AN ADAPTIVE BOOK APPROACH TO TEACHING AND LEARNING LINEAR ALGEBRA

Ananda D. Gunawardena
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
guna@cs.cmu.edu

Surender .K. Jain
Department of Mathematics
Ohio University
Athens, OH 45701
jain@math.ohiou.edu

ABSTRACT

This paper describes our experiences in teaching a linear algebra course in an interactive environment using a new technology platform known as the Adaptive Book (AB). AB allows all core content and supplements to be integrated in a meaningful way and provides user tools to markup the content. A markup consisting of highlights, annotations, web and file links can be reused, edited and shared. We shall also discuss ways in which real world applications can be introduced to make the linear algebra course more fun and effective in this adaptive book learning environment.

BACKGROUND

Linear Algebra is a discipline where theory blends with applications nicely to provide an opportunity for developing a highly interactive course. The availability of computer algebra systems (CAS) such as Matlab [1], Mathematica [2] and Maple [3] allow students to solve complex computational problems and visualize abstract concepts in Linear Algebra. Matlab[1] is a computer algebra system that is being widely used by the scientific community since 1980's. Matlab comes with a rich set of commands that allow users to enter a single or multiple command(s) to perform tasks that otherwise would have required good programming skills. Furthermore, Matlab's programming environment provides basic programming constructs so advanced users can write few lines of Matlab code to solve difficult problems. This allows many opportunities for experimentation and conjecturing of interesting matrix properties.

The tools for teaching mathematics using technology have come a long way since the inception of the web and the introduction of computer algebra systems. Platform independent Programming languages like Java has allowed the opportunity to create interactive Java applets to demonstrate linear algebra concepts and share them with remote users. In this paper we will discuss a new and innovative way to conduct a linear algebra course using a state-of-the-art software platform known as the adaptive book. The adaptive book (AB) can be used as a teaching, or learning tool. Furthermore, the AB software comes with a sophisticated markup manager that allows various markups to be created, reused, edited and shared among its users. The ability to connect Matlab and other tools to AB provides the additional benefit of keeping all course related content under one platform.

ADAPTIVE BOOK

Adaptive Book [4] is a new client/web based hybrid software platform developed to provide interactive learning experience to its users. Adaptive Book (AB) technology platform consists of two components; authoring system and a display/markup manager. Authoring system allows selecting of appropriate content, including media from an IMS compatible repository to create a customized lesson profile. A profile is defined as a collection of core text content and all other supplementary material including images, and video clips grouped into a learning module. A profile and the related content are downloaded into the student machine and viewed using Adaptive Book display manager. A sample screen shot of the Adaptive Book is shown in Figure 4. AB interface contains navigation tools, bookmark features, and built-in note taking facilities as well as links to all supplements. The content can be personalized using AB's markup tools. A markup is defined as a collection of highlights, annotations and web links grouped into a semantically meaningful package. Markup can also contain images, java applets, or video clips integrated directly into the content. Markups can be shared, edited or organized by topic or any other criteria. For example, student using the AB can create a digital markup of his work by highlighting, annotating and linking content. Users can save markups by assigning a title, author information as well as editable or non-editable status. For example, an instructor may wish to send a lesson plan markup as a non-editable markup. A lesson plan markup may contain highlighted pages, annotated sections and web or file links that are related to the lesson. A student receiving a lesson markup may use it to do a quick tour through the content to understand the concept well. Several studies have shown that students learn more (65% more) when core material is presented in an annotated mode [5]. In summary, Adaptive Book provides highly interactive tools for its users to personalize learning experience and share them with others when necessary.

OUR APPROACH

Our linear algebra course is an introductory course that adheres to the curriculum guidelines recommended by International Linear Algebra Society. The standard topics such as Linear Systems, Matrix Algebra, Vector Spaces, Eigenvalues/Eigenvectors, linear transformations and their applications are covered in this course. The course is taught in a lab setting where all students have access to computers. This computer class setting encourages an interactive learning approach where discussion of concepts is supported by computer experiments using Matlab. Furthermore, the availability of Adaptive Book allows the opportunity to use multiple representation principle [6] to make students better understand abstract mathematical concepts. Multiple representation principle is based on the theory that students learn better when concepts are presented in multiple forms including concept demonstrations, text and drill activities. The benefit of a multiple representation principle is well documented in [5][6]. The adaptive book in linear algebra contains text, concept demos, Matlab drills and concepts tests. We conduct the class in an interactive setting and the class meets three times a week.

Before each lecture, we prepare a lecture markup by highlighting important sections of the book, annotating additional material, and linking to other resources on the web (Figure 5). The markup then is saved as an editable or non-editable markup (Figure 6). An editable markup can be edited by the receiver and personalize to fit his or her learning needs. An example of a non-editable markup may be a very carefully planned tour through the digital content intended to reinforce a concept. A markup is then sent to the entire class as an email attachment or placed on a course management system (Figure 7) such as eCollege, BlackBoard or WebCT. Before the class, each student download the markup object that contains highlighted text, annotations, and web links, and/or other supplementary material required to prepare for the lecture. All students are encouraged to go through the markup as a preparation of the lecture/lab. Markups allow students to focus on specific parts of the text that are important to the lecture.

In the classroom/lab, after introducing students to basic linear algebra concepts, we demonstrate concepts using graphical demos (Figure 9). These concept demos contain demonstrations of linear algebra algorithms as well as animations to help understand difficult concepts. It is quite common that most students have a good grasp of the concepts after presentation of lecture combined with concepts demos. Next step in our interactive class is to practice drills to reinforce the concepts. The Matlab drills (Figure 8) are carefully selected exercises that cover all-important concepts. Many drills are available to practice the skills. We present drills at all levels of difficulty and hope that completing the drills with help provided in the AB allows students to master concepts easily. As extended classroom/lab activities, students are asked to solve practical real world problems as projects by applying linear algebra concepts. Projects that are due in 1-2 weeks and require activities that involve careful planning and programming activities in Matlab. Students draw from many examples learned in drills to help them solve projects. Completing a project requires the mastery of individual concepts and the ability to combine them to form a complex solution. For example, a student applying Singular Value Decomposition (SVD) to perform data compression must not only understand how to extract image data into a matrix form, but also how to apply SVD and reconstruct the matrix to represent the compressed image. We have presented many of these applications in our textbook, Linear Algebra-An Interactive Approach [6]. While there are many interesting examples in the book, we will explain an application of SVD to data compression below.

DATA COMPRESSION ALGORITHM USING SVD

Data compression is a problem faced by many in everyday life. There are many real world algorithms for compressing data. Some are lossless (original data is recovered) and some are lossy (original data is not recovered – however the image is "close" enough). An example of a lossless compression is applying a program like WinZip to compress a data file. A lossy compression algorithm such as JPEG requires the understanding of rigorous mathematical theories. We describe here a lossy compression algorithm based on singular value decomposition (SVD). SVD allows the decomposition of the matrix into a form $A = U S V^T$, where U and V are orthogonal matrices and S is a diagonal matrix containing the singular values of A. The matrices U and V can be chosen so that diagonal elements of S are ordered in decreasing order. First we will start with a bitmap image of size 64x64 pixels. Since Matlab student version has a size limit, we decided to use a 64x64 image. Otherwise one may even use 128x128 image for better clarity. Colored bitmap images store information in an n by n by 3 matrix, where the third dimension is for RGB (Red-Green-Blue) components of the image, each of which is a n by n matrix. The entries of these matrices are in the range 0 to 255 representing the color depth of the image.

We will start with the image shown in Figure 1. This is a color image that can be represented by a 64x64x3 matrix A. The matrix A is a 3D matrix combining three 64x64 matrices for the (Red, Green, Blue) portions of the image.

Step 1: We can read this image into Matlab using the following command.

A=imread('c:\temp\castle64','bmp');

Step 2: We can then extract R, G and B matrices and apply SVD to each 64x64 matrix individually.

R = A(:,:,1);G = A(:,:,2);B = A(:,:,3);

Step 3: Matlab svd command allows us to find singular value decomposition of R, G, and B as follows.

[ur,sr,vr]=svd(double(R)); [ug,sg,vg]=svd(double(G)); [ub,sb,vb]=svd(double(B));

Step 4: The rank-k approximation to each of the matrices R, G and B is given by the sum

$U_1\lambda_1V_1 + U_2\lambda_2V_2 + U_3\lambda_3V_3 + \ldots\ldots U_k\lambda_kV_k$, where λ_r is the r-th singular value and U_i (or V_i) is the I^{th} column of U (or V).

We will find the sums (for rank-8) for each of the colors using the following Matlab commands.

RK=sr(1,1)*ur(:,1)*transpose(vr(:,1)); for k=2:8, RK=RK + sr(k,k)*ur(:,k)*transpose(vr(:,k)); end

GK=sg(1,1)*ug(:,1)*transpose(vg(:,1));for k=2:8, GK=GK + sg(k,k)*ug(:,k)*transpose(vg(:,k)); end

BK=sb(1,1)*ub(:,1)*transpose(vb(:,1));for k=2:8, BK=BK + sb(k,k)*ub(:,k)*transpose(vb(:,k)); end

Step 5: Now we combine RK, GK and BK to find the 64x64x3 matrix and graph the rank-8 approximation of the image.

AK(:,:,1)=RK; AK(:,:,2)=GK; AK(:,:,3)=BK; imagesc(AK);

Figure's 2 and 3 shows rank-1 and rank-8 approximations of the original image in Figure 1.

CONCLUSION

Linear Algebra is an ideal course to integrate technology to enhance teaching and learning. As described in this article, there are many opportunities to design interesting application problems that can be solved by applying linear algebra concepts and with the help of Matlab. Other computer algebra systems such as Mathematica and Maple also have significant computational and graphing tools to experiment and learn. Using the markup tool set of the Adaptive Book, interactivity can be created on the text. The platforms such as Adaptive Book help author, share, and extend highly interactive material. Concepts can be introduced in multiple forms, combining text, graphics and computational elements. We can develop Matlab .m files for greater understanding of concepts. We can use tools such as Maplets [3] to create interactive environments on the web. There are plenty of things to do to make linear algebra fun and effective. We have presented this complete interactive platform for learning and teaching linear algebra in our textbook [7] published in 2003. The textbook using the Adaptive Book platform, combines concept demos, Matlab drills, and application projects and tests all under one platform. Using this platform, you can create markups and share them with students. We believe combining all components of the course under the Adaptive Book platform, and using markup tools to highlight, annotate and link content, and using Matlab to compute and visualize, make teaching and learning of Linear Algebra fun and effective.

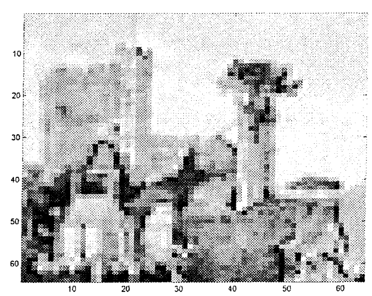

Figure 1 – Original 64x64 Image

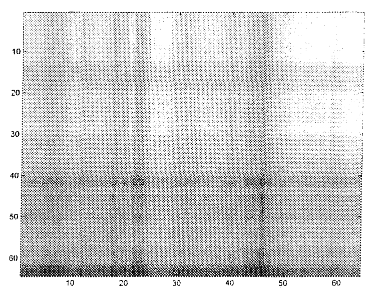

Figure 2 – Rank-1 Approximation

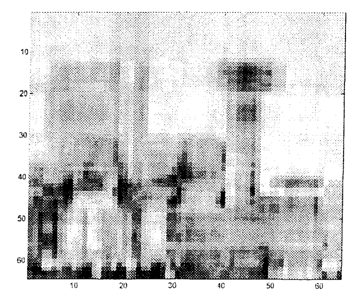

Figure 3 – Rank-8 Approximation

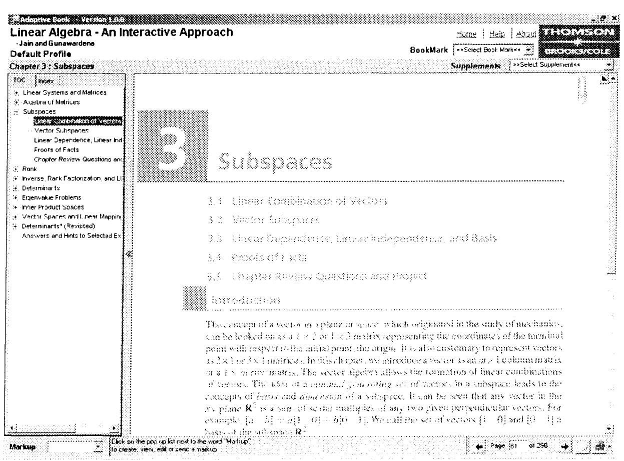

Figure 4 – Adaptive Book Interface

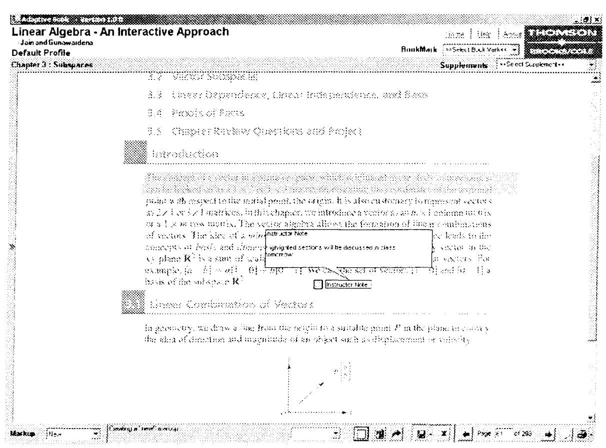

Figure 5 –Creating a Markup using highlights & annotations

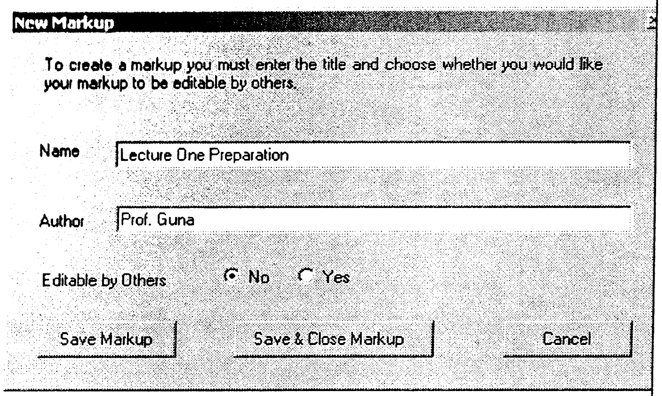

Figure 6 – Saving a Markup

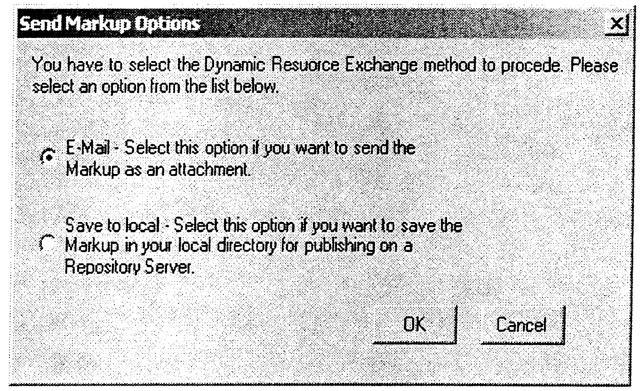

Figure 7 – Sending a Markup

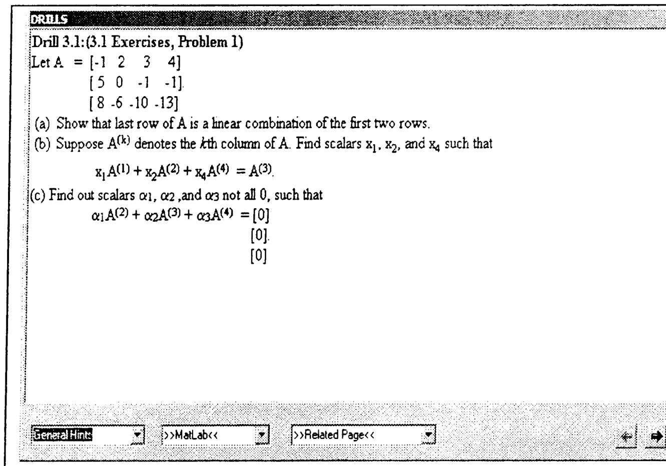

Figure 8 – A Sample Drill Window

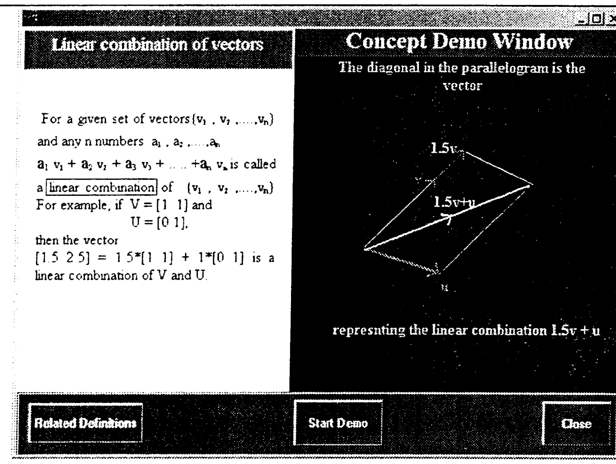

Figure 9 – A Sample Demo Window

References

[1] Matlab, the language of technical computing, http://www.mathworks.com/

[2] Wolfram Research, Champaign, IL, http://www.wolfram.com/

[3] Maple Software, http://www.maplesoft.com/

[4] Adaptive Book, TextCentric Technologies, http://www.textcentric.com/products/adaptive_content_tech.html

[5] Mayer, R.E., Systematic thinking fostered by illustrations in scientific text. Journal of Educational Psychology (1989), 81:240-6.

[6] Mayer, R.E.&Gallini, J.K, When an Illustration worth thousand words, Journal of Educational Psychology (1990), 82:715-26.

[7] Jain, S.K, Gunawardena, A.D., Linear Algebra – An interactive Approach, Brooks-Cole/Thomson Publishing, ISBN 0-534-40915-6, 2003