Web Based Laboratories for Teaching Electromagnetics for TEMPUS eLab Project

Youssra ElQattan and Hani A. Ghali
Electrical Engineering Department
Faculty of Engineering, British University in Egypt (BUE)
youssra.elqattan@bue.edu.eg & hani.amin@bue.edu.eg

Abstract

This paper presents a successful step towards the development of a "web-based laboratory" for teaching basic, and even advanced, electromagnetic concepts. The main idea is to develop a reusable model for the student to be used exactly as a hardware experiment in a physical laboratory, where he/she can change some of the experiment's physical parameters and get corresponding results. The two main problems which had to be solved are;

I. The model should be accessible remotely through the internet using any browser, taking into account the fact that the student has no COMSOL installed on his/her computer

II. The model should not require a student's experience in using COMSOL, which means that only physical parameters that represent input variables and output data should be visible to student.

Keywords

Web-based electromagnetism labs, client-server model, Web services, JavaScript, HTML, COMSOL MATLAB Live-Link

1. Introduction

Experimentation and laboratory work play an important role in the learning process of any engineering course. They help students to grasp the concepts and theories they learn in classes through practicing on real or simulated applications. However, the implementation of laboratories that accommodate large numbers of students & are supported with satisfactory technologies represent a challenging financial objective. Moreover, students who have difficulties meeting the time frame where laboratories sessions are offered represent another challenging objective. Consequently, the need for the development of web-based laboratories that can be accessed anytime from anywhere & can be repeated any number of times has become essential for the current, as well as for the future, learning process. Furthermore, to get the maximum benefits of the learning activities, it is recommended that all software simulation details are to be masked from the student's interface.

This study describes an approach to be used in teaching electromagnetic concepts through a web-based simulation laboratory which has been built using COMSOL Multiphysics and can be remotely accessed using any web browser.

2. Platform

The platform used in the web-based lab is a website that allows students to access the lab, provides them with the required experiments' information, contains the interface of the experiment and helps in assessing students' performance. The student logs into this website, selects the experiment he/she would like to perform where he/she will find the experiment's theoretical background together with some pre-lab questions. The student's will then be directed to the experiment's interface with guidelines on steps and tasks to follow.

It worth noting that using a standard learning management system (LMS) facilitates building such platforms with all necessary tools and activities. Consequently, the Moodle [1], which is an open source LMS was planned to be used in building this web-based lab platform.

3. Suggested approaches for the implementation

The following are some approaches that were investigated.

3. I. Building a java applet that's embedded in a web page & runs on the client's computer. This idea takes advantage of COMSOL's Java API [2]. The advantage of this approach is that all the work is done on the client side which reduces the load and traffic on the server. An applet was built as shown in Fig. 1. However, it was figured out that this approach can't be adopted since it needs COMSOL to be installed on the client's machine which violates one of the intended objectives of building this web-based lab, and consequently this approach was discarded.

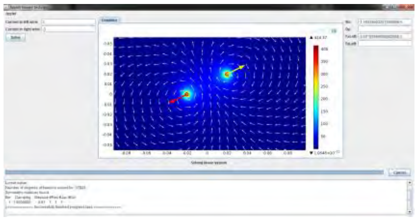


Figure 1: The implemented java applet running in the applet viewer

3. II. The second approach, shown in Fig. 2, was to design an HTML web form with some text boxes & a submit button which, when clicked, sends a request asynchronously using the XMLHttpRequest object [3] to a PHP script that gets executed on the server side. This script runs a Java program on the server. This java program initializes the COMSOL model with parameters that are sent along with the request, runs the solution and extracts the required numerical results from the model & saves them in a text file on the server. It also exports the required images to the server. Afterwards another XMLHttpRequest is sent to the server to get those saved numerical results and images & display them on the HTML page using JavaScript. This approach worked fine but the problem was that every time a request is made to change a parameter in the model, the model had to be initialized from scratch which would take a very long time & would harm the user's experience.

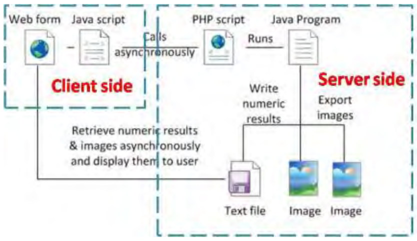


Figure 2: Second approach web flow

3. III. Due to the problem faced in the previous approach, there was a motivation to find another solution. The COMSOL MATLAB live link [4] was investigated. The COMSOL MATLAB live link is based on the client-server model, where MATLAB acts as the client & COMSOL server acts as the server. Through a MATLAB script a model can be initialized on the COMSOL server, acted upon, solved and results can be extracted from it same as with the JAVA API. The advantage is that this model can be held in the server's memory and then the parameters and variables of that model can be tuned & the solution can be rerun without having to reinitialize the model every time, which minimizes the time consumed to a great extent. In this case, the approach was to use the HTML page designed in the previous approach but this time when the submit button is clicked, an XMLHttpRequest is sent to a PHP script on the server that runs MATLB in batch mode and let it execute a MATLAB function that takes the new values of the parameters or variables to be tuned as arguments. This function connects to COMSOL server and gets a reference to the required model. Then it tunes the needed parameters, reruns the solution, extracts the required numerical results, exports the required images and saves them on the server. Finally, as in the previous approach, another XMLHttpRequest is sent to the server to get those saved results and images and display them on the HTML page. The flow of this approach is shown in Fig. 3.

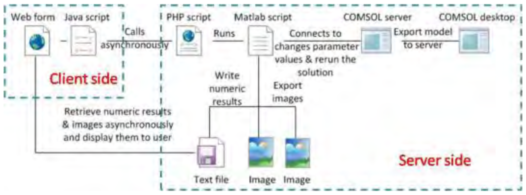


Figure 3: Third approach web flow

3. IV. For the sake of further improvements & to have more control over the flow & make it less error prone, the part of running MATLAB batch mode through a PHP script was replaced by a Java servlet that makes use of the MatlabControl class [5] which is a Java API that allows for calling MATLAB from Java. Consequently, the Java applet calls the Java servlet which controls a MATLAB session that runs the MATLAB function mentioned in the previous approach. The web flow of this approach is shown in Fig. 4.

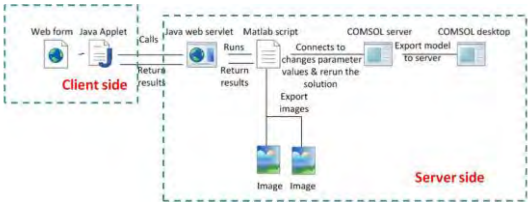


Figure 4: Fourth approach web flow

4. Implementation

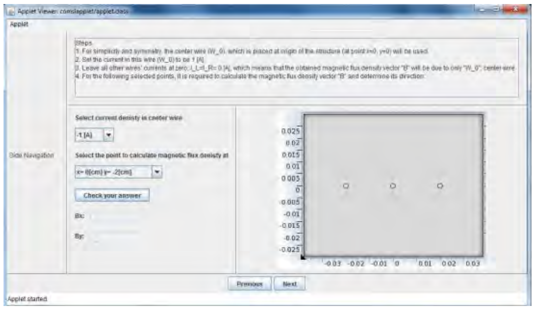
The described fourth approach has been selected and implemented on a simple test experiment that deals with the calculation of the magnetic flux density vector due to a single or multiple parallel current carrying conductors. For this purpose, a model has been created which includes three parallel current carrying conductors. The experiment has mainly three objectives:

I. Familiarization with the use of simple expression based on Biot-Sarat law for the determination of the magnetic flux density vector "B" due to a single infinite long thin current carrying conductor.

II. Calculation of the magnetic flux density vector "B" due to different current configurations of parallel current carrying conductors.

III. Appreciation of how different current configurations, magnitudes and directions, in parallel current carrying conductors could be used to produce a predefined magnetic field configuration.

To implement this experiment, the developed model was exported to COMSOL server. A Java Applet was then designed including a representation of the geometry of the model and two dropdown menus: 1) to set the current in the center wire, and 2) to select the point at which the magnetic flux density value is required, together with a calculate button as shown in Fig. 5.


Steps
1. For simplicity and symmetry, the center wire (W_0), which is placed at origin of the structure (at point x=0, y=0) will be used
2. Set the current in this wire (W_0) to be 1 [A]
3. Leave all other wires' currents at zero: I_L=I_R= 0 [A], which means that the obtained magnetic flux density vector "B" will be due to only "W_0", center wire
4. For the following selected points, it is required to calculate the magnetic flux density vector "B" and determine its direction

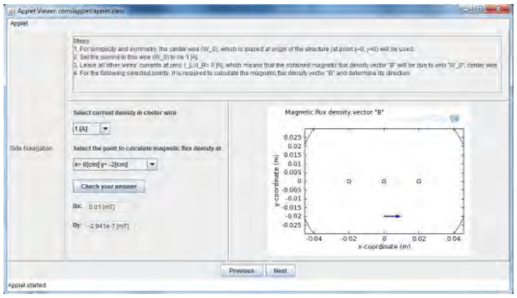
Select current denisty in center wire: -1 [A]

Select the point to calculate magnetic flux denisty at: x= 0[cm] y= -2[cm]

Applet started

Figure 5: Developed Java applet

When the "calculate" button is clicked a Java servlet is called. This servlet uses the MatlabControl class to connect to a MATLAB session & execute a MATLAB script. This MATLAB script connects to COMSOL server & gets a reference to the previously exported model. It then sets the position of the point at which the magnetic flux density is to be calculated to the coordinates the user selected. Then the solution is rerun, a plot of the magnitude & direction of the magnetic flux density at that point is exported. In addition, the numerical results of the magnetic flux density are calculated and both are returned to the servlet. Finally, the servlet returns those results to the Java Applet, which then displays the results & the plot as shown in Fig. 6.



Steps
1. For simplicity and symmetry, the center wire (W_0), which is placed at origin of the structure (at point x=0, y=0) will be used.
2. Set the current in this wire (W_0) to be 1 [A].
3. Leave all other wires' currents at zero, I_L=I_R= 0 [A], which means that the obtained magnetic flux density vector "B" will be due to only "W_0", center wire.
4. For the following selected points, it is required to calculate the magnetic flux density vector "B" and determine its direction.

Select current denisty in cneter wire: 1 [A]

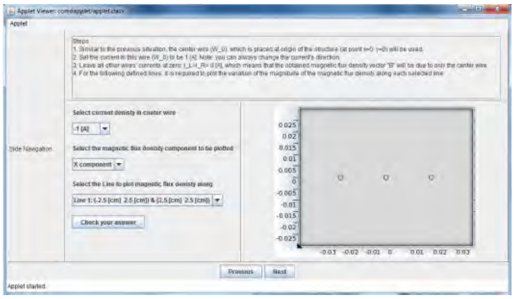
Select the point to calculate magnetic flux denisty at: x= 0[cm] y= -2[cm]

Bx: 0.01 [mT]

By: -2.941e-7 [mT]

Figure 6: Displayed plot and numerical results for the magnetic flux density vector at a selected point

Afterword's the student clicks "Next" to perform another task of the experiment as shown in Fig. 7.



Steps
1. Similar to the previous situation, the center wire (W_0) which is placed at origin of the structure (at point x=0, y=0) will be used.
2. Set the current in this wire (W_0) to be 1 [A]. Note: you can always change the current's direction.
3. Leave all other wires' currents at zero: I_L=I_R= 0 [A], which means that the obtained magnetic flux density vector "B" will be due to only the center wire.
4. For the following defined lines, it is required to plot the variation of the magnitude of the magnetic flux density along each selected line.

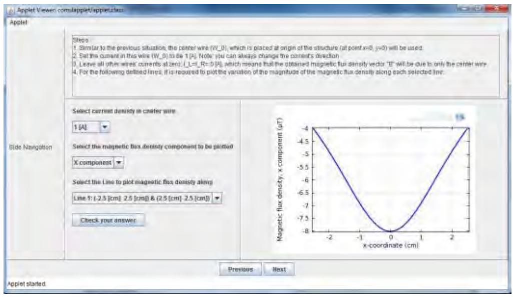
Select current denisty in cneter wire: -1 [A]

Select the magnetic flux denisty component to be plotted: X component

Select the Line to plot magnetic flux denisty along: Line 1: (-2.5 [cm], 2.5 [cm]) & (2.5 [cm], 2.5 [cm])

Figure 7: Developed Java applet for the second task interface

In this task the student is requested to plot either the "x-component" or the "y-component" of the magnetic flux density along the path he/she selects. Then clicks "check your answer", and then the obtained plot will appear as shown in Fig. 8.



Steps
1. Similar to the previous situation, the center wire (W_0), which is placed at origin of the structure (at point x=0, y=0) will be used.
2. Set the current in this wire (W_0) to be 1 [A]. Note: you can always change the current's direction.
3. Leave all other wires' currents at zero, I_L=I_R= 0 [A], which means that the obtained magnetic flux density vector "B" will be due to only the center wire.
4. For the following defined lines, it is required to plot the variation of the magnitude of the magnetic flux density along each selected line.

Select current denisty in cneter wire: 1 [A]

Select the magnetic flux denisty component to be plotted: X component

Select the Line to plot magnetic flux denisty along: Line 1: (-2.5 [cm], 2.5 [cm]) & (2.5 [cm], 2.5 [cm])

Figure 8: Screenshot of the second task interface with the required plot displayed

5. Conclusions

A Java applet has been developed based on the use of COMSOL MATLAB Live Link. This Java applet enables the implementation of "*web-based simulation experiments*" that could be used in teaching related topics. As a test case, the developed Java applet has been implemented to build a basic core simulation experiment in electromagnetics related to the determination of the magnetic flux density due to parallel current carrying conductors.

Each task of the experiment has been packed in a corresponding Java applet, with both input variables and output data. The simulation details are fully masked from the student's interface thus enables more focus on the experiment objectives and learning outcomes. The Java applet runs on student's computers without COMSOL being installed. However, the student's computer should be connected to the internet where on the server side; both COMSOL & MATALB are installed.

The presence of these "*web-based labs*" makes the experiment available 24/7 with an unlimited number of experimentation possibilities. In addition, these experiments could be performed from anywhere using an internet browser.

6. Acknowledgement

This research is part of the project entitled, "*e-Laboratories for Physics and Engineering Education*" funded through TEMPUS project# 517102-TEMPUS-1-2011-1-SETEMPUS-JPCR.

7. References

[1] Moodle http://moodle.org/?lang=ar
[2] COMSOL JAVA API guide.
[3] http://www.w3.org/TR/XMLHttpReques/
[4] COMSOL MATLAB livelink guide
[5] http://code.google.com/p/matlabcontrol/