Online Library of Scientific Models - A New Way to Teach, Learn, and Share Learning Experience

Hatem Elrefaei, Tamer Aboelfadel, Maha Elmeseery, Ali Elmofty, Marwah Shafee

HAL Id: hal-00257139

https://telearn.hal.science/hal-00257139

Submitted on 18 Feb 2008

Online Library of Scientific Models
A New Way to Teach, Learn, and Share Learning Experience

Dr. Hatem Elrefaei[1] IEEE member, Dr. Tamer Abo Elfadel[2], Eng. Maha Elmeseery, Ali Elmofty, Eng. Marwah Shafee

[1] Ain Shams U., Faculty of Engineering, Mathematics and Engineering Physics Department, and CEO of EngNet Inc.

[2] Cairo U., Faculty of Engineering, Electronics and Communication Engineering Department.

Key words: *Scientific models, e-learning, engineering, equations, java applets*.

Abstract:

While scientific models are usually communicated in paper format, the need to reprogram every model by every user results in a huge loss of efforts, time and money, hence lengthening the educational and research developing cycle and loosing the learning experience and expertise gained by every user. We demonstrate a new portal www.imodelit.com that hosts a library of scientific models for electrical engineers in the form of java applets. They are all conformal, informative, with strong input and output filing system. The software design allows a fast developing cycle and it represents a strong infrastructure that can be shared by researchers to develop their own applets to be posted on the library. We aim for a community based library of scientific models that enhances the e-learning process for engineering students.

1   Introduction

Historically, teaching scientific models (a scientific model is a mathematical representation of a physical phenomena or device) went hand by hand with the technological development of the teaching tools. For centuries, the teaching process was depending on board teaching where the lecturer derives all mathematical formulations. To show the model performance, he had to draw point by point the mathematical function on the board. Though, for many times, the models were so complicated that he was only able to describe intuitively the whole performance without being able to go in details on how the model changes as parameters change.

With the introduction of computer and LCD projectors, the teacher facility was extended to be able to actively exercise scientific models in front of student that he previously programmed using any scientific programming language. This enhanced the learning process and provided to students a better way to understand and capture in depth a lot of the behind physics and mathematics.

With the introduction of the internet in middle nineties of the last century, it opened another dimension for the learning process where learners could be anywhere and want to have access to material anytime. In addition, the Internet opened a huge opportunity to share information, experience, and files. For science and engineering students, sharing knowledge of scientific models and designs of devices are very important to shorten a lengthy learning process of heavy mathematical problems of the phenomena under study. In that context,

learners could be undergraduate, graduate students, or researchers in institutions or companies.

It took few years since the introduction of the Internet to develop special tools that allowed Internet users to produce interactive online programs. Among these, java from Sun Microsystems [1] was a wonderful open source tool. Since the late nineties of the last century, we started to see professors posting java applets on their university home pages that describe physical phenomena or device operation. Worldwide, many professors have adopted this trend over time. Though these applets are useful for students to capture some physical concepts easily, there are few deficiencies that should be addressed.

1. Most of the applets relay on the visual representation of the phenomena – e.g. moving electrons in magnetic field, electric field distribution of charges –. In most of the applet, no actual data can be extracted from the applet that can be useful for a student or a researcher to make a design or quantify the output.

2. The java applets have different interfaces, even for applets within the same professor's website. Thus users have to spend some time for each one to learn how to use it which lengthens the learning process.

3. The learning experience by one user can not be transferred to another. Similarly, the user can not save the applet condition to continue work on it in another day. There is a lack in filing system usage for the applets.

The aim of the work presented in this paper is to demonstrate how we have addresses these deficiencies in the proposed portal of scientific models at www.imodelit.com.

2   Online library of scientific models

Here, we present a library of Java applets for scientific models in the electrical engineering domain that can be used for e-learning both for the undergraduate and graduate level. The infrastructure that was built can be reused to serve other branches in engineering and science. Currently the library is hosted at www.imodelit.com where it currently contains more than 220 different models. The aim is to build huge library of Java applets where each is devoted to simulate, analyze, synthesize one single component, or physical phenomena. This library's hierarchy goes from the top level which is "societies", into middle level "categories", down to the small ingredients which are the applets. Users – university staff, students, researchers, or designers in relevant companies – can electronically subscribe to a society, a category, an applet or a combination of them on **monthly/quarter/annual** bases.

For educational and research activities in universities, the financial burden to buy licensed software is another obstacle for development. Due to global accepted norm of Intellectual Property Rights (IPR), and with more countries issuing laws criminalizing software infringing, the use of open source technology became an attractive solution for IT development. In developing www.imodelit.com library of scientific models we chose to use open source technologies for both internal operation and product development [2], [3], [4], and [5].

Scilab [6] was adopted for internal development where deep understanding of scientific models is a must. Scilab is a very efficient software development environment with a huge library of scientific functions required by researchers and engineers.

3   Java applets description

3.1   Software design overview

While the principle of java applets for teaching already exists, what is offered in this project is a community based library that can easily capture a lot of scientific models with the minimum amount of efforts.

Thus, a generic software design that consists of two parts is employed in all applets. The first part; framework module; is a fixed module for all applets and contains all generic features of the applets such as "zoom", "markers", "axis setting", "generate report", … extra. The second part; the component model module; varies from an applet to another and contains the definitions of all parameters and the governing equations or algorithm of the device or problem under study. These two modules are separate such that any component model module can fit inside the standard framework one. Figure (1) illustrates the relation between these two modules. Such a design has two advantages:

1. It makes the developing cycle of an applet as short as possible, since what it needs to develop a new applet is just putting the model equations in the right place inside the generic design and the rest is already done.
2. It facilitate the usability of the applets, since once one learns how to use one applet, the rest of the applets will be the same.
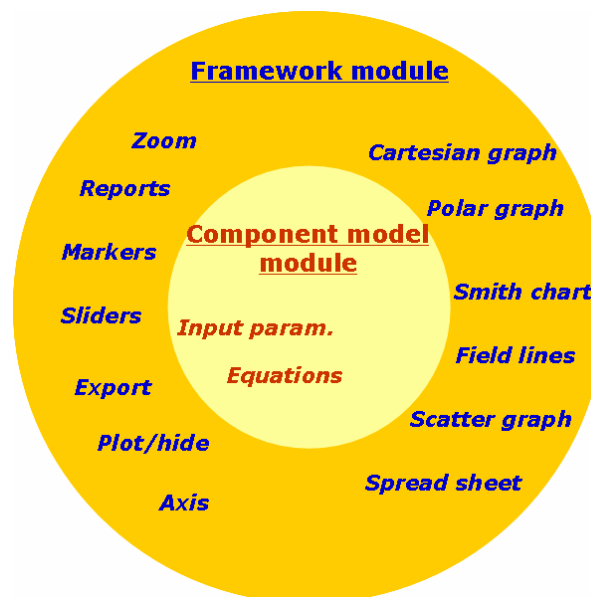

Fig. (1) The two modules of the applet design.

3.2   GUI overview

All applets in the library have the same interface. The applet consists of three main sections; the input section, the control section, and the output section as shown in Figure (2), where an applet for afocal compensator is chosen as an example. Typically, after the applet is opened it shows example parameters in the input section, and a graph of the device or phenomena under study in the output section as shown in Figure (2).
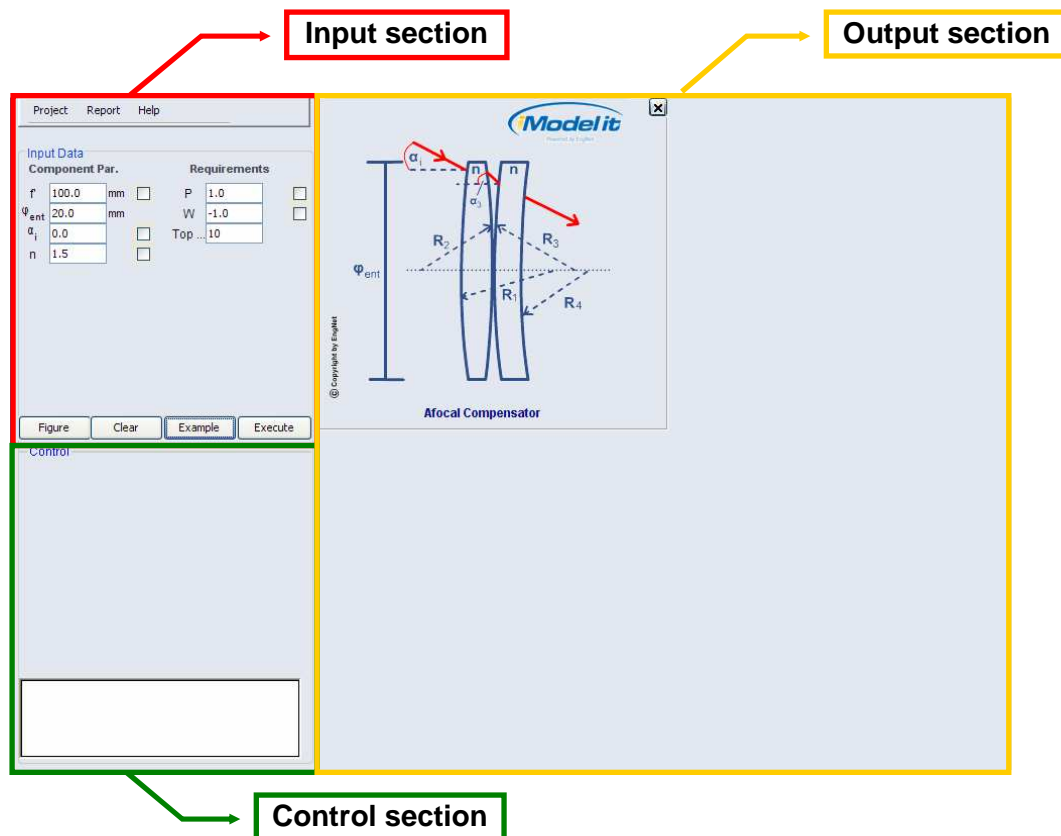
Fig. (2) The applet interface just after opening. It shows example parameters set in the input section and a model graph opened in the top left quadrant of the graph area.

3.2.1 The input section

This section contains all the input parameters of the scientific models which are divided into related groups. The parameters are entered in text fields or selected from the drop down menus.

It also contains the primary buttons which are displayed when the applet opens and they control the functionality of the applet. These buttons are:
1. **Figure button**: It displays the figure representing the device or phenomena understudy.
2. **Clear button**: It clears all outputs and inputs of the applet.
3. **Example button**: It provides the user with a default example values for all parameters in the input section.
4. **Execute button**: It reads and validates all input parameters before calculating the outputs.

3.2.2 The control section

It contains the control buttons which are located below the primary buttons and only appear after clicking on the "Execute" button as shown in Figure (3). They control the mathematical functions that will be plotted in the output section.

It also contains parameter tuning sliders as shown in Figure (3) below the control buttons. Users can have five sliders at maximum at any one time and they appear by checking the required check box beside the required parameter in the input section.
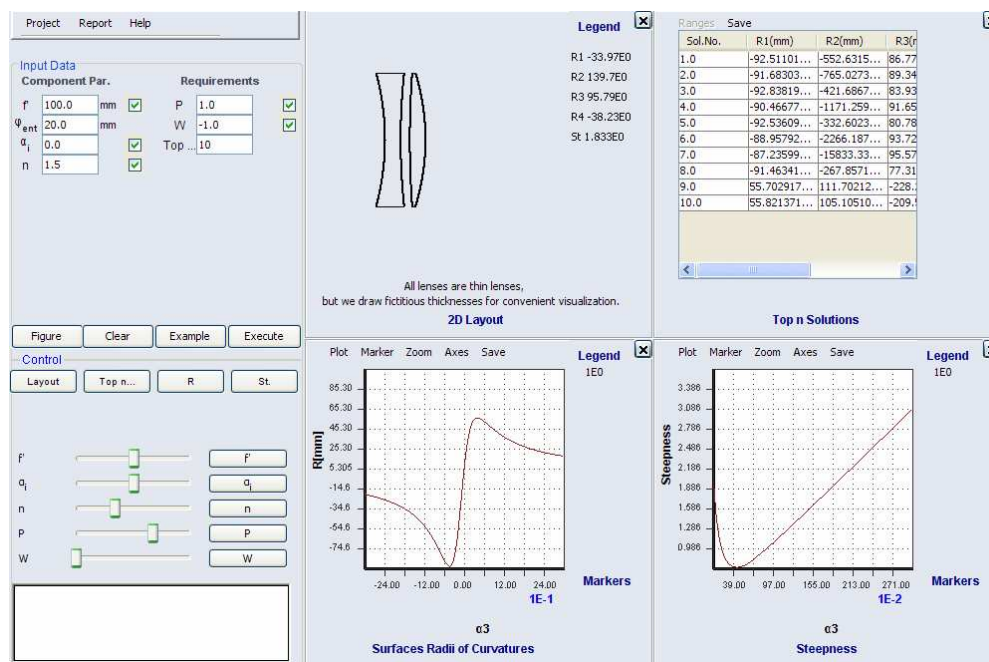
Fig. (3): Snapshot of afocal compensator applet after pressing the Excute button and open 4 different graphs.

3.2.3 The output section

It is where every graph or function will be displayed. This section is divided into four graphs locations (you can only view four graphs at a time). A graph opens when a user clicks on any of the control buttons. If 4 graphs are open occupying all 4 allowed places in the output section as shown in Figure (3), any new graph will override the top lift graph. There are 9 different types of graphs that can be used in the applets as shown in Figure (4).
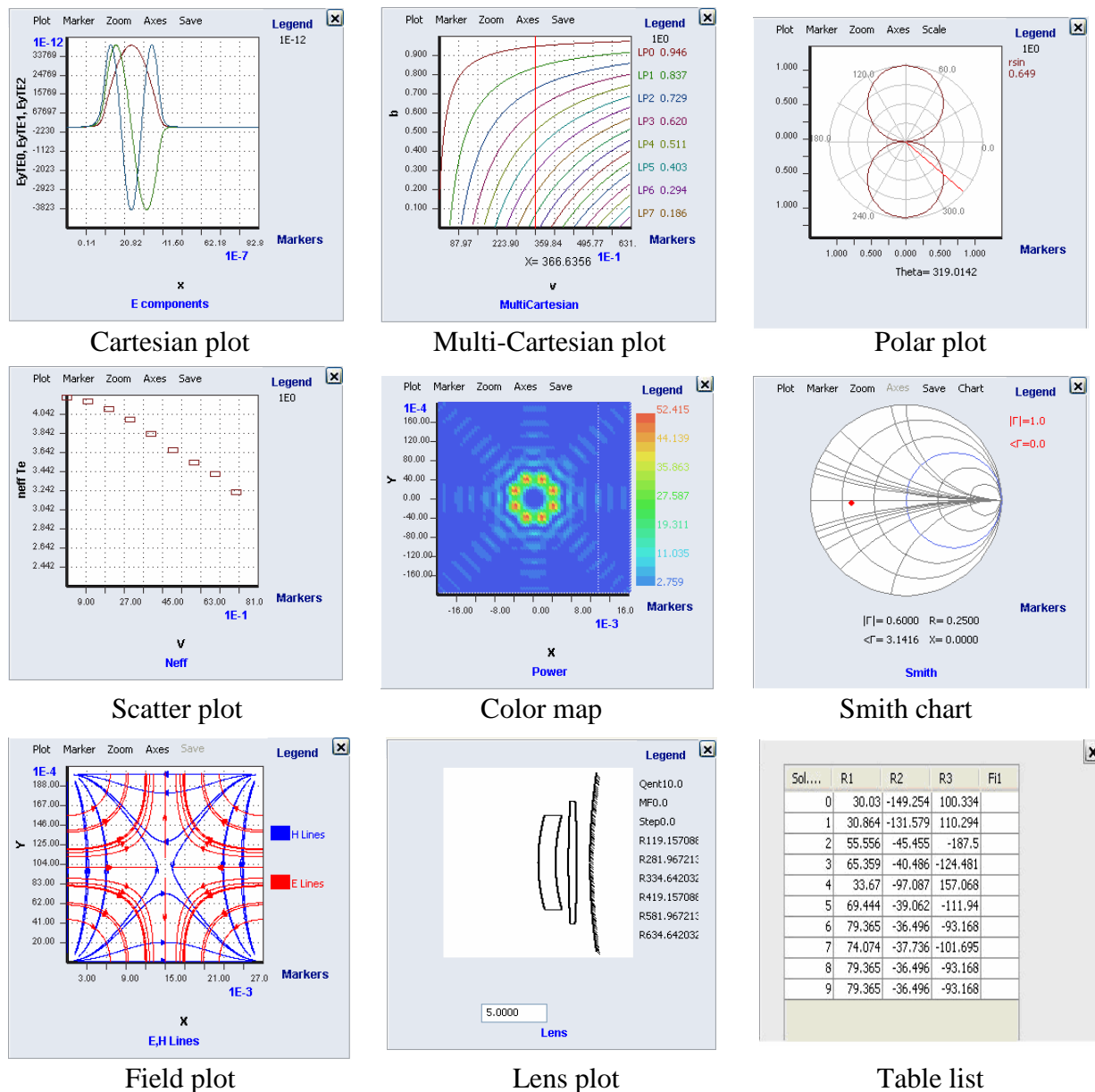
Cartesian plot	Multi-Cartesian plot	Polar plot
Scatter plot	Color map	Smith chart
Field plot	Lens plot	Table list

Fig. (4): 9 different types of graphs available to be used in any java applet.

3.3 Java applets features

The intent here is not to give full detailed list of the software features but rather to highlight features that we think do not exit in similar scientific applets. Full detailed features of the applets can be reviewed in the software technical help [7].

The main menu bar in the input section allows the user to perform certain operation on the whole applet.

1. All the model parameters can be exported or imported as XML file. This allows users to save their designs and restart again from their last point. In addition, it allows users to exchange their designs through exchanging the XML files.
2. A PDF report can be generated containing all input parameters and output graphs plotted. This report could be useful fro students as a way of communication with the instructor.
3. A TXT report can be also generated containing all input parameters and output performance as raw data that can be opened by any other scientific platform to further carry on post processing on them.

The input section has very useful features that apply on the input data:
1. Self checking on the physical constrains of the values of the parameters. This ensures that the model will work properly.
2. Different types of parameters are allowed;
    a. Simple text field that accommodates integer or double numbers.
    b. Double text fields that accommodate complex numbers.
    c. Drop down menus to select different conditions of the model.
    d. Editable table for tabulated type of input parameter that is normally loaded with example value or can be loaded from the client computer.

The menu bar of any graph allows the user to perform many operations on the corresponding graph:
1. Plot/hide/clear any curve from the graph area.
2. Zoom in and out.
3. Put/clear markers on the graph that specifies the coordinates of the marker point.
4. Change the axis setting.
5. Save only the data of the corresponding graph as TXT to do post processing on it.

4   Benefits of online scientific models

4.1   Online services for the undergraduate level

For undergraduate level, this library can give support for both instructors and students.
1. **Staff support**: This portal gives any lecturer a very rich library of models that he can exercised during the lecture. While he will use the board to make the necessary mathematical derivations, he will be able directly to show students how parameter can affect the final performance.
2. **Student support**: The library of the java applets is also useful for students to exercise the models and get deeper understanding. They can develop reports from the applets that contains all input parameters and output performance to hand it over to the instructor. They also can share their experience by exchanging XML files containing input parameters of the model. Though not developed yet, we plan to have online library of XML files where users can save these files on the server for others to download.

4.2   Online services for the graduate level

For master and Ph.D students, we see important value for the library.
1. **Shortening the research cycle**: Many of the models currently exist in the library are models that were published in scientific refereed journals. The aim is to assist researchers in developing the required research background they need in the begging of their research programs. Electrical engineering researchers normally need to develop the models they use to compare their results. To do that, they start by understanding scientific papers and then program the models they are going to use. While the accessibility of online papers is now common, the time spent in converting a mathematical model published in a paper into a working program is a huge overhead in terms of time. The library offers a quick access to scientific models to test, and compare results without need to spend time programming them.
2. *Cross referencing with scientific papers*: Though this is not implemented yet, but we envision the time when this library of scientific models of published papers can be cross referenced with the original online papers from scientific publication houses. It will be

very helpful for researcher who reads the scientific paper or exercising the model to easily switch from one to another.

4.3   Online User generated content (on going work)

For the library to be populated with thousands of models, we understand that the cooperation of the scientific community is very important. Currently a free source is underdevelopment that will be available to all researchers to use to develop their own scientific models in the form of java applets. The produced applets will have the same interface as the ones already exist in the library to ensure library conformity. Users will be allowed to add their applets to the library with their credentials thus allowing a fast growing library of scientific models.

References:

[1] Sun microsystems http://www.sun.com/java/
[2] Java development environment, http://www.eclipse.org/
[3] Colt provides a set of Open Source Libraries for High Performance Scientific and Technical Computing in Java, http://dsd.lbl.gov/~hoschek/colt/
[4] Scientific library used for complex and trigonometric functions, http://jsci.sourceforge.net/
[5] Java library to generate PDF reports from the applets, http://itextpdf.sourceforge.net/
[6] Mathematical development platform for science and engineering http://www.scilab.org/
[7] Software technical help http://www.imodelit.com/Admin/Files/Software_help.pdf

Author(s):

Hatem, Elrefaei, Dr.
Ain Shams U., Faculty of Engineering, Mathematics and Engineering Physics Department.
1 Elsarayat St., Abbassia, Cairo, Egypt
Also CEO of EngNet Inc.
Smart Village, Building B115, Giza, Egypt
hatem.elrefaei@imdoleit.com

Tamer, Aboelfadel, Dr.
Cairo U., Faculty of Engineering, Electronics and Communication Engineering Department.
Giza Square, Giza, Egypt.

Maha, Elmeseery, Software Engineer
EngNet Inc.
Smart Village, Building B115, Giza, Egypt

Ali, Elmofty, Java developer
EngNet Inc.
Smart Village, Building B115, Giza, Egypt

Marwah, Shafee, Microwave Engineer
EngNet Inc.
Smart Village, Building B115, Giza, Egypt