORLD problems with ease.

References: http://press.princeton.edu/chapters/s9763