CODEE Journal

Volume 8 Article 2

7-26-2011

A Project Approach in Differential Equations Courses

Robert Borrelli

Courtney Coleman

Recommended Citation

Borrelli, Robert and Coleman, Courtney (2011) "A Project Approach in Differential Equations Courses," *CODEE Journal*: Vol. 8, Article 2.
Available at: http://scholarship.claremont.edu/codee/vol8/iss1/2

A Project Approach in Differential Equations Courses

Robert Borrelli and Courtney Coleman
Harvey Mudd College

Keywords: project-based learning
Manuscript received on May 11, 2011; published on July 26, 2011.

Abstract: From the late 60's the mathematics department at Harvey Mudd College (HMC) has been active in introducing independent study projects into its math courses, especially courses involving differential equations. This paper describes two such approaches and the features that were constructed to support them. With the change in technology in the late 90's, it was clear that these project approaches needed to be updated. These changes are underway and are described in this article.

1 Introduction

Before computers became easily accessible to students, ordinary differential equations (ODEs) were briefly covered in calculus courses for the benefit of science and engineering students. The differential equations topics usually covered in calculus courses were mostly techniques for finding solution formulas, especially for linear ODEs with constant coefficients. Some easy applications like falling bodies were discussed. Since computers weren't available to students, graphing of solutions had to be done with paper-and-pencil. For example, when we arrived at Harvey Mudd College (HMC) in the early 60's, HMC had some 350 students and only four majors: Engineering, Physics, Chemistry and Mathematics. So all HMC students were required to take a common core of mathematics for the first two years. The sole textbook for the course was the textbook *University Mathematics*, whose principal author was the founding chair of the HMC Math Department. A full-fledged introductory-level ordinary differential equations course was not offered until the Junior year. The computing resources at HMC in those early days were pretty dismal and almost no effort was made to bring interesting applications into the ODE courses. So the topics covered in that Junior-level course were mostly theoretical, but some applications like the two-body problem were treated because solution formulas could be derived. Also, ODE textbooks in those early days did not discuss mathematical modeling or stress the graphing of solutions.

With the emergence and ready availability of computers in the 70's, things began to change nationally. Pencil-and-paper graphing of solutions was no longer stressed and computer software could be used to experiment with the behavior of solutions when the data in differential equations change. So textbooks began to include “labs” or computer experiments appropriate for an introductory ODE course. In the late 60’s, we decided to create a Junior-level year-long course we called *Applied Analysis* which would be required of all HMC students (there was a special section for math majors). This course would include partial as well as ordinary differential equations. The purpose of such a course was to give all HMC students a glimpse into some important areas of applicable mathematics along with some traditional and contemporary applications. To make our Applied Analysis course more appealing to all HMC students regardless of major, we introduced “open-ended” projects as a required element of the course. The purpose of the projects was to give students a substantial practical application of the abstract mathematical ideas covered in the course. The open-ended project concept was picked up in other HMC courses as well.

We discuss ODE computer experiments and “open-ended” projects in more detail below.

2 Computer Experiments in ODE Courses

The simplest approach to independent project work (which is less labor intensive for the instructor) are “hands-on” experiments or case studies which ask the student, alone or as part of a team, to study the behavior of solutions of ODEs and communicate the results either orally or in a written report. This approach works well in introductory differential equations courses. Ideally, platforms for these experiments should support

- A robust and reliable, easy-to-use ODE solver that can handle a fairly wide selection of predefined functions. It would be nice (but not crucial) if the solver accepted user-defined functions and handled first-order systems with at least three state variables.
- Two and three dimensional graphics capability for visualizing the output of the solver.
- A printer for producing hard copy of graphics displays.

Usually there is little choice in hardware/software platforms provided by the institution, especially at very small colleges (or even very large universities). By now, most institutions have at least one computer lab with a pre-selected commercial software package which contains several ODE solvers. Computer labs are usually staffed with lab assistants who may or may not know anything about ODEs. Software ODE solvers have their own peculiarities depending on the algorithms employed, the hardware platform, and so on. All solvers allow some adjustment of the solvers parameters. Some solvers are menu-driven and interact with the user through elaborate displays, while some interact in a linear command/response stream or via a graphical user interface. Some solvers act like oracles (i.e., giving no hint that a mistake has occurred), whereas others print warnings and/or supply information about what went wrong. However, some peculiarities common to all solvers affect the accuracy of the output. Users should be aware of these peculiarities and recognize them when they occur; solvers cannot be run mindlessly with the results always accepted without question. Visual displays of solutions of ODEs can be misleading or difficult to interpret unless the basic theory is well in hand. It is ironic that the introduction of computers into a course often leads to a heightened interest in the theory rather than less.

Although many resources are available to ODE instructors who encourage experimentation, we have observed that most instructors who opt for this approach rely mainly on a "modeling and computing" textbook (see Appendix I). Many of these textbooks come with "labs" and easy-to-use solvers that run on the student's own platform. For example, the solver *ODE Architect* comes with several textbooks and contains many labs and examples. CODEE's java-based solver called ODE Toolkit is delivered for free over the internet and, hence, allows students to use it on their own platforms in a wireless-equipped classroom. To download ODE Toolkit look at this CODEE website under "Software."

Most instructors follow a "case study/experimentation" approach rather than an "open-ended" approach (described below). This is hardly surprising because the "open-ended" approach is much more labor intensive for the instructor, especially when large classes are involved. The reason, of course, is that the overworked instructor must spend many hours searching for suitable material, become familiar with the peculiarities of the software, and carefully evaluate student lab work.

3 HMC's Open-Ended Project Concept

Currently, HMC offers majors only in six areas: Engineering, Physics, Chemistry, Mathematics, Biology, and Computer Science. So it is not surprising that all HMC students are required to take a 2-year core of courses involving all of these areas before declaring a major. Early on, we decided to make our upper division differential equations courses more relevant to all HMC students (and not just math majors) by requiring open-ended modeling projects based on a physical system. These projects are a formal part of those courses and bring the Senior Thesis concept to all students enrolled in courses involving differential equations and not just math majors. Since working together in teams is an important part of the training of scientists and engineers (and mathematicians who work in industry), we thought it would be a good idea to use a team approach for these independent study projects.

Here is a brief overview of how our project concept in differential equations classes works: At the beginning of the course, each student is directed to a file of project topics and given a copy of the *Handbook for Projects* which contains, among other things, summaries of past projects. The Handbook was designed to help students organize the necessary resources for the successful completion of a project. It describes the project experience in some detail and lays out a calendar for project work as well as a check list for final reports. Also in the *Handbook* is a *Guide to Scientific Literature Searching* which students might find helpful in their project work. To see one version of the *Handbook* visit `http://ccdl.libraries.claremont.edu/u?/mmm,4688`.

It is the instructor's task to find a collection of open-ended projects and assemble the project files before classes begin (this is the time-consuming part!). Naturally, the projects in the file would involve mathematical ideas to be covered in the course. Folders in the project file contain relevant published papers as well as final reports written by earlier teams. In the first week, students self-select into teams and scan the list of possible projects supplied by the faculty member (or choose a project of their own). We found that the ideal size for a project team is three to five members; hence, there are enough students to handle a job of broad scope, without having so many that they would get in each other's way. There is no hard and fast rule against one-person projects; they are allowed in special cases. In general, however, we found that a one-person team would not have the time to handle a significant project adequately during the course or have the benefit of multiple points of view. The team then submits a proposal for their project outlining what kind of an outcome they wish to achieve. The faculty member in charge then decides whether or not the team has the resources to accomplish their stated goal and works together with each team to narrow their goal down to something doable during the course. Project work goes on during the course and is monitored by a midterm progress report. Finally, students are dismissed from the last two weeks of the course so that the teams can finish their final written reports. In assigning project grades to individual students in a team, the faculty member makes use of a *Project Self-Grading Form* in the *Handbook*.

Independent projects within a course are open-ended in nature. They begin with the construction of a model and end with an analysis of how well the model fits the system of interest. Naturally, the models chosen must be appropriate to the level and nature of the course, and the mathematical analysis of the model should give a substantial practical application of the abstract mathematical ideas covered in the course. The successful completion of a project requires that each student team communicates its model and its analysis via a final report. In the *Handbook* there are abstracts for all the final project reports since the 1972 inception of the projects program. These projects came from 6 courses at HMC, Pomona College and the Claremont Graduate University and the reports themselves are housed in a file which is accessible to students. There were at one point about 200 final reports on file (listed in the *Handbook*).

As an example of a student project, look at the paper *Modeling Caffeine in the Human Body* by Lindsay Crowl '04 (`http://www.math.hmc.edu/~depillis/MATH164/MATH164_StudentProjects_2003/LCROWL/CoffeeReport.doc`). There are many abstracts in the project file listing at the end of the *Handbook*, but unfortunately the projects themselves were not saved online.

4 Setting Up Your Own Open-Ended Projects Program

If you are thinking about setting up an open-ended projects component of your ODE course, then there are several things to do in advance.

Getting started. It is critically important to have a decent ODE solver available that students can use and feel comfortable with. Larger institutions probably have Matlab or Maple or Mathematica available in a computer lab somewhere on campus; these packages have everything needed to solve and graph solutions of ODEs. The syntax used in these packages takes some getting used to, but many students manage to learn it on their own with little difficulty. These packages are prohibitively expensive for many smaller institutions to support large numbers of users, but there are other ODE solvers that are delivered free with some textbooks (like, for example, *ODE Architect*) or available free over the internet (like, for example, pplane, dfield, *ODEFactory* and ODE Toolkit which all run under JAVA). The navigation in these packages is point-and-click and is intuitively easy to learn. To access pplane and dfield, *ODEFactory* or ODE Toolkit just look at this CODEE website under "Software."

Finding Appropriate Projects. Success in using the open-ended project approach in a differential equations course depends, of course, upon finding a half-dozen or more projects that are suitable for your own students. We think the best projects are those which involve the construction first of a model for some physical phenomenon, but theoretical projects may be appropriate for some mathematics students. There are many sources that can be consulted and some of these are listed in Appendix I and Appendix II, but there are many, many other sources. Perhaps your colleagues in other departments can suggest projects that would be suitable for your students (especially check with the Biology Department!). Project descriptions and supporting material should be ready to go on the first day of class. This is a bit of a chore for the instructor at first, but becomes easier later on when some projects can be "recycled" (that is why we have maintained a student-accessible file of completed projects which are abstracted in the *Handbook for Projects*).

Supporting Items. It would also be very helpful if you could design your own Handbook for Projects, perhaps along the lines of the one we used. To see it, visit `http://ccdl.libraries.claremont.edu/u?/mmm,4688`, but modernized along the lines of the current HMC Senior Thesis Handbook. To see the HMC Thesis handbook, visit `http://www.math.hmc.edu/seniorthesis/guidelines/handbook/`. Your handbook can also be online so that it can be easily for updated and distributed.

It is very important to create a calendar for work on projects during your course, and then to stick to it! For example look at the calendar in our *Handbook for Projects* (`http://ccdl.libraries.claremont.edu/u?/mmm,4688`).

We also suggest that you set up student-accessible project files which contain background material for the projects you have selected. Past reports can be kept on line and hence do not have to be kept in a physical place (as ours were).

Nowadays, students are usually asked to submit reports in LaTeX which is free and handles mathematical formulas very well. The HMC *Senior Thesis Handbook* has a primer on using LaTeX to create reports. Graphs can easily be imported into LaTeX reports as EPS or PDF files.

5 Facilities Developed in Support of Project Activity

Beginning in the early 1970's (and with the help of colleagues and generations of HMC students) we designed several features which proved to be useful in supporting the project concept within differential equations courses. We describe them below:

Textbooks. As there was no text for the Applied Analysis course, we decided to write one. We titled the text, *Mathematical Methods, Models and Applications for Engineers, Mathematicians and Scientists*. Every one of the 20 chapters began with an application.

The approach was firmly grounded in mathematics; proofs and models of real-world phenomena were both done carefully (in deference to our pure math colleagues). Mathematicians in particular, would feel comfortable using this text. Physics and Engineering instructors would find in the book more of the mathematical foundations of applied analysis and related areas than in most books of similar scope at this level. Work on the in-house text extended over the period 1968-1977, but it was never published. It contained more than 2200 typed pages. It is now freely available online. To see it, visit `http://ccdl.libraries.claremont.edu/col/mmm`. One look at it tells why the students affectionately (or not?) called it the "monster."

Sometime in the early 1980's the math core at HMC was changed to allow separate one-semester courses in Linear Algebra and Differential Equations, both at the sophomore level. Some parts of the "monster" were used as texts for those courses. When it became clear in the mid-80's that our in-house text was too large to attract a publisher, we decided to write a more focused introductory-level ODE text which emphasized modeling, applications and computing. Prentice-Hall published our text in 1987 under the title *Differential Equations: a Modeling Approach*. We used ODE Toolkit as the solver tool in our DE classes (see below). Perhaps our text was ahead of its time in emphasizing modeling and computer experimentation, but in any case the textbook was not a commercial success. We noticed with some satisfaction though that in the 1990's almost all introductory-level ODE books began emphasizing modeling, applications and computing. We significantly revised our 1987 ODE textbook and it was published by John Wiley & Sons in 1998 under the title *Differential Equations: a Modeling Perspective*. A second edition of this Wiley text was published in 2004. A CD-ROM with the solver *ODE Architect* (see below) was distributed along with this text.

Software. It is absolutely crucial to have reliable and affordable software around to do project work. In the 70's and 80's HMC didn't have access to such software so we decided to create our own. MATHLIB was written by three very bright HMC students in the early 1980's. MATHLIB was an effective tool for obtaining accurate results in a useful format for everything from plotting a set of simple data points to solving a large system of differential equations. Hence, MATHLIB was very useful to students doing project work (and it was free!). In the 80's a front-end for the ODE solvers in MATHLIB was designed; we called it ODE Toolkit. Unfortunately, MATHLIB was designed to run under the VAX/VMS operating system, and when DEC went belly-up in the 90's we had to mothball it.

From 1992-1997, HMC headquartered an NSF supported project to CODEE (Consortium for ODE Experiments) for the publishing of a newsletter that provided a regular source of ideas, inspiration, and experiments for ODE instructors. CODEE also produced a software solver package called *ODE Architect*, a Windows software package that provides a highly interactive software environment for constructing and exploring the user's own ODE models of real-world phenomena. In the summer of 2010, we received numerous reports that *ODE Architect* did not work under Windows 7 (shades of MATHLIB?). We asked Jeho Park, the Scientific Computing Specialist at Harvey Mudd College, to look into this matter. Jeho found a couple of facts to explain the reason of this incompatibility and has come up with a solution to the problem. His report appears elsewhere in this online CODEE Journal; to see it click "software" and then "*ODEArchitect*."

In 2007, HMC headquartered the 5-year NSF funded a CODEE project, Phase II. The goal of this project is to develop a website that would be useful for the teaching and learning of ODEs, primarily by encouraging broader use of modeling projects and computer experiments. This website (the one you are on!) contains articles, modules and projects, reviews and descriptions of useful ODE software solvers. Part of this project is to develop a Java-based ODE solver. This new solver, also called ODE Toolkit, is an ODE solver package which is available over the internet to instructors and students at no cost. It will run on any platform equipped with a web browser and Java (free to download from Sun Microsystems). The hardware requirements are rather modest, so it will function well even on older equipment. Instead of a command-line interface, ODE Toolkit features an easy-to-use graphical interface that minimizes typing and mouse clicks whenever possible. Users enter any first order system using a natural syntax; ODE Toolkit's parser allows for parameters and user-defined functions and can recognize standard mathematical and engineering functions (e.g., on-off functions, square waves, etc.). Users can also interact with computed solutions using 2D or 3D views via a simple tab navigation system. The viewing range of the graphing window can also adjust automatically to the computed solution, eliminating the tedium of setting and resetting viewing ranges. Users can open and save files containing all of their computations on their computer; these files can be shared across computing platforms. Documentation and tutorials are also available online. ODE Toolkit can run inside a browser window, eliminating the need for administrators to install the software; it can also be downloaded and run as a stand-alone program. ODE Toolkit's code is modular in design and hence will allow for future development. The current version of ODE Toolkit is always available on a server at Harvey Mudd College and can always be accessed at this CODEE website.

Journals. So many good project final reports were being produced that we thought it was a shame to just stick them away in a file cabinet somewhere. To make them available to a wider audience we decided to create a journal of interdisciplinary student research called *Interface*. First published in 1973 as a print publication, *Interface* was designed to promote and disseminate student research at Harvey Mudd College. *Interface* was distributed free of charge to every HMC student as well as HMC Trustees and Math Clinic clients. A few hundred copies of each issue were mailed out to alumni and other university math departments for a modest fee. Some articles in *Interface* have come from projects in differential equations courses. There is an *Author's Guidebook for Interface*. In this guidebook, the editors give some advice to potential authors on how to write up manuscripts intended for publication in *Interface*. To see the *Guide* and all the issues of *Interface* visit (`http://ccdl.libraries.claremont.edu/col/ija`). *Interface* ceased publication as a print journal in 2000 (strangely about the time we retired)—but it is coming back in 2011 as an online journal!

We remember one *Interface* paper in particular: it was a paper entitled "Mathematical Treatment of an Inner Ear Disease" by Robert Finley, an HMC student who graduated in 1976. Rob was at first undecided about which major to declare and what to do after graduating from HMC. The project Rob chose involved modeling mathematically the warping of the cochlea, a small structure within the inner ear. Using his model he was able to come up with a surgical technique for slowing down the warping and a number of local medical doctors became very interested in his paper. This initial experience convinced Rob to go to medical school, and he tells us that his *Interface* paper played an important role in gaining him admission to medical school. To see Finley's article, visit `http://ccdl.libraries.claremont.edu/u?/ija,189`.

Appendix I: Sources for Projects

CODEE Newsletters, 1992-1997. Published during the NSF/CODEE project, Phase 1. To see these newsletters, visit the CODEE web site and click on "CODEE Journal" and then on "Newsletters", or just visit `http://www.codee.org/library/newsletters`.

College Mathematics Journal. There are many articles in the CMJ that involve differential equations; to see a list of them all along with reviews, on this web site click on "CODEE Journal" and then click on "Reviews." There is one issue of the CMJ which is a special issue devoted to differential equations; it is Vol. 25, No. 5, November 1994; guest editor was Beverly West.

Differential Equations Laboratory Workbook, Borrelli, R., and Coleman, C., Wiley (1992). To see this Workbook, on this CODEE web site click on "CODEE Journal" and then click on "Projects."

Interface, especially Vol 1, #1; Vol 1, #2; Vol 4, #1; Vol 5, #1; Vol 8, #2; Vol 15, #1. To see all the actual issues of *Interface* visit `http://ccdl.libraries.claremont.edu/col/ija`.

Mathematical Models, Models and Applications for Engineers, Mathematicians and Scientists, Borrelli, R., and Coleman, C., unpublished. Now resides in the Claremont Colleges Digital Library. To see this text visit `http://ccdl.libraries.claremont.edu/col/mmm`.

Modeling and Visualization with ODE Architect: An Example Set. To see the Example Set, visit `http://www.math.hmc.edu/resources/odes/odearchitect/examples`.

Modules in Applied Mathematics, Vol 1, 2, 3, and 4: Differential Equations Models, Braun, M., Coleman, C., and Drew, D., editors, Springer-Verlag (1983).

ODE Architect and the *Architect Companion*, Wiley (1999). Produced by the NSF/CODEE project, Phase I. The companion was designed to accompany *ODE Architect* by extending the modules contained in the solver package *ODE Architect*. *ODE Architect* contains an extensive library of examples which illustrate various topics covered in ODE courses.

Official web site of the NSF/CODEE project, Phase II, is `http://www.codee.org`. It contains articles, modules and projects, reviews and the JAVA version of ODE Toolkit which contains a library of examples usually covered in ODE courses.

Revolutions in Differential Equations, editor Kallaher, M., Mathematics Association of America, MAA Notes #50 (1999). Explores ODEs with modern technology.

There are many other sources for finding projects involving differential equations which are suitable for teams of undergraduates. One in particular shows a lot of promise: the National Science Digital Library Pathways. Pathways are digital library portals developed and managed in partnership with organizations and institutions that have a history and expertise in serving their portal's target audiences. To see all the current partners, visit `http://nsdl.org/about/?pager=pathways`. There are currently at least two pathway partners where faculty can find projects that involve differential equations; they are: AMSER (visit `http://amser.org`) and MathDL (visit `http://mathdl.maa.org`). There is one collection within MathDL that contains material suitable for differential equations courses; that one is the Connected Curriculum Project (CCP). The CODEE project looks forward to someday becoming a pathway within NSDL (or a collection within MathDL).

Appendix II: Textbooks that Discuss Models Involving ODEs

Here are a few textbooks that emphasize modeling and computing in ODE courses. There are many others.

Blanchard, P., Devaney, R., and Hall, G., *Differential Equations*, 4th ed., Brooks Cole (2011).

Borrelli, R., and Coleman, C., *Differential Equations: A Modeling Perspective*, 2nd ed., Wiley (2004). Includes a CD-ROM with *ODE Architect* containing a solver tool, multi-media modules, and a library of examples. The *ODE Architect Companion* extends the features of the multi-media modules.

Brannan, J., and Boyce. W., *Differential Equations, An Introduction to Modern Methods and Applications*, 2nd ed., Wiley (2010).

Braun, M., *Differential Equations and Their Applications, An Introduction to Applied Mathematics*, Springer-Verlag (1992).

Edwards, C. H., and Penney, D. E., *Differential Equations Computing And Modeling*, 4th ed., Prentice Hall (2007).

Farlow, J., Hall, J., McDill, J., and West, B., *Differential Equations and Linear Algeba*, 2nd ed., Prentice Hall (2007).

Giordano, F., and Weir, M., *Differential Equations: A Modeling Approach*, Addison Wesley (1991).

Nagle, R. K., Saff, E., and Snider, A., *Fundamentals of Differential Equations*, 7th ed., Addison Wesley Longman (2010).

Polking, J., Boggess A., and Arnold, D., *Differential Equations*, 2nd ed., Prentice Hall (2005).

Strogatz, S., *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry and Engineering*, Addison Wesley (1994).