
AC 2012-3828: DYNAMICS SIMULATION USING SERVER CLUSTER TECHNOLOGY

Dr. Claudio Luiz Curotto, Federal University of Parana, Brazil

Claudio Luiz Curotto is Associate Professor, UFPR, Curitiba, PR, Brazil, 1980-present. He was Visiting Professor, OU, Norman, Okla., USA, 2011-2012; Postdoc, civil engineering, UFRJ, Rio de Janeiro, RJ, Brazil, 2005-2007; and Visiting Professor, KUL, Leuven, Belgium, 2006. He has a D.Sc., civil engineering, UFRJ, Rio de Janeiro, RJ, Brazil 2003; a M.Sc., civil engineering, UFRJ, Rio de Janeiro, RJ, Brazil, 1981; a civil engineer degree, UEL, Londrina, PR, Brazil, 1976; and a Licentiate in Mathematics, UEL, Londrina, PR, Brazil, 1975.

Dr. Kurt C. Gramoll, University of Oklahoma

Dynamics Simulation using Server Cluster Technology

Abstract

Learning Dynamics through Computational Experiments (LDCE) is a method that can achieve better results in basic engineering Dynamics courses. This method is a Problem Based Learning (PBL) approach, where students seek for knowledge by developing projects during the course.

Projects include one or more experiments through which students can learn how a dynamics problem works. They are able to modify parameters and retrieve graphical and numerical results. Their primary task is to develop the theoretical basis of the problems and compare the numerical results with those produced by the experiments.

To accommodate a large number of students simultaneously accessing such experiments, a High Performance Computing (HPC) server cluster technology will be developed and implemented. The cluster will also help for those experiments that are computational intensive.

The students will be using a web browser on their own device, where the parameters of the experiment can be modified by a client application. The client sends input data to a HPC server cluster simulation application that runs the experiment. The server sends output data back to the student's device where the web browser shows motion and numerical results of the experiment.

The implementation of a time-based collision physics engine on the server cluster uses parallel programming techniques for multi core processors.

The HPC server cluster approach allows the development of different clients on many different platforms. This paper will show the details of the implementation of both, the client and the server applications as well as how the data interchanges between them. The paper will also illustrate the behavior of the system with some experiments.

The development of the system includes only open source free software. The complete software package will be of both free distribution and access.

Introduction

The focus of this research relates to a basic Dynamics course, taught to sophomore engineering (civil and mechanical) college students. Hibbeler¹ covers all course contents, comprising Kinematics and Kinetics of Particles and Planar Kinematics and Kinetics of Rigid Bodies, including Force and Acceleration, Work and Energy, Impulse and Momentum.

Currently, there are only a few research efforts investigating methods to improve the learning quality in basic Dynamics. These efforts range from using spreadsheets for formulas and graphics² to developing sophisticated applets³⁻⁵. Applets are small programs, usually written in

Java or Flash (Flashlets), which can be included in an HTML page and viewed by a web browser displaying graphics and simulating physical problems.

Since the main problem is showing moving objects, many of the current approaches include the development of multimedia content. These approaches may or not use applets. In that direction, Liang⁶ developed solid objects of mechanisms for Dynamics classes.

Cornwell⁷ has been using computational tools to support the learning process of Dynamics since 1996. All students of the Rose-Hulman Institute of Technology, including his students, have been using Maple⁸ or Mathematica⁹ in their Calculus classes since that date. In addition to these tools, he utilizes simulations for around half of the problems used in his classes using Working Model 2D¹⁰. He also uses concept maps¹¹, which are pictorial representations of the course topics, to help students better understand the relationship among them. He reported all of his experiences in the 2000 ASEE Conference¹².

Since 1998, Gramoll¹³⁻¹⁵ has been developing multimedia content for several engineering basic courses, included among them Dynamics. This content is available for free, to all teachers wishing to use it in their courses, at the eCourses¹⁶ web site. This web site includes all instructional material to conduct a course. There is no cost to either the instructor or student using it. Features include eBook (with simulations or Flashlets in each topic), database of homework/quiz/test problems and their solutions, lectures in both QuickTime and Flash format, computer grading, and utilities. To help facilitate communications between students, instructors, and TAs there is an integrated web board and collaborative drawing board. The instructor controls and manages his own web-based course.

However, it must be noticed that, the utilization of multimedia content and computational tools must be carried out along with the implementation of a new learning process methodology. This methodology must include new ways for both the learning and the assessment processes. From authors' experience, the simple use of multimedia content together with the traditional methods of learning does not achieve satisfactory results. The students will simply ignore the multimedia content if it is used only as an additional reference. The multimedia content has to be fully integrated with both, the learning and the assessment processes. Thus, Problem Based Learning (PBL) may be a better methodology to be applied in that case.

PBL was initially conceived in 1960 for Medical Schools, but today PBL is common in many other fields such as Engineering. Among the several references related to PBL, it can be found one particularly useful, the Ribeiro's paper¹⁷, related to this methodology applied to engineering courses. There are other methods similar to PBL, such as the one adopted by Gray¹⁸, named Problem Centered Approach (PCA), which is similar to the Gramoll's approach¹⁶.

Indeed, PCA and PBL are similar because both of them show real problems in order to motivate the theory learning. The main difference among them is that PBL does not make use of a formal presentation of the content. The students are the players of the learning process, and the teacher is only a facilitator instead of a primary deliver of information.

On the other hand, in the PCA methodology, the teacher continues to be the main player. The learning process uses multimedia contents, but the assessment process necessarily does not.

However, the content continues to be taught in traditional classes. Gray¹⁸ justifies PCA in high enrollment courses such as Statics and Dynamics since PBL demands much more time per student from the teacher.

Since 2004, the first author has been using PCA in Dynamics classes. A multimedia projector shows PDF files with the course content in traditional classes. Almost all of the problems developed in class have a movie produced by Working Model 2D¹⁰. The assessment includes a set of eCourses¹⁵ homeworks and two projects (a more complex problems), which must be developed by using computational tools (spreadsheets, word editors, etc.). The students may work at home, in groups of two, orienting their study along the semester developing the projects.

The main goal is to implement the PBL methodology integrating the multimedia content with both, the learning and the assessment processes. While eCourses¹⁵ is suitable for the multimedia content and the homeworks, there is a lack of project oriented learning. That is the essence of this research.

In order to fill out that lack, a resource has to be implemented, the project. A project is a more complex problem, similar to a real engineering task, that includes, as much as possible, topics of the course content. A project includes a computational experiment where students can see how a dynamics problem works by modifying various parameters and then obtaining the corresponding graphical and numerical results. Their main task is to develop the theoretical basis of the project comparing their numerical results with those produced by the experiment. The assessment is personal and done by an exam where the student must answer questions about his or her project.

Three basic premises will give the directions of that implementation, described to follows:

1. A student must be able to access an experiment in any device, a computer, a smartphone, a tablet, etc., by using only a web browser without any plugin or extension.
2. An experiment must provide graphical and numerical results. These results must have enough accuracy to be compared with those theoretical computed by the student.
3. Since the entire package will be of free distribution, only open source free software must be used in the development.

In order to fulfill those premises, there are still a few choices. The most known software packages such as Working Model 2D¹⁰ and Adams¹⁹ are automatically out of scope. The first program is expensive and is obsolete (not updated in over 10 years). The second program is free for students, but it is not for faculty and academic institutions. Furthermore, it is exceedingly complex to be used in basic Dynamics (there is not even a simple example in their Academic Customer Case Studies).

The only logical solution is to create a tailored system, which will be call “B2DII” in this paper. The main goal of this research is to develop such a system, which details will be described in the following sections.

The results of two experiments will be presented demonstrating a successful implementation. A discussion is provided about the success of the approach, and what future work needs to be done.

Tools

A basic windows laptop (Intel i5 M430 2.27 GHz processor (2 cores) 4 GB of RAM installed memory), running Microsoft (MS) Windows 7 Enterprise SP1, is the main hardware development tool.

A 36 nodes server cluster (each node has 2 Intel Xeon X5650 2.66 GHz processors (6 cores) 12 GB of RAM installed memory), running MS Windows Server 2008 HPC Edition SP1. The total number of cores in the cluster is 384 and has a theoretical calculation capacity of 4.1 teraFLOPS (floating point operations per second).

MS Visual Studio 2010²⁰ (MSVC) (MS Visual C and C⁺⁺) and the Intel Parallel Studio XE²¹ (C⁺⁺ Composer and Parallel Advisor) are the main software development tools. HPC Pack 2008 R2 SDK²² provides job management on the server cluster. Box2D²³ is a C⁺⁺ 2D physics engine originally designed for games.

Libwebsockets²⁴ is a web socket²⁵ library built in C and MS Winsock²⁶. JavaScript can be used on the client side. It supports HyBi²⁷ 00 to 13. Winserv²⁸ creates a Windows service that runs any application.

HTML5²⁹ is the specification that defines the 5th major revision of the core language of the World Wide Web: the Hypertext Markup Language (HTML). HTML5²⁹ supports the Canvas element, which allows dynamic, scriptable rendering of 2D shapes and bitmap images. Safari³⁰ Develop menu provides tools for web developers such as easy debugging of JavaScript.

Macromedia Dreamweaver 8³¹ is the web design software, visual HTML and code editor. HTML5²⁹ is still a draft specification, but it will certainly determine the future for web content as shown by the recent news: the massive adherence of the web content publishers, including several newspapers; Adobe ceasing development of Flash for mobile in favor of HTML5; and likelihood that Microsoft will drop Silverlight in favor of HTML5.

The B2DII System

The B2DII system involves different components that allow any user to setup and run a complex dynamics simulation from a web page. The system architecture is outlined in Figure 1. On the client side, a user can use any web-enabled device, such as a personal computer (desktop or laptop), a smartphone, or a tablet (either iPad or Android). The device runs a web browser supporting HTML5²⁹ Canvas and web sockets²⁵. Three browser currently support these features: Chrome³², Firefox³³ and Safari³⁰. The user begins the session by opening the experiment's web page.

The B2WebSocket application runs as a service on the head node of the server cluster. The Winserv²⁸ utility supports such a task. The service will automatically restarts in the case of a failure. Thus, B2WebSocket is always listening for requests on the head node of the server cluster.

The experiment's web page includes only HTML5²⁹ and JavaScript code that sends a request (using text string format) to B2WebSocket through a web socket²⁵ connection. Two different

requests are possible: define – retrieves a one step movie that draws the model; run – retrieves a multi step movie that shows the motion of the model. The movie retrieves using text string format. This conversation follows the web socket protocol²⁷ implemented in each web browser. When a movie is retrieved, the connection closes which frees resources on the server cluster.

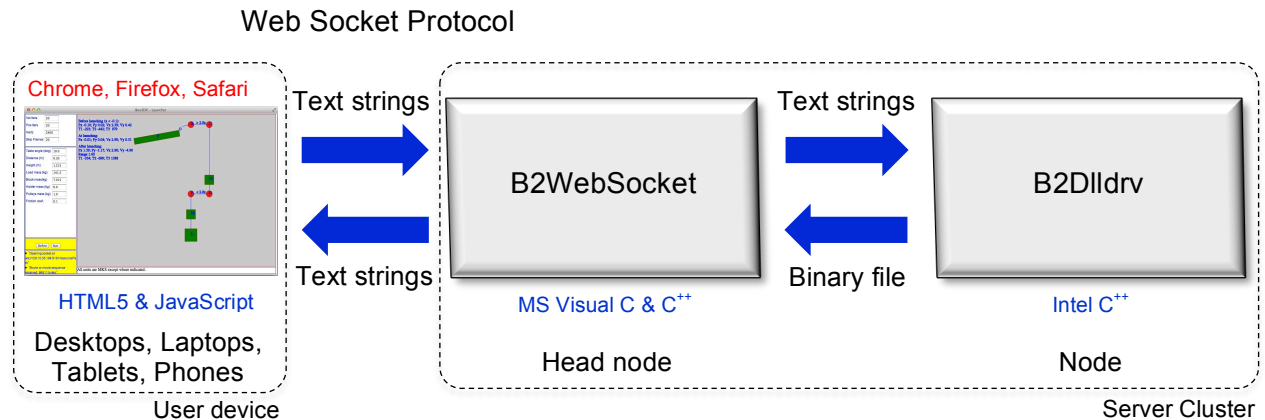


Fig. 1 – B2DII Architecture

B2WebSocket is a 32 bits application developed using the MSVC (MS C & C⁺⁺), the Libwebsockets²⁴ library and the HPC Pack 2008 R2 SDK²². When B2WebSocket receives a request, a B2Dlldrv job starts in one of the available server cluster nodes through the HPC Management System. HPC allows starting jobs in only one single processor (or in as many as necessary, till the maximum available) of any node of the server cluster.

The B2Dlldrv job starts using the parameters sent by the web page. When the job finishes, a binary file is available to be sent back to the web page, which will convert it back into text strings. When the system runs in a standalone computer, B2Dlldrv starts as a sub-process of B2WebSocket. This situation is particularly useful during the development, when only a standalone computer is available.

B2Dlldrv is a 64 bits application developed using the MSVC (Intel C⁺⁺) and the Box2D²³ library. This library satisfies almost all needs in the research scope. Parallel programming has not used yet, but the code is ready for that task. The most time consuming task of this application, the generation of the binary file containing the movie of the motion, uses pointers, which are 17 times faster than vectors. This application can also run as a standalone program by using a basic command line and displaying the movie on the monitor of the computer where it is running. That feature is particularly useful during the development and also may allow distributing experiments to be run in personal computers without the use of both, a web browser or the server cluster.

Results

Two experiments illustrate the behavior of the developed software. They are available for testing at <http://dcluster.ou.edu>.

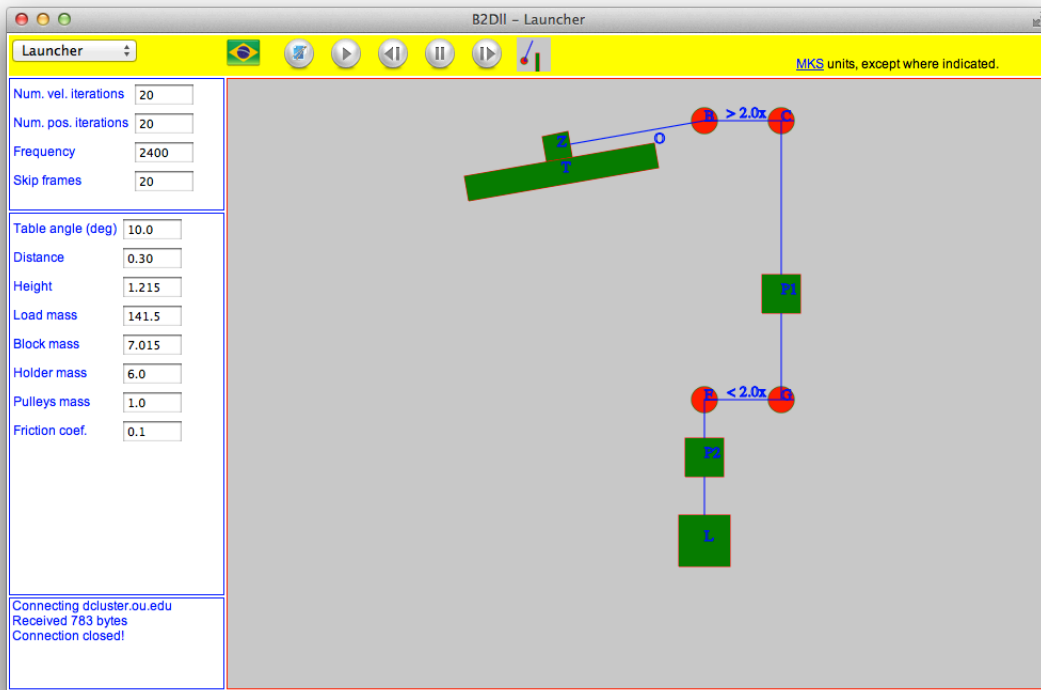


Figure 2 – Launcher – Model definition

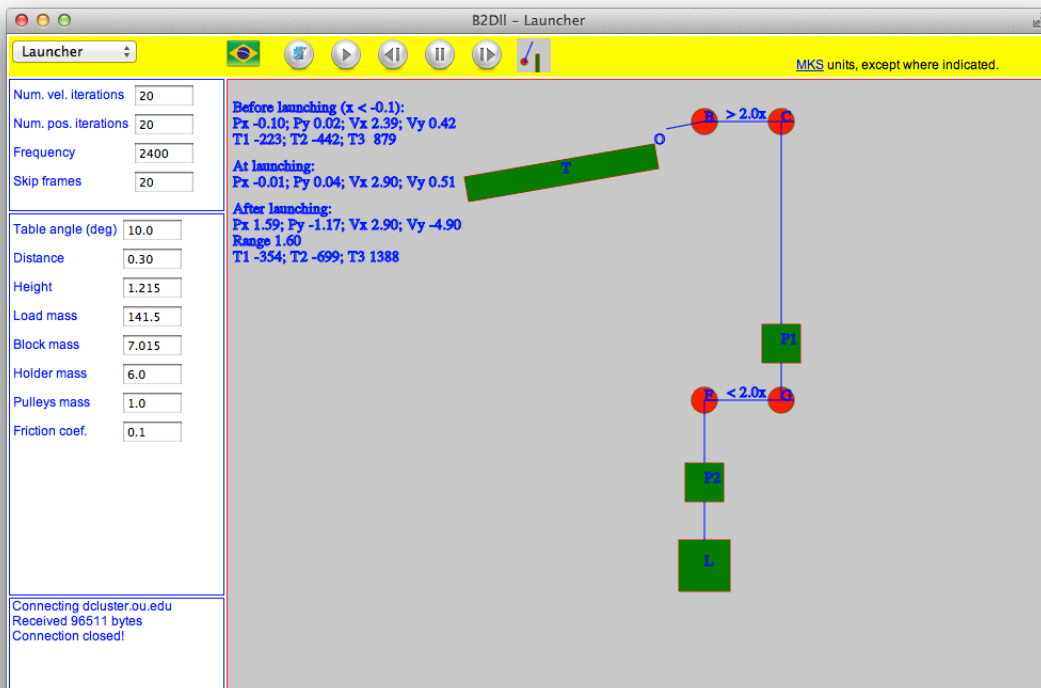


Figure 3 – Launcher – End of the motion

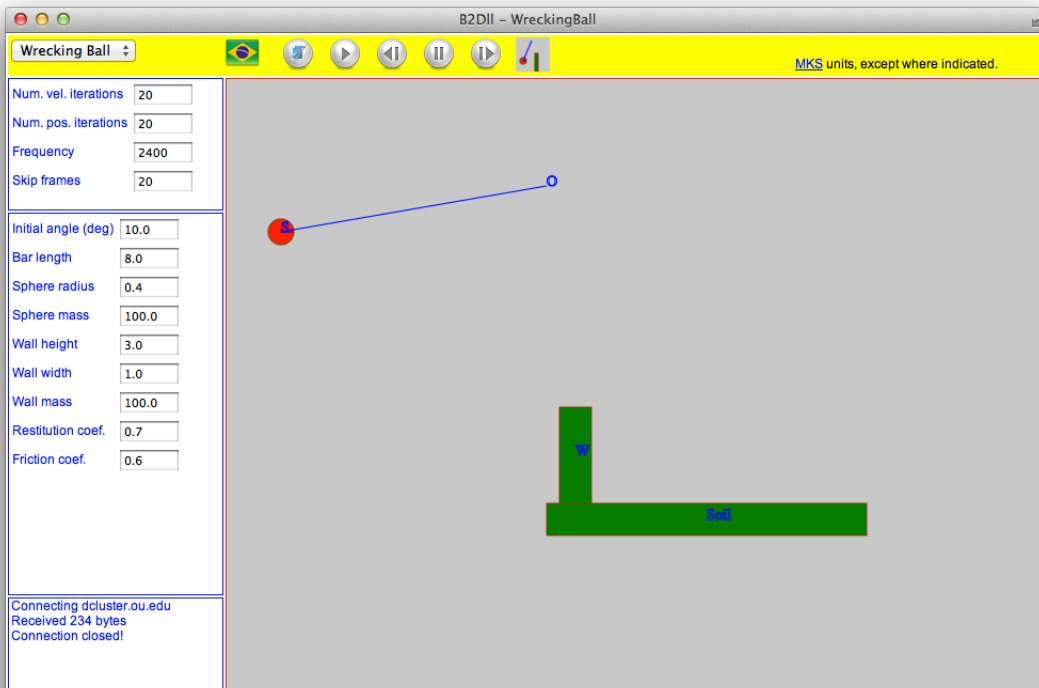


Figure 4 – Wrecking Ball – Model definition

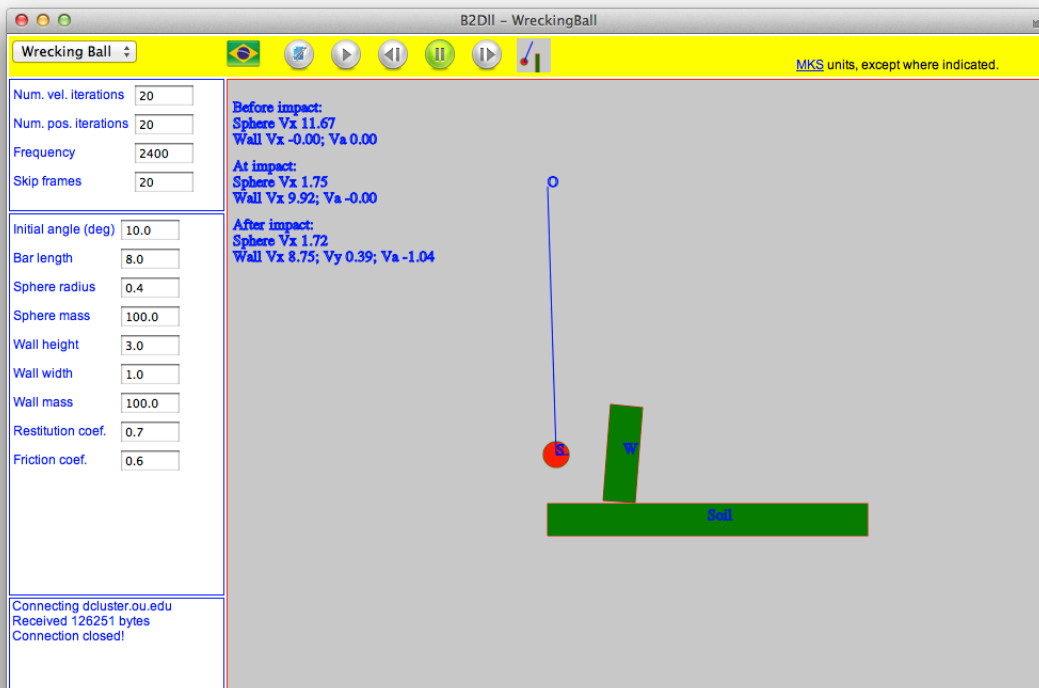


Figure 5 – Wrecking Ball – After the impact

The first experiment shows an object launcher powered by a load through two pulley systems. The main goal is to evaluate the maximum range of the launcher by varying the table angle. It includes Rectilinear Kinematics: Continuous Motion, Dependent Motion Analysis, Kinetics of a Particle, Equations of Motion, Motion of a Projectile and Friction. Figure 2 shows the model definition and Figure 3 shows the model at the end of the motion. Each pulley system has a pulley ratio equal to 2. For the upper pulley system, that means the box P1 moves twice faster than block Z. For the lower pulley system, that means the box P2 moves twice faster than P1. That is the way B2Dll models multi-pulley systems. The blocks P1 and P2 represent the masses of the pulleys.

The second experiment shows a wrecking ball colliding with a wall. The main goal is to evaluate the initial velocities of the wall right after the impact. It includes Planar Kinematics of a Rigid Body, Rotation about a Fixed Axis, Moment of Inertia, Equations of Motion: Rotation about a Fixed Axis, Impact and Friction. Figure 4 shows the model definition and Figure 5 shows the experiment a few moments after the impact.

Since the students must answer questions mostly about the intermediary results of the experiment, only a few numerical results should be given in the movie.

Conclusions and Future Work

Several tests were made on a variety of devices, such as MS Windows laptops and desktops, Apple Mac computers, iPhones, iPads and Android tablets. All tests were successful.

Since B2WebSocket can handle hundreds of requests using multi threads and hundreds of cores are available on the server cluster, hundreds of users can be attended simultaneously.

The chosen time-based collision physics engine fits extremely well in the research scope. Since it does not provide acceleration results, this feature has to be implemented as soon as possible.

Parallel programming still has to be done, as well as the implementation of some cosmetic effects on the web page, more experiments and a complete user's guide including the specifications of the projects.

Since the system is new and just started operation on spring semester, 2012, there is no students' feedback. The authors plan to introduce this methodology in a Dynamics course this academic year.

The entire package will be available for free distribution as soon as possible, since its development includes only open source free software.

Acknowledgment

The University of Oklahoma has partially supported the first author during his visiting professor appointment.

References

1. Hibbeler, R.C., *Engineering Mechanics: Combined Statics & Dynamics*, 12/E, Pearson Prentice Hall, Upper Saddle River, NJ, USA, 2010.
2. Fowler, M., *Physics 581 Part 1: Teaching Dynamics with Excel97*, <http://galileo.phys.virginia.edu/classes/581,1998> [Jan/2012].
3. Fowler, M., *Fowler's Physics Applets*, http://galileoandstein.physics.virginia.edu/more_stuff/Applets, 1998 [Jan/2012].
4. Fowler, M., *Fowler's Physics Flashlets*, http://galileoandstein.physics.virginia.edu/more_stuff/flashlets, 2003 [Jan/2012].
5. Devenport, W. et al., *Java Applets for Engineering Education*, Proc. IEEE Annual Conf., San Jose, CA, USA, Vol. 1, pp. 91-92, 1999. <http://www.engapplets.vt.edu>.
6. Liang, W., *Using Multimedia in Teaching Dynamics*, Proc. ASEE Conf. for Ind. And Ed. Collab., USA, 2002.
7. Cornwell, P., *Teaching Dynamics using Modern Tools*, Computers in Education Journal, Oct-Dec, 1996.
8. Maplesoft - Waterloo Maple Inc., *Maple 14*, <http://www.maplesoft.com>, 2010 [Jan/2012].
9. Wolfram Research, *Mathematica 7*, <http://www.wolfram.com>, 2010 [Jan/2012].
10. Design Simulation Technologies Inc., *Working Model 2D*, <http://www.design-simulation.com/WM2D>, 2006 [Jan/2012].
11. Cornwell, P., *Concept Maps in the Mechanical Engineering Curriculum*, Proc. of the ASEE Annual Conf., Washington, DC, USA, 1996.
12. Cornwell, P., *Dynamics Evolution – Chance or Design*, Proc. ASEE Annual Conf., Saint Louis, MO, USA, 2000.
13. Vikas, Y. & Gramoll, K., *Design and Implementation of an Internet Portal for Basic Statics and Dynamics Courses*, Proc. ASEE Annual Conf., Albuquerque, NM, USA, 2001.
14. Gramoll, K., *A Web-Based Electronic Book (eBook) for Solid Mechanics*, Proc. ASEE Annual Conf. (Mech. Div. Best Pres. Award), Honolulu, HI, USA, 2007.
15. Gramoll, K., *eCourses Dynamics – Multimedia Engineering Dynamics*, <https://ecourses.ou.edu/cgi-bin/ebook.cgi?topic=dy>, 2004 [Jan/2012].
16. EML/OU, *Engineering Media Lab*, <http://eml.ou.edu>, 2011 [Jan/2012].
17. Ribeiro, L.R.C., *The Pros and Cons of Problem-Based Learning from the Teacher's Standpoint*, Journal of University Teaching & Learning Practice, 8(1), 2011. <http://ro.uow.edu.au/jutlp/vol8/iss1/4>.
18. Gray, G.L. & Constanzo, F., *A Problem-Centered Approach to Dynamics*, Proc. ASEE Annual Conf. (Mech. Div. Best Pres. Award), Pittsburgh, PA, USA, 2008.
19. MSC Software Corp., *Adams – Multibody Dynamics Simulation*, <http://www.mscsoftware.com/Products/CAE-Tools/Adams.aspx>, 2011 [Jan/2012].
20. Microsoft Corp., *Microsoft Visual Studio 2010*, <http://msdn.microsoft.com/en-us/vstudio>, 2011 [Jan/2012].
21. Intel Inc., <http://software.intel.com/en-us/articles/intel-xe-product-comparison>, 2011 [Jan/2012].
22. Microsoft Corp., *HPC Pack 2008 R2 SDK*, [http://msdn.microsoft.com/en-us/library/cc853440\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/cc853440(VS.85).aspx), 2011 [Jan/2012].
23. Catto, E., *Box2D – C++ 2D Physics Engine for Games*, <http://box2d.org>, 2011 [Jan/2012].
24. Green, A., *C Websockets Server Library*, <http://git.warmcat.com/cgi-bin/cgit/libwebsockets>, 2011 [Jan/2012].
25. Hickson, I., ed., *The WebSocket API*, <http://dev.w3.org/html5/websockets>, 2011 [Jan/2012].
26. Microsoft Corp., *Winsock - Windows Sockets 2*, [http://msdn.microsoft.com/en-us/library/windows/desktop/ms740673\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740673(v=vs.85).aspx), 2011 [Jan/2012].
27. Fette, I., ed., *The WebSocket protocol*, <http://tools.ietf.org/html/draft-ietf-hybi-thewebsocketprotocol-17>, 2011 [Jan/2012].
28. Kovalenko, A., *Winserv*, <http://www.sw4me.com/wiki/Winserv>, 2005 [Jan/2012].
29. Hickson, I., ed., *HTML5 - A vocabulary and associated APIs for HTML and XHTML*, <http://dev.w3.org/html5/spec>, 2011 [Jan/2012].
30. Apple Inc., *Safari 5.1.2*, <http://www.apple.com/safari/download>, 2011 [Jan/2012].
31. Adobe Systems Inc., *Macromedia Dreamweaver 8*, <http://www.adobe.com>, 2011 [Jan/2012].
32. Google Inc., *Google Chrome 17.0*, <http://www.google.com/chrome>, 2011 [Jan/2012].
33. Mozilla Foundation, *Firefox 9.0.1*, <http://www.mozilla.org/firefox>, 2011 [Jan/2012].