An interactive Java-based educational module in electromagnetics

M. Chetty[1], S. Hu[2] and J. Bennett[2]
[1]*Gippsland School of Computing and Information Technology, Monash University, Churchill, Australia*
[2]*Department of Electrical and Computer Systems Engineering, Monash University, Clayton, Australia*
E-mail: Madhu.Chetty@infotech.monash.edu.au

Abstract This paper explains the design, development and implementation of a web-based educational module for an introductory electromagnetic (EM) course at Monash University. It contains tutorials, interactive simulation and animation. The two most important sections of the module, namely 'electric dipole' and 'experimental field mapping', are described here. Both these sections are interactive and with the help of visual graphical displays and audio files they 'stimulate' the sight and sound senses for understanding. The module can act as an instructional aid and helps not only in understanding the fundamental concepts but also in providing a greater appreciation of the applications of EM theory. The responses from interactive simulation are displayed directly on the client browser. The overall package is developed using Java, HTML, CGI scripts written in Perl and MATLAB.

Keywords electromagnetics; Java; web-based education

Many universities are feeling the need for some form of substitute for a human tutor because a student is often unable to approach a tutor as and when a difficulty is experienced or a need for discussion arises. As a result, many applications have been developed in the field of engineering and science education to meet this objective of educators who see the need to improve the way introductory courses are taught to engineering students.[1–5] With the advent of computer technology, a medium for such a substitute is now available in the form of the Internet or world wide web. The web-based material is accessible anywhere, anytime, as often as required, and to a large number of people simultaneously. These advantages of almost unrestricted access alone makes the web a valuable medium even if expert human help were abundant because it would allow students to make 'silly' mistakes, and that too without the embarrassing presence of a tutor.[6] Thus, the very impersonal nature of the web is a persuasive reason for using it as a medium for structured instruction even if the amount of feedback offered to the students is very limited in quantity and quality. Further, the web can provide a highly supportive learning environment to distance education.[7]

Given these obvious advantages of the web, this paper proposes a web-based educational module for electromagnetics (EM), which is one of the cornerstones of electrical engineering practice. The subject is perceived as one of the most difficult courses due to its extensive mathematical content and physical/intuitive concepts. Further, the subject matter relies heavily on vector mathematics with closed-form field solutions only available for highly symmetric, idealised geometries.[8] As a result, students have difficulties in understanding and visualising the subject material using conventional teaching methods.

Traditionally, EM has been introduced to students by limiting any illustrations to textbook-type examples that can be easily replicated on the whiteboard. The use of interactive multimedia tools can allow students to have a graphical display of abstract quantities such as fields, potentials etc. and to visualize these; thus making the subject more exciting.[8] Another objective in developing a multimedia-based teaching tool is to give students a sense of the 'bigger picture'. To achieve these objectives, both tutorial-type and simulation-type modules across the Internet are found useful. The tutorial-type modules are primarily static and are intended to complement the lectures. Simulation-type modules support interactive visualization with the key parameter variations affecting the performance of a component or system and help to form mental models of the theory, thus improving problem-solving skills. The module presented in this paper is designed to contain both components, i.e. tutorial and simulation.

Efforts to develop both tutorial- and simulation-type modules have been made. An example of one such EM tutorial can be found where the tutorial contains a series of hyperlinks which take the user to different 'static' HTML pages containing explanatory text, figures and equations.[9] However, a tutorial with such an abundance of hyperlinks can cause the users to be lost within such a section. Another example of an EM module dealing with simulation is Visual Electromagnetics for Mathcad which is arranged in an electronic book format.[11] However, it has to be used in conjunction with the text. Further, it requires Mathcad 8.0 to be installed and can be viewed only locally. Other simulation-type modules for EM are the Graphically Driven Electromagnetics (GDE), and Contact.[8,10] Both GDE and Contact have a Graphical User Interface (GUI) to make it user friendly, and are capable of displaying graphical results. However, these interactive modules cannot be easily accessed from the web, nor do they provide both tutorial and simulation component.

This paper deals with the design, development and implementation of a web-based educational module containing not only tutorial and interactive simulation but also animation effects. The two sections of the module, namely, 'electric dipole' and 'experimental field mapping' being presented here can easily be extended to other EM topics. The module has multimedia features which, with the help of visual displays and audio files, 'stimulate' the sight and sound senses for understanding. The responses from interactive simulation are displayed directly on the client browser. The module is developed using Java, HTML, CGI scripts written in Perl and MATLAB.

The electromagnetics module

As shown in Fig. 1, the main page of the EM module is split into two frames. The smaller frame on the left contains the 'contents' of the module and is used for navigation. For example, under the *Chapter 2* heading is a hyperlink to 'Coulomb's Law & Electric Field' which takes the student to the tutorial and simulation sections of this topic. Due to the 'mouse over effect' done using JavaScript, a small red arrow

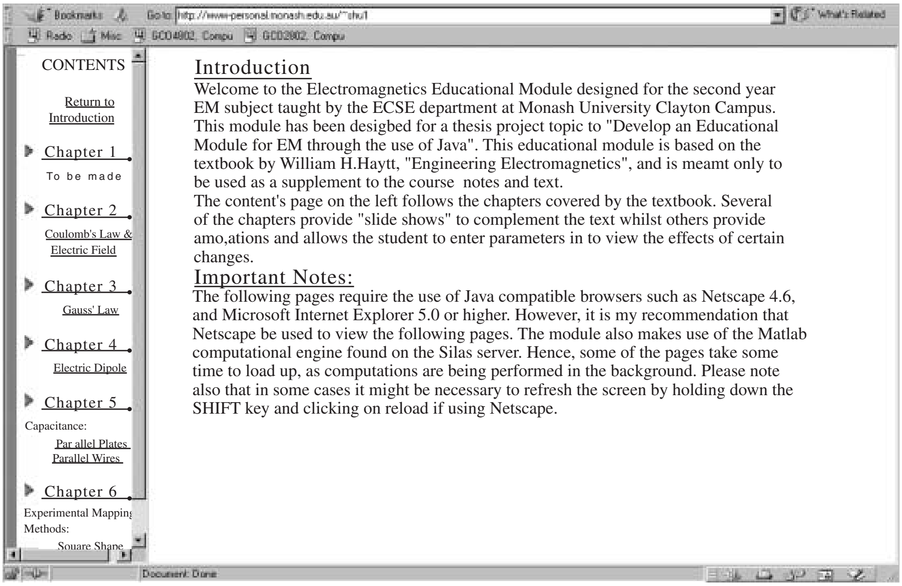

Fig. 1 *Main page of the educational module.*

appears next to each hyperlink that the mouse passes over. It essentially involves loading two images, namely an ‘on’ image when the mouse passes over the hyperlink and an ‘off’ image otherwise. The larger frame on the right provides the ‘introduction’ to the module, a brief background on the purpose of the module and general advice. We next look briefly at the tutorial section before we discuss the interactive simulation and animation effect.

Tutorial section

Fig. 2 is the screen capture of one of the slides in the slideshow. The user is presented with three buttons: Forward, Back, and Reload. Each button also has an associated small audio file that starts playing whenever the button is clicked. Paint Shop Pro 6.0 has been used to design each slide.

Before taking up interactive simulation studies and implementing animation to visualize the results and/or analysis, the tutorial section can be completed to strengthen the concepts and understanding of a topic. The various topics such as Coulomb’s Law and electric field, Gauss’s Law, electric dipole, parallel plate capacitance, TEM transmission, etc are all supported by tutorials. As mentioned earlier, these tutorials are mainly static without any interactivity or animation. When the user clicks, for example *Coulomb’s Law and Electric Field*, in the left frame of the main page shown in Fig. 1, an HTML page with an embedded Java applet’s class file ***ESlide.class*** is loaded. The section of the code that causes this loading is the ⟨**APPLET**⟩ tag shown in Fig. 3 below. As can be seen from this figure, the code

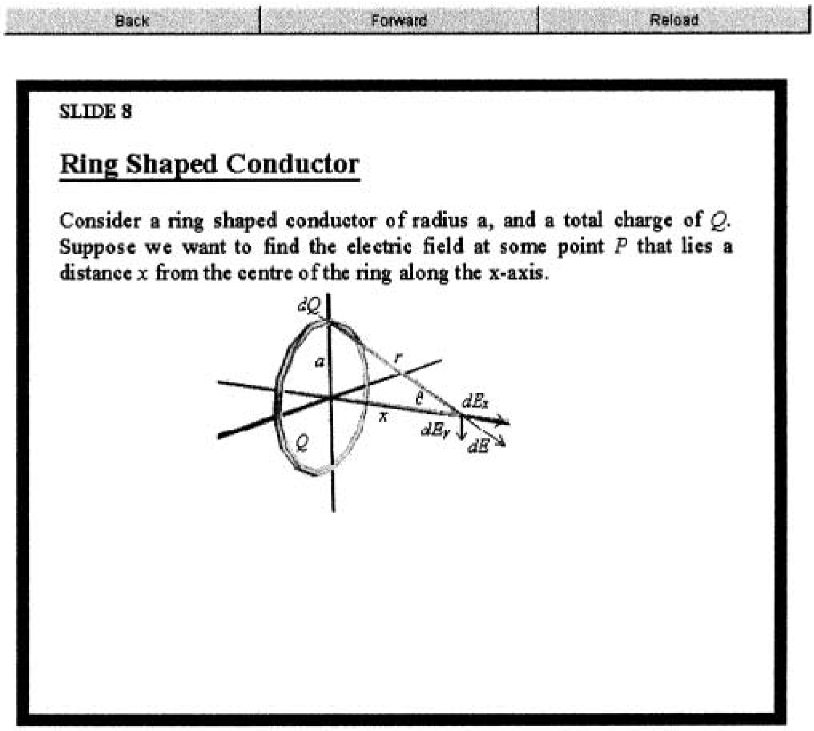

Fig. 2 *A slide from the slideshow.*

<APPLET CODE = "ESlide.class" HEIGHT = 650 WIDTH = 650></APPLET>

Fig. 3 *The HTML tag that loads up the applet ESlide.*

attribute of the tag indicates the Java class file to load, whilst the height and width attributes specify the size of the applet on the browser screen in pixels.

The tutorial sections on other topics are similarly implemented. Next we consider the interactive simulation studies designed for two topics, namely electric dipole and field mapping. The approach is generic and hence can be easily extended to other topics without difficulty.

Interactive simulation studies for electric dipole

To have an effective educational value, an interactive simulation module in EM should contain the following features:

- present the problem in some diagrammatic form;
- prompt for initial conditions value;

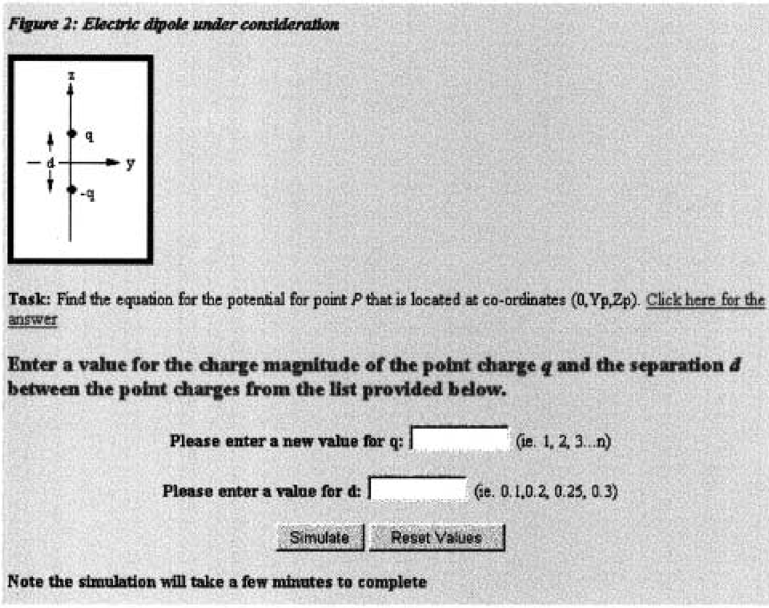

Fig. 4 *Form requesting the user to input values for the simulation.*

- perform the computational process;
- display the results in suitable format on the browser.

By clicking a particular topic, say 'Electric dipole' in the left frame of the index page of the module (see Fig. 1), the user is provided with three options: Tutorial, Simulation and Animation in the following page (not shown). Clicking on 'Simulation' brings the user to a screenshot as shown in Fig. 4. The user is asked to enter two values, the magnitude of the dipole charge q and the distance between the charges, d. The suggested values on the side of the text boxes show the type of values that should be entered. In particular, d should be kept small for the simulated results to make sense, and to keep the problem realistic. Clicking the *Reset Values* button simply resets any value entered into the form's text boxes while clicking 'Simulate' starts the simulation.

The program

Clicking on the *Simulate* button calls up a CGI script that accepts the values of d and q. The CGI script, ***simulate.pl*** written in PERL, implements the client-server interaction. The script creates a MATLAB script file called ***dipole.m***. The CGI script also invokes the UNIX command *system* and *nohup* to run the MATLAB computational engine in the background and to ignore any quit calls. After MATLAB completes its computations, it produces three JPEG files. The CGI script then proceeds to generate a new HTML page that contains the JPEG files and also the input values

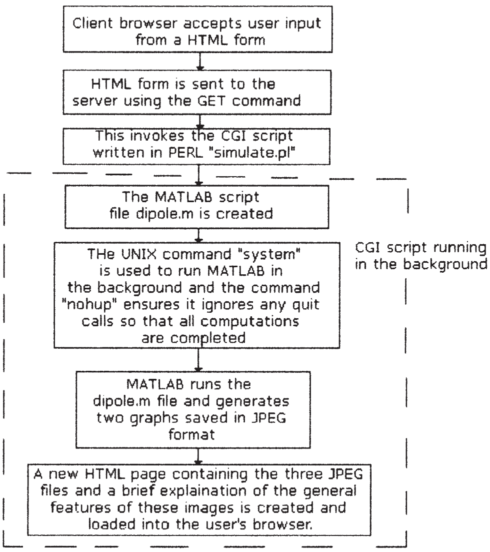

Fig. 5 *Flowchart depicting the communication between the client browser and the server.*

of charge q and distance d. This page is then displayed in the browser, allowing evaluation of the results.

Fig. 5 below shows the flow chart that provides a diagrammatic depiction of the interaction between the client browser, the CGI script, and the MATLAB computational engine, to complement the explanation mentioned above.

Simulation studies

Fig. 6 gives the display on the browser of a 3-dimensional plot of the electric potential obtained as a result of interactive simulation carried out. The students can make observations and draw conclusions from this figure such as the occurrence of peaks of the potential plots at the locations of the positive and negative point charges or conclude that the largest positive potential is where the positive point charge is located.

Fig. 7 shows another result of the electric field simulation on the web. Here, the students can make observations of the electric field e.g. the lines starting from the positive charge and terminating at the negative charge. Given that the relationship between the electric field and potential is $E = -\nabla V$, the magnitude and direction of the electric field is proportional to the slope of the scalar potential plot. It can further be observed that as the negative slope of V is directed away from the positive point charge, the electric field lines will be directed away from this charge. Similarly, the electric field lines will be directed towards the negative point charge.

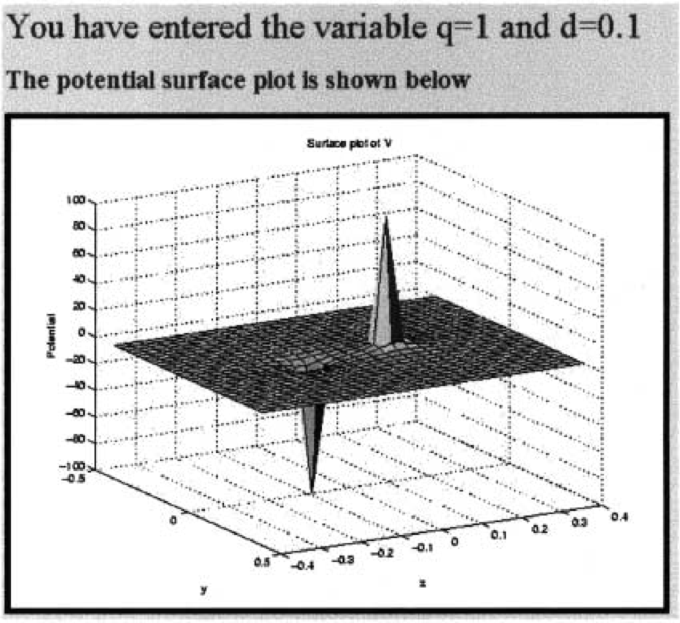

Fig. 6 *Surface potential plot at the conclusion of the simulation.*

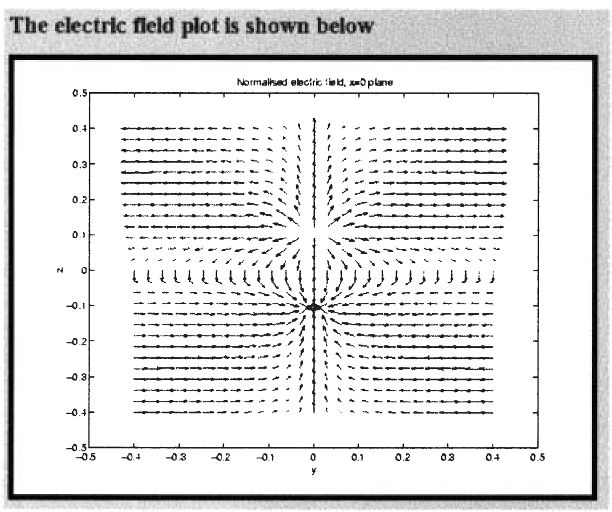

Fig. 7 *Electric field plot at the conclusion of the simulation.*

Animation effects

The ‘Animation’ section seeks to illustrate the results in a manner different to the ‘simulation’ method described above. It provides an understanding of these changes through the use of short animations. Cycling a sequence of images (obtained by varying values of a parameter) in quick succession produces an animation effect. The individual stored images are generated plots by MATLAB while the actual animation sequence is implemented on the web via a Java applet.

Figure 8 shows shows a screen capture for a dipole interactive animation. It presents a list of six choices and allows the user to choose an option by clicking any one of the ‘radio’ buttons shown. By putting the radio buttons in a *checkbox group* it is ensured that only one option can be selected at any given time. Clicking the ‘Start Animation’ button starts the animation. A sequence of images will then be cycled through at a high frame rate. These images are contained in an array, so cycling through them only involves incrementing the array index. Upon reaching the end of the array, the array index is set back to point to the beginning of the array. The images are then cycled through again as before. This makes the animation repeat in a continuous loop. The user is able to stop the animation at any point by clicking the ‘Stop Animation’ button.

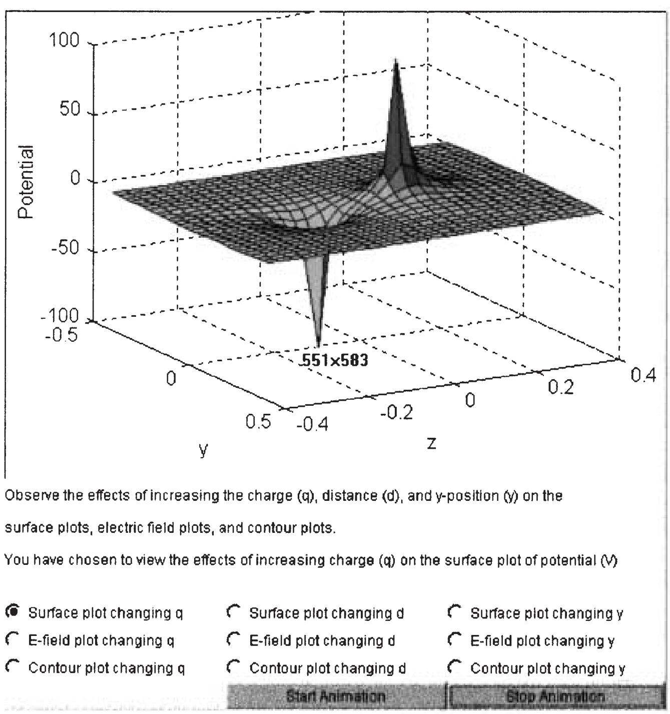

Fig. 8 *Dipole Animation applet.*

Simulation studies with experimental field mapping

Simulation studies for experimental field mapping can be carried out by clicking on 'Experimental mapping methods' shown in Fig. 1 and then clicking on 'Simulation' in the screen following (not shown). The numerical algorithm underlying the experimental mapping method that has been used is the 'Iterative method'.[12] The given region is divided into squares and an initial estimate of the potential at each corner of each of the squares is made. Applying the iterative process improves the estimates at each corner until a solution is reached. The better the initial estimate the shorter the number of iterative steps required. It should be noted that the final answer is not affected in any way by these initial estimates. The iterative method is implemented using Java.

To perform simulation studies, three options for the configurations of the duct regions are available, namely the Square, Triangular and Oblong. When a user clicks, say *Square Shape* in the index page (Fig. 1), it loads a page ***Square.shtml*** (which in turn loads up the applet ***Square.class***) as shown in Fig. 9 below. The user next enters the potential value of boundary **A** (say as 100 V), and the initial estimates (say 10 V) at each of the points (points **a** to **i**), as depicted in the screenshot. The page

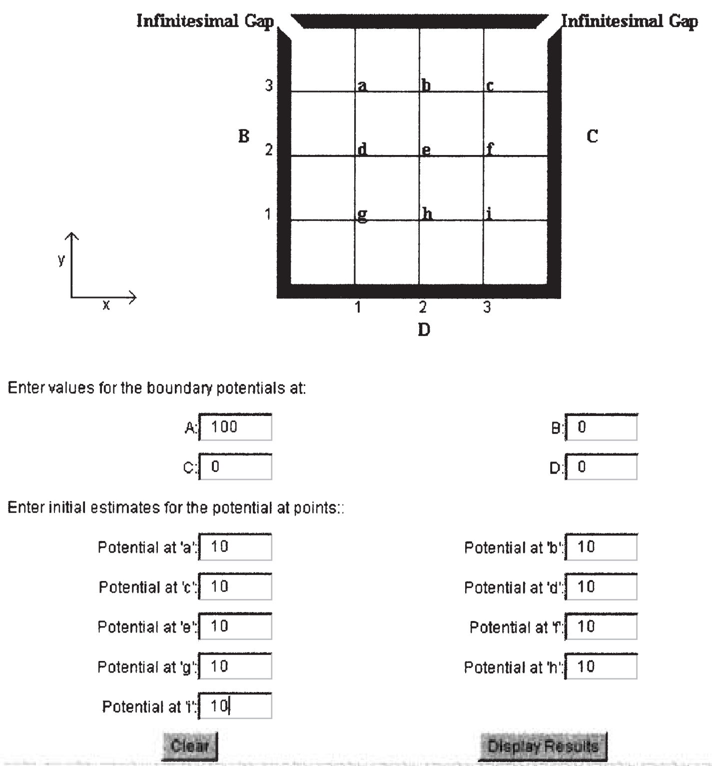

Fig. 9 *Screenshot for simulation of the Square duct.*

also contains 'Display Results' and 'Clear' buttons. The 'Clear' button resets all the values entered by the user to the default values. On clicking the 'Display Results', the iterative computational process begins and stops when all the potential value variations fall below a specified degree of tolerance (<0.00001). At this point a new window pops up (not shown) in the browser displaying the number of iterative steps and the potentials at each point **a** to **i**.

Animation effects

The animation effect for experimental field mapping serves as a visual aid to assist in understanding the variation of potentials. The animation is designed to animate the changes in potential inside the duct due to variation of boundary conditions at A. Only the 'square duct' problem has been taken up for the animation described in this section since the various grid points can be conveniently mapped into the *x*- and *y*-axis. When the user clicks '*Square Results*' on the contents frame (Fig. 1) and then clicks 'Animation' in the following screen (not shown), it brings them to a screen similar to that shown in Fig. 10. To implement the animation effect, pre-determined results for potential values obtained from simulation (for varying values of the boundary condition A) are used to generate a three-dimensional plot of the potential at each point within the duct.

As can be seen from Fig. 10 above, the user is presented with a 'Start Animation' button, a 'Stop Animation' button and a scrollbar. Like in all the previous applets,

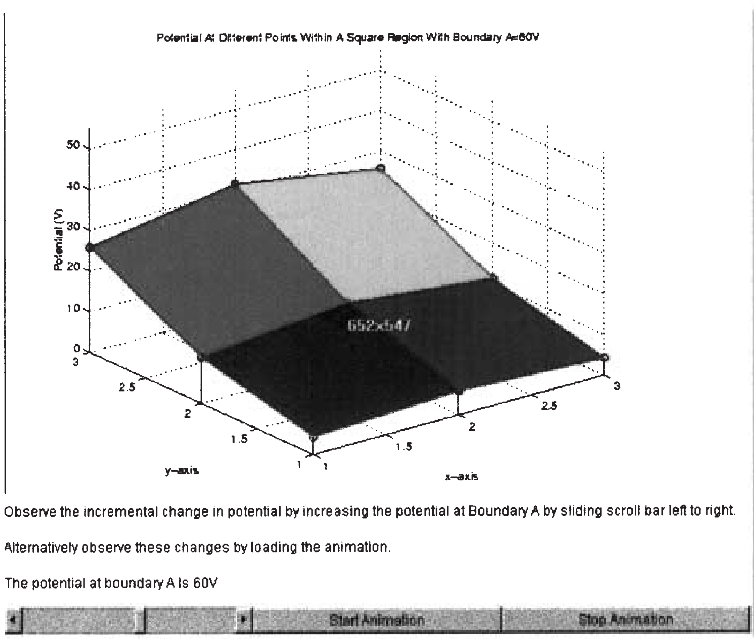

Fig. 10 *The Square Results applet with* A = *60V.*

a small audio file is also attached to each button and the scrollbar. When the scrollbar is moved from its leftmost to the rightmost position, the potential of boundary A increases from 10 V and 100 V respectively. Moreover, instead of viewing a static image, the user can click 'Start Animation' button to start and 'Stop Animation' button to stop the animation at any point. Although the image files are small to download, problems such as image distortion may be noticeable in the initial animation sequence if network connections are slow. However, once the images have been downloaded into the browser's cache, the animation runs without any distortions, i.e. the applet runs smoothly on the second cycle of the animation sequence. This scheme possesses an obvious limitation, namely that the user has no real control over the parameter variation since 'canned displays' are being displayed.

Conclusions

Electromagnetics is perceived as one of the most difficult courses in engineering due to its extensive mathematical content. The use of interactive multimedia tools incorporating animation effects can allow students to have a graphical display of abstract quantities such as fields, potentials and help visualization, thus making the subject easier to understand. This paper presents a web-based interactive module for electromagnetics. It is designed and implemented using Java applets, Perl scripts and MATLAB. Clicking buttons, such as 'Display result', and multimedia features such as the audio files are also included in various applets for stimulating the user's sound senses. The module contains not only the tutorial and the simulation components but also incorporates animation effects. Design and implementation schemes for interactive simulation studies for two topics, namely 'electric dipole' and 'experimental mapping' are also presented. All the graphical plots and the animation results are displayed directly on the browser.

References

1 B. Oakley, II, 'A virtual classroom approach to teaching circuit analysis', *IEEE Trans. Educ.*, **39** (3) (1996), 287–296.

2 P. J. Mosterman, J. I. Campbell, A. J. Brodersen and J. R. Bourne, 'Design and implementation of electronics laboratory simulator', *IEEE Trans. Educ.*, **39** (3) (1996), 309–313.

3 R. O. Harger, 'Teaching in a computer classroom with a hyperlinked, interactive book, *IEEE Trans. Educ.*, **39** (3) (1996), 327–335.

4 B. Aktan, A. Bohus, L. A. Crowl and M. H. Shor, 'Distance learning applied to control engineering laboratories', *IEEE Trans. Educ.*, **39** (3) (1996), 320–326.

5 S. I. Mehta and S. M. Gronhovd, 'Instrumentation and communication modules on CD-ROMs for enriching engineering education', *IEEE Trans. Educ.*, **39** (3) (1996), 304–313.

6 M. Chetty and K. P. Dabke, 'New approaches to system and control teaching', *Proc. Pacific Region Conference on Electrical Engineering Education, Melbourne, June 1999*, pp. 46–51.

7 P. Thomas, 'Teaching over the Web: the future,' *Computing and Control Engng J.*, **8** (3) (1997), 136–142.

8 B. Beker, D. W. Bailey and G. J. Cokkinides, 'An application-enhanced approach to introductory electromagnetics', *IEEE Trans. Educ.*, **41** (1) (1998).

9 http://www.treasuretroves.com/physics/topics/Electromagnetism.html
10 F. Buret, D. Muller and L. Nicolas, 'Computer-aided education for magnetostatics', *IEEE Trans. Educ*., **42** (1) (1999), 45–49.
11 K. Whites, 'Visual Electromagnetics for Mathcad', http://www.engr.uky.edu/~visualem/, 5 November, 1999.
12 W. Hayt, *Engineering Electromagnetics*, 5th edn (McGraw-Hill, New York, 1989).