CODEE Journal

Volume 9 Article 9

11-14-2012

Evolution of the Modern ODE Course

Beverly West

Robert Borrelli

Lawrence Moore

Recommended Citation

West, Beverly; Borrelli, Robert; and Moore, Lawrence (2012) "Evolution of the Modern ODE Course," *CODEE Journal*: Vol. 9, Article 9.
Available at: http://scholarship.claremont.edu/codee/vol9/iss1/9

Evolution of the Modern ODE Course

Beverly West
Cornell University

Robert Borrelli
Harvey Mudd College

Lang Moore
Duke University

Keywords: teaching differential equations, differential equations solvers, differential equations textbooks
Manuscript received on March 28, 2012; published on September 14, 2012.

Abstract: The rapid development of technology in the latter part of the twentieth century has revolutionized the teaching of differential equations. In this paper we will try to trace the evolution of this important change. We tried to include the most important efforts in this regard, but we apologize in advance if some efforts have slipped our attention.

1 Introduction

Before the 1980s, ordinary differential equations were rarely taught as a stand-alone course in the mathematics core (the freshman/sophomore years). Calculus and Linear Algebra required the full two years of the mathematics core. At the behest of science and engineering departments differential equations were worked into the core from time-to-time, mostly to present techniques for finding solution formulas, especially for linear equations with constant coefficients. Graphing of solutions was done with pencil and paper and some simple models like falling bodies and population growth were covered. Stand alone differential equations courses were definitely for the upper division, and those courses emphasized theory more than applications.

For example, all Harvey Mudd College (HMC) students were required to take a common mathematics core for the first two years, and there was no separate differential equations course in the core. The sole textbook for the entire math core was *University Mathematics*, whose principal author was the founding Chair of the HMC Math Department. An upper division differential equations course was offered which used one of the excellent textbooks: *Differential Equations* by Edward L. Ince (originally published in 1926, and still around today as a Dover publication), *Theory of Differential Equations* by Earl Coddington and Norman Levinson (originally published in 1955 by McGraw-Hill and still being used today), *Ordinary Differential Equations* by Garrett Birkhoff and Gian-Carlo Rota (first published in 1962 and still available today in paperback), *Differential Equations* by Lester R. Ford (first published by McGraw-Hill in 1933; copies are still available on amazon.com).

But with the emergence and ready availability of computers in the 1970s, things began to change nationally. Numerical analysts began to design algorithms for finding good approximations to initial value problems and coded them to run on various platforms. Many of these algorithms are open source because they were produced with NSF support and were archived in the *Netlib* web site. A great number of these algorithms were coded in FORTRAN. A classic work on the numerical solution of ODEs was written by Larry Shampine (SMU) [61]. Software differential equation solvers began to spring up seemingly everywhere which relieved students of the drudgery of pencil-and-paper graphing. Computer solvers could also be used to experiment with the behavior of solutions when the data in the differential equations changed. Also, textbooks began to change. They started including computer experiments or *labs* which allowed students to play with the differential equations arising in mathematical modeling. Some faculty began to design open-ended labs called *projects* which allowed students to work in teams to create and validate mathematical models of physical and biological situations that involve differential equations.

2 The Early Days

In the 1950s and 1960s computers were clunky affairs that came in several parts and occupied much space. Users communicated with them via punched cards, and the computer printed results on long rolls of paper. Undergraduates only had access to these computers for research purposes and definitely not for instructional purposes. In the 1970s things began to change. Digital Equipment Corporation came out with a time-sharing concept using a central computer processing unit; also, personal computers were beginning to evolve. See the article by Wood and Flint [73] for a review of early ODE solvers. Here are the stories of those beginning days at Cornell University, Harvey Mudd College, and Duke University:

Cornell University (as told by Bev West). In the late 1970s I was teaching Math 213, third semester calculus, for students *not* majoring in mathematics, physics, or engineering. Most of the students were in biology, meteorology, textiles, nutrition, and other diverse disciplines, and Math 213 would probably be their last math course. I considered the semester a success whenever some students would go on to take linear algebra or Math 421, an “applicable mathematics” course centered on differential equations and most often taught by John Hubbard. The syllabus of Math 213 was absolutely crammed, covering multivariable calculus, ordinary differential equations, and an introduction to partial differential equations, all of which I had to make relevant to my varied audience. I covered direction fields (with hand-drawn sketches only) because I thought that they were a good way for students to *see* what was going on with solutions of ODEs. One day, John Hubbard walked into the copy room and saw one of my handouts cranking out on the ditto machine, with those sketches of direction fields. He was so excited he immediately insisted he had to teach me about “fences” and “funnels”. My syllabus had no room for formally teaching those new concepts, but the vocabulary was very handy. And so began my long collaboration with John.

I continued to follow Hubbard and was enchanted by his teaching of Math 421. At some point in the early 1980s he got permission for *his* class to use a computer room in the engineering college. For the first time Cornell students could enter *any* first order differential equation and see the direction field drawn automatically; then they could point/click to choose an initial condition and watch solutions *evolve* on the computer screen. Hubbard's Math 421 students could do all their homework with this assistance. However, as a sign of the times, I was not able to obtain such computer access, even for a single session, for the students in the Arts College course Math 213. (That situation persisted until the introduction of the Macintosh computer in 1983. Although, for some years, this only meant I could do an occasional class demo on my original little Mac; even that small exposure was a huge eye-opener for my students.)

We were greatly (but most pleasantly) surprised when Hubbard gave his first exam after introduction of computer drawing into Math 421. We both had always taught that hand-drawn solutions must *follow* (and *never cross*) the direction field, yet on every exam we'd receive papers with solutions blithely drawn *across* vectors. This time there had been no such instruction, because the direction fields and solutions were drawn by the computer. There would be *no* computer on the exam however, so we expected even more erroneous papers. We were astonished, and delighted, to discover on the contrary that *every* exam solution drawn on a direction field followed the field vectors *exactly*! The obvious lesson was that just *seeing* the solution *evolve*, carefully following a direction field, instilled the concept far better than any verbalization from instructors, and the students didn't even realize that they were learning it!

During this first year with computers in the classroom, Hubbard designed (with work/study students doing the programming) several open-ended teaching tools. There was a function *Analyzer*, the ODE solver *DiffEq*, a *Numerical Methods* program that studied the effect and errors of Euler, midpoint Euler, and Runge-Kutta methods, and a *DiffEq Phase Plane* program for systems of two first-order ODEs. This collection became the first *MacMath* software. Cornell's Academic Computing Division took a shine to our efforts, and in 1985 or '86 loaned us some larger Macs for further development of our software. They also scheduled once-a-week access for Math 213 in the first Mac computer labs Cornell had on campus.

The 1980s were the heyday of complex dynamics and fractals, so we were in the thick of initial research in those subjects as well, and right in the forefront internationally. Our own mathematics department however was slow to understand that we were doing anything but "pretty pictures" (i.e., not *real* mathematics), so it was not until the 1990s that we got our own department computer lab.

In January 1984 my husband and I were in Paris, where I met Hubbard's colleagues, Michèle Artigue and Véronique Gautheron (both at University of Paris-Diderot, Paris 7). They had published a wonderful little book [13] of computer experiments in differential equations, with tiny programs written on cash register tape! Later Véronique translated most of Hubbard and West into a lovely single volume French edition [42], improved by deleting all specific references to *MacMath*, acknowledging that any suitable software could be used for this exploratory sort of course.

In 1985 I got to attend one of the first conferences for using technology in teaching, ICMI, in Strasbourg. There I met Bernard Winkelmann (University of Bielefeld, Germany) and two Englishmen, David Tall (University of Warwick, UK) and Douglas Butler (Oundle School, UK), who were doing similar computer graphics teaching for differential equations. Tall, in particular, had created a wonderful suite of programs, in color, on the BBC personal computer, which at that time had only 7K memory for writing programs! I was particularly enchanted with Tall's 3D version, and we were invited to write a joint paper for the Proceedings of the Conference [69]. Later we were able to invite Tall to Cornell for some talks and a good exchange of ideas and techniques. (Seeing David Tall and John Hubbard playing/singing duets was an unforgettable experience!)

MacMath. Two Danish colleagues, Bodil Branner (Danish Technical University, Copenhagen) and Bjørn Felsager (Haslev Gymnasium, Denmark) each visited us at Cornell several times and contributed greatly to the development of *MacMath* [43, 44] and the two texts [40, 41] that grew out of Math 421: *Differential Equations: A Dynamical Systems Point of View, Parts I and II* (back when "dynamical systems", in mathematics rather than physics, was a very new concept). Bjørn also wrote for Math 213 an entire lab manual with over 100 lab/homework exercises.

Our software adventures were extensive, and complicated, because we did not have department support, and university support only for the Macintosh. During the first summer (1984) after the introduction of the Macintosh computer, Hubbard hired students to write more Mac software, and *MacMath* expanded to include *DiffEq-3D Views*, *1-D Periodic Equations*, *Cascade* (of bifurcations), *2-D Iterations*, *EigenFinder*, *JacobiDraw*, *Fourier*, and *Planets*. Hubbard also hired students to create the same programs for IBM PCs, but had to give up that effort because of IBM's lack of graphics support. In fact, there were a number of PC efforts at graphics beyond the bar graphs that businesses wanted, but because there was zero standardization of these graphics efforts, there was no way to write for them all. Eventually (1988?), parallel programs for IBM PCs were produced, but they never really took off. They seemed to work well only on machines with the same hardware as those on which the programs were written.

Nevertheless, *MacMath* went through several versions. *MacMath 9.0* was published by Springer in 1992, and *MacMath 9.2* [43] followed soon after. *MacMath 10.0* [44] was created by student Ben Hinkle as an even better package, with postscript printing and many more options and capabilities to locate bifurcations as parameters changed. However funding became ever more difficult, operating systems changed rapidly, and programmers graduated and had to be replaced. Although *MacMath 10.0* was never finished and published, that is what Hubbard and West still use for all their own work. It is available online at Hubbard's dynamical systems website [44], but *MacMath* only works on the Mac Classic operating system. (Hubbard and West each retain sufficiently old Macs in order to continue using *MacMath* in teaching, writing, and publishing. No satisfactory substitute has been found.)

At Cornell there were spinoffs of *MacMath*. Douglas Alfors (Cornell University) and West, with support from Cornell's Academic Computing, developed *Analyzer**, a more powerful program for analyzing functions in calculus courses, including such capabilities as parametric equations, polar coordinates, Taylor series, step functions, and so on. *Analyzer** won an EDUCOM/NCRIPTAL award for educational mathematical software, and was later published by Addison Wesley. Richard Sours (Wilkes University) wrote a lab book for *Analyzer**, also published by Addison Wesley.

In 1990 Hubbard and West obtained an NSF grant to extend the calculus reform movement to second year calculus, including both multivariable calculus and differential equations. This enabled us to support a hugely productive year's work from Bjørn Felsager, teaching us a great deal and creating many new materials. Work progressed on *DiffEqSys*, an effort to extend the basic open-ended ODE solver in *MacMath*. Unfortunately our most excellent student programmer for that project disappeared suddenly and unexpectedly because he graduated and got a real job that demanded his immediate acceptance. Moreover, our application to renew the NSF grant was denied. Despite a one-year effort from a non-student full-time programmer, this package was not finished – a big disappointment. Students were our best programmers, and affordable, but they move on and the times change. We no longer have this option.

Harvey Mudd College (as told by Bob Borrelli). After a stint in industry, I arrived at HMC in 1964 as the token applied mathematician. It did not take me long to discover that the department did not regard applied mathematics as "real" mathematics. The two-year mathematics core did not have a differential equations course per se, but some basic differential equations and linear algebra were worked into the core single variable and multivariable calculus courses. So in 1967, a colleague Courtney Coleman (Harvey Mudd College) and I created a year-long junior-level course called *Applied Analysis* which would be required of all HMC students. As there was no text for such a course, we decided to write one. We titled the text, *Mathematical Methods, Models and Applications for Engineers, Mathematicians and Scientists* [20]. The purpose of this in-house text was to give students a glimpse into some important areas of applicable mathematics involving differential equations along with some experience in traditional and contemporary applications. 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). Work on the in-house text extended over a decade, but it was never published. It contained more than 2200 typed pages and was used as a text for 7 different courses at HMC at one time or another. It is now freely available online [20]. One look at it tells why the students affectionately (or not?) called it the "monster"! Some alumni told us later that the "monster" came in handy in helping them prepare for their prelims in grad school.

Thankfully, the computing facilities at HMC were in fairly good shape in the early 1970s. All HMC students had access to DEC's time-sharing computer (the VAX-11/780), and terminals were located in several large labs staffed with student computer assistants. Of course, the engineers seemed to be fond of a simulation program called Continuous Systems Modeling Program (CSMP) that ran on an analog computer. The major software package available on the VAX was International Mathematical and Statistical Library (IMSL), a library of subroutines including ODE solvers and graphers. HMC's Computer Services made these routines simple and convenient for students and faculty to use.

To make our *Applied Analysis* course more appealing to all HMC students regardless of major, we introduced "open-ended" team projects in 1972 as a required element of the course. Work on projects extended over the entire year. Projects began with the construction of a model involving differential equations and ended with an analysis of how well the model fit the system of interest. Naturally, the models chosen were appropriate to the level and nature of the course, and the mathematical analysis of the model was intended to give a substantial practical application of the abstract mathematical ideas covered in the course. The successful completion of a project required that the student team communicate their model and its analysis via a final report. Digging up good projects was a very labor intensive task for the instructor, but it got easier in succeeding years because some of the already completed projects had room to be recycled.

A *Handbook* [14] was designed to help students organize the resources needed to successfully complete a project. The *Handbook* described the project experience in some detail and laid out a calendar for project work as well as a check list for final reports. The *Handbook* contained abstracts for all project reports from 1972, inception of the projects program. Completed project reports were housed in a file located in an office that was accessible to students. As of 1982 there were 196 final reports on file. In the *Handbook* there was also a Guide for Scientific Literature Searching and a description of computer resources at HMC (now completely outdated!).

For successful project work it is crucial to have reliable high-quality software available. In the 1970s, computer algebra systems (CASs) such as *Maple*, *Matlab* and *Mathematica* were in their infancy, and they were too expensive for HMC to afford anyway. So we used HMC students to take excellent public domain solvers and graphing software to produce a package that would run on our VAX and be useful for students involved in project work. We called this package *Mathlib*. The students who produced *Mathlib*, Ned Freed (an Engineering major), Kevin Carosso (a Physics major), Dan Newman and Kristin Hubner (math majors), were the best programmers I have ever seen.

ODEToolkit. *Mathlib* is a general purpose interactive mathematical workbench for research that fills the need for an analysis tool that is simple enough for an inexperienced user but sophisticated enough to realistically cope with the complex problems encountered in scientific research. *Mathlib* is 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. For a differential equations solver engine we used the excellent open source adaptive solver LSODA written by Alan Hindmarsh and Linda Petzold of Lawrence Livermore Labs. (It was written in FORTRAN.) In the late 1980s, a HMC student, Tony Leneis, developed a front end for *Mathlib* to make it easier for users to solve and display solutions of ordinary differential equations. Called *ODEToolkit*, it was used to support project work in differential equations classes until its demise in the late 1990s when DEC stopped supporting VAX machines. So *Mathlib* is mostly of historical interest today. Even so, there is one VAX at HMC on which *Mathlib* resides. The *Mathlib Reference Manual*, the *Mathlib User's Guide* and the *Mathlib Example Set* are all currently stored in the Claremont Colleges Digital Library [33].

So many good project final reports were being produced that we thought it a shame to just stick them 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 gratis to every current HMC student as well as all HMC Trustees. A few hundred copies of each issue were mailed out to alumni and other university math departments for a modest fee. There is also 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*. The *Guide* and all the issues of *Interface* are stored online [5].

Interface was published as a print publication for 27 years, but ceased publication in 2000 (strangely about the time Coleman and I retired). But the good news is that my young colleagues, with the help of the Claremont Colleges Digital Library are setting up a mechanism for digitally archiving all HMC mathematics student research papers and senior theses. There is an endowed fund at HMC to support a student editor, office and other expenses.

In 1981, a differential equations course was inserted into the core and a year later the *Applied Analysis* course was reduced to one semester. In the early 2000s the core was chopped up into half-courses which allowed a half-course in Linear Algebra between two half-courses of differential equations.

Duke University (as told by Lang Moore). Technology was a major force in the calculus renewal movement of the late 1980s and 1990s with lasting impact. In particular, simple numerical methods for solving initial value problems enabled textbooks and teachers to examine important applications of calculus early in the first-year curriculum. This, in turn, brought increased attention to ordinary differential equations as a tool to understand population growth and the spread of disease, as well as more traditional applications in physics and engineering.

Project CALC. My colleague David Smith (Duke University) and I received major funding from the National Science Foundation's calculus initiative of the late 1980s and early 1990s. Our *Project CALC* was a major player in the Calculus Reform Movement, and made one of the most significant uses of notions from differential equations of any of the calculus reform projects. This use was made possible by the existence of software that enabled the easy creation of slope fields and numerical solutions of initial value problems.

The focus of *Project CALC* was, as the name suggests, on projects. Our decision on what calculus topics to cover in our course was determined by what was necessary for the projects. While the Mean Value and Intermediate Value Theorems never made it into the course, the notions of a differential equation and initial value problem were introduced very early. For example, Euler's method is easily understood by beginning calculus students and gives an understanding of what is involved in the numerical solution of an IVP. This method can be combined with a more sophisticated method for a careful representation of the solutions.

In our course we developed the concept of a differential equation, initial value problem, and solutions of a differential equation at the same time as we developed the concepts of differentiation and exponential function. It is easy to show that $2^x, 3^x$, and 4^x must satisfy a relation of the form $\dfrac{d}{dx}n^x = cn^x$, where c is a constant. Then **the** exponential is the function with $c = 1$. Exponential functions are extremely important for applications, particularly negative exponentials which appeared in multiple examples of processes that settled to a steady-state.

Initially we thought that we could support the course with just the student projects. However, we soon realized that we needed to write text material to explain the concepts under investigation, and, after many years, these notes became a textbook that was published by D. C. Heath in 1996 [67]. Soon after publication, Heath was bought by

Houghton Mifflin. Our text was of no interest to Houghton Mifflin, since their flagship calculus text was the one written by Larson. So there was essentially no marketing of our text.

We eventually got the rights back when Houghton Mifflin gave formal notice of their decision not to publish a second edition. In 2010 a second edition of the text was published in a digital/online format by the Mathematical Association of America. Information on that edition and sample chapters may be seen at http://www.calculuscourse.maa.org/. We are currently finishing work on an **iPad** version of the text.

The importance of differential equations is illustrated in the differential calculus portion of our text. After Chapter 1, devoted to pre-calculus topics, Chapter 2 introduces our notion of derivative of a function at a point as the slope of the straight line obtained by zooming in infinitely far at the point. This is easily illustrated using computer graphing programs. Exponential functions then appeared as solutions of the simple growth equation, $\frac{dP}{dt} = kP$. Chapter 3 takes a more detailed look at initial value problems in the context of Newton's Law of Cooling, used to solve a murder mystery. In Chapter 4, while obtaining the remainder of the calculational tools for differential calculus, we use the emphasis on simple differential equations to address topics needed for physics and engineering, well in advance of the more difficult concept of the definite integral. Chapter 5 introduces a more serious look at Euler's Method in the context of the SIR-model of epidemics.

Project CALC began in the 89-90 academic year. We eventually equipped three computer labs with 8 PS2s each (students working in pairs on each machine) connected to a Zenith 386 Model 25 file server with a 150-megabyte hard drive. Software was *Mathcad* Student Edition for numerical calculations and graphing, *Derive* for symbolic calculation and graphing, and *EXP* for writing reports. *Project CALC* received the 1991 EDUCOM Higher Education Software Award, Best Mathematics Curriculum Innovation and a Kaleidoscope Project That Works Award in 1993.

There was similar work in other reform projects, though not many introduced differential equations as early as we did. Keith Stroyan in his reform text, *Calculus: the language of change* [68], introduces models for epidemics at the very beginning of his course. In the Five Colleges Project text [24], differential equations were introduced in Chapter 3. In their reformed course, *Calculus and Mathematica* [28] authored by Jerry Uhl, Horatio Porta (University of Illinois), and Bill Davis (Ohio State), the entire course was delivered on *Mathematica* with considerable emphasis on differential equations.

Quite a number of these innovations in the teaching of calculus now appear in the mainstream texts. For example, in the College Board Advanced Placement syllabi for calculus [17], we have the following topics:

- Geometric interpretation of differential equations via slope fields and the relationship between slope fields and solution curves for differential equations. (AB and BC)

- Numerical solution of differential equations using Euler's method. (BC)

- Solving separable differential equations and using them in modeling (including the study of the equation $y' = ky$ and exponential growth). (AB and BC)

- Solving logistic differential equations and using them in modeling. (BC)

Our experience with technology largely has determined the later directions of our careers into digital libraries and online mathematics journals. In fact, David and I next moved on to projects for additional courses (see later).

3 Going National

In the mid-1970s the market for pure mathematicians dried up and interest in applied mathematics exploded nationally. NSF support for pure mathematics was cut back. Mathematical modeling started finding its way into mathematics courses, and colleagues across the nation got together to further this activity. Below we mention some of these activities and connections:

Modules of Applied Mathematics. Bill Lucas (Cornell University) received an NSF grant in 1976 to invite a group of mathematicians for a three-week stay at Cornell to produce some mathematical modeling modules. Beverly West, Courtney Coleman and Bob Borrelli met while there and worked on a number of modules involving applications of differential equations:

- Setting Up First-Order Differential Equations from Word Problems
- Qualitative Solutions to First-Order Differential Equations
- Combat Models
- Quadratic Population Models: Almost Never Any Cycles
- Biological Cycles and the Fivefold Way
- Hilbert's 16th Problem: How Many Cycles?
- Shaking a Piece of String to Rest

These modules were later published [23]. So when Beverly and Bob met in the late 1980s at a Joint Mathematics Meeting, they were anxious to share how they were bringing computer graphics and applications of ODE's into the classroom in an effective way.

Consortium for Mathematics and its Applications (COMAP). COMAP was founded in the late 1970s by Sol Garfunkel and a small group of colleagues with the mission of improving mathematics education for students of all ages [1]. Its headquarters are in Bedford, MA. COMAP's educational philosophy is centered on mathematical modeling and using mathematical tools to explore real-world problems. Products are produced in various formats: print, video, and multi-media. Among many other things, COMAP publishes *modules* which use differential equations to create mathematical models of real-world phenomena and also two journals: the *UMAP Journal* and *Consortium*, both available electronically or in print. COMAP also sponsors two modeling contests: the international *Mathematical Contest in Modeling (MCM)*, and the *Interdisciplinary Contest in Modeling (ICM)*.

The CODEE Consortium, Phase I. From 1992–1997, Courtney Coleman and Bob Borrelli were co-PIs of an NSF grant to CODEE (*Consortium for ODE Experiments*); the Consortium involved 7 institutions: Harvey Mudd College, Cornell University, Rensselaer Polytechnic Institute, Washington State University, St. Olaf College, West Valley Community College, and later, Stetson University [2]. The goal of CODEE was to share the rapidly growing wealth of computational instructional techniques for teaching ODEs with as many teachers and students of differential equations as possible. This was done in several ways: Over the summers 1992-1995, each of the seven institutions hosted a week-long workshop for about 30 colleagues, to help them become familiar with the experimental approach to teaching differential equations, and its many advantages. CODEE organized talks and panels at various math meetings each year, especially JMM and ICTCM. Concurrently, CODEE published a Newsletter [3] that provided a regular source of ideas, inspiration, and experiments for ODE instructors. Publication of the printed Newsletter stopped when funding ran out in 1997, but all the issues are available on line at the CODEE web site.

ODE Architect. Because computers are vital tools in helping students understand and visualize concepts in differential equations, CODEE also produced a software solver package called *ODE Architect* [9]. The numerical solvers in *ODEA* were custom designed by Larry Shampine to be appropriate for the Windows platform that *ODEA* was coded to run on. Larry's papers [62], [63], [64], and [65] published in the CODEE Newletter give an overview of the algorithms that Larry employed in *ODEA*. CODEE designed *ODEA*, but a small software house called Intellipro, Inc., wrote the code. *ODEA* is a Windows software package that was designed to be easy to use and to provide a highly interactive software environment for constructing and exploring the user's own ODE models of real-world phenomena.

ODE Architect has three components: the *ODE Architect Tool*, the *Multimedia ODE Architect*, and the *ODE Architect Library*. The *ODEA Tool* employs a graphical user interface to enter and edit equations, control solver settings and features, and to create and edit a wide variety of 2D and 3D graphics. The Tool solves systems of up to ten first-order ODEs, which are entered using a simple, natural scripting language. *Multimedia ODE Architect* illustrates the modeling process in detail, with dozens of examples, each with its own theme, supported by highly interactive simulations. (See the Table of Contents below.) Students explore the modeling process via "what-if" scenarios and exercises. Along the way, students are guided to build their own ODEs that model a given situation and then solve them numerically and graphically. The *ODEA Library* has dozens of preprogrammed, editable, and interactive ODE files covering a wide range of topics from mathematics, physics, chemistry, population biology, and epidemiology. Each Library file has explanatory text along with the ODEs and includes illustrative graphs. The Library files are organized into folders with descriptive titles to facilitate browsing.

ODE Architect is currently published by John Wiley & Sons and is available on a CD-ROM along with an accompanying booklet, *ODE Architect Companion* [25]. There is also a User's Guide, written by Michael Moody [55]. For more information on *ODE Architect*, visit `http://www.math.hmc.edu/resources/odes/odearchitect`.

ODE Architect won the distinction of being named by Forbes Magazine as one of the "nine best digital projects on the planet" in December 1998. This was *not* just a list of *math* programs; *ODEA* was the *only* math program so honored. There had been 1,080 entries in 40 categories submitted to the New Media INVISION '98 Awards and *ODEA* was one of only nine given an Award of Excellence. *ODE Architect* also won a gold medal in the category of Higher Education.

ODE Architect Table of Contents

[1] Modeling with the Architect
by Douglas Campbell and Wade Ellis
(West Valley Community College)
- Fishing in Monterey Bay (logistic model)
- Modeling Tutorial in Solver
- Experiment with Solver (adds harvesting)

[2] Introduction to ODEs
by Margie Hale [66] and Michael Branton
(Stetson University)
- ODE's and Their Solutions
(solution formulas and graphs)
- Slope Fields and Golf
(emphasis on graphs)
- The Juggler
(modeling, analytic solutions, graphs)
- The Skydiver
(modeling, graphs; adds air resistance,
then time for the parachute to open)

[3] Cool Applications of First Order ODEs
by Margie Hale (Stetson University)
- Cooling an Egg
(Newton's Law of Cooling)
- Cooling a House
(outside temperature fluctuations)
- Melting a Snowman
(volume vs surface area)

[4] Second Order ODEs
by William Boyce and William Siegmann
(Rensselaer Polytechnic Institute)
- Linear spring
(homogeneous, linear)
- Forced Oscillations
(nonhomogeneous, linear)
- Upended Pendulum (on a cart)

[5] Models of Motion
by Robert Borrelli and Courtney Coleman
(Harvey Mudd College)
- Dunk Tank (target practice)
- Longer to Rise or Fall
(includes Indiana Jones jumping
off a bridge onto a moving train)
- Ski Jumping

[6] Linear Systems
by William Boyce and William Siegmann
(Rensselaer Polytechnic Institute)
- Pizza Shop and Video Store (2-D)
- Parameter Movies (2-D)
- Coupled Springs (4-D)

[7] Nonlinear Systems
by Michael Branton (Stetson University)
- Spider and Fly (linearization)
- Saxophone Reed (vibration)
- Spinning a Tennis Racquet
(stability of steady rotations)

[8] Compartment Models
by Robert Borrelli and Courtney Coleman
(Harvey Mudd College)
- Lake Pollution (1-D)
- Allergy Relief (2-D)
- Lead in the Body (3-D)
- Autocatalator (4-D)

[9] Population Models
by Michael Moody
(Harvey Mudd College)
- Sneezy, Wheezy, and Grumpy
(epidemic SIR models)
- Modeling Competition in Populations
(game)
- Managing Resources
(game)

[10] The Pendulum and Its Friends
by John Hubbard and Beverly West
(Cornell University)
- Linear/Nonlinear Pendulum
- Pumping a Swing
(length varies)
- Geodesics on a Torus
(similar to damped pendulum)

Continued on next page.

[11] Applications of Series Solutions
by Anne Noonburg and Ben Pollina
(University of Hartford)
- Robot and Egg
  (accuracy of series solution)
- Aging Springs
  (introducing Bessel functions)

[12] Chaos and Control
by John Hubbard and Beverly West
(Cornell University)
- Forced Pendulum
  (unpredictability –what causes it?)
- Poincare Sections
  (fixed and periodic points)
- Tangled Basins
  (including Lakes of Wada, Control Game)

[13] Discrete Dynamical Systems
by Thomas LoFaro
(Washington State University)
- Discrete Models
  (linear with fixed points, stability)
- Nonlinear Behavior
  (periodicity, chaos, bifurcation)
- Complex Dynamics
  (Mandelbrot and Julia sets)

[APP] Fundamentals of Numerical Methods
by Larry Shampine
(Southern Methodist University)
- Euler's Method
- Error
- Taylor Series Method; Comparisons
- Runge-Kutta Method; Comparisons

In the summer of 2010, numerous reports were received that *ODE Architect* did not work under the newly-released Windows 7 (shades of *ODEToolkit*!). Jeho Park, the Scientific Computing Specialist at Harvey Mudd College, looked into this matter. He found a couple of facts to explain the reason for this incompatibility and proposed ways to address it:

1. *ODE Architect* (including its installation program) is a 16-bit application. So it does not run on any 64-bit Windows systems regardless of the Windows version (XP, Vista, or Windows 7) because 64-bit Windows does not have a 16-bit emulator. Hence, *ODE Architect* does not run on any 64-bit Windows systems and not just on Windows 7. All versions of 32-bit Windows come with a 16-bit emulator. So if a user has a 32-bit Windows system, there will be no problem running *ODE Architect* on it. Also, Microsoft's "Windows XP mode" on Windows 7 doesn't work for *ODE Architect.* The XP mode is a kind of virtual machine environment provided by Microsoft for Windows 7 as a solution to backward compatibility. In the XP mode, *ODE Architect*'s front-end interface is working, but whenever it runs the solver, it crashes.

2. Jeho Park confirmed that *ODE Architect* runs correctly inside 32-bit Windows virtual machines built with VMware and VirtualBox. Host machines for this test were 64-bit Windows 7 and OS X 10.6.4 (the latest OS's from both companies.) That is, if users want to use 64-bit Windows or Mac, they can obtain VMware player or VirtualBox for free and run XP virtual machine to install *ODE Architect* inside the virtual machine.

Based on these facts, Jeho was encouraged to delve into a cloud computing feasibility test for *ODE Architect.* He tested CloudShare.com service which offers virtual machine environment through the Internet (i.e., a form of cloud computing) and confirmed that the *ODE Architect* tool runs fine inside the virtual machines on CloudShare's cloud computing facility. Once a 32-bit Windows virtual machine (XP in our case) is setup and *ODE Architect* is installed in the XP virtual machine, (invited) users can log in to CloudShare.com and run the XP virtual machine from inside their browsers. Jeho believes this cloud computing approach would solve *ODE Architect*'s incompatibility issue with 64-bit Windows and other platforms such as OSX and Linux. See the paper by Jeho Park [58] for more information.

What remains now is for Wiley (who holds the copyright on *ODE Architect*) to contact CloudShare or similar commercial cloud computing service to get more information on how to set up the administrative details of such a service and offer it as an option for *ODE Architect* users experiencing the incompatibility issue.

Meanwhile, there were other developments beyond direct involvement with CODEE. In 1992, Beverly West was invited to give two high profile plenary talks on the revolution that computer graphics had enabled in the teaching of differential equations and other mathematics courses. The first was for the Grand Opening of the Fields Institute in Ontario, the second was for ICME 7 in Quebec. An entire issue (November 1994) of the College Mathematics Journal was devoted to innovations in the teaching of differential equations [72]. In 1999, MAA published an entire volume, edited by Michael Kallaher, in its Notes series on teaching differential equations with technology [48].

One of the articles in the special issue of the College Math Journal, *A New Look at the Airy Equation with Fences and Funnels*, by John Hubbard, Jeanie McDill, Anne Noonburg (University of Hartford) and Beverly West, was selected in 1995 to be redone with interactive illustrations, by the Centre for Experimental and Constructive Mathematics (CECM) in Vancouver, BC. Their Organics Mathematics conference for that purpose was very exciting, and the results were published in the CMS (Canadian Mathematical Society) Conference Proceedings **20**, American Mathematical Society, Providence, RI. Although the electronic version [45] is outdated, it may be of historical interest to check it out. The introduction to the Conference Proceedings provides an informative context for trying to innovate when you know the resources will not last.

So, by the late 1990s a great number of colleagues started to use these new tools for teaching ODEs. CODEE members, in addition to colleagues who had attended the CODEE workshops, were busy speaking and giving workshops at every meeting and many a colloquium. It was so *easy* to see the advantages to teaching and applying differential equations to real world situations using computer graphics. Students were empowered in their first ODE course with techniques and skills to handle what had previously been seen only in later courses.

Connected Curriculum Project Consortium. With the support of two new NSF grants, a consortium of faculty from Duke University, Cal Poly San Luis Obispo, and Montana State University created the *Connected Curriculum Project* (CCP). These grants supported the development of modular lab activities for courses beyond calculus, including differential equations. David Smith and Lang Moore (both at Duke University) were PIs for the project. Frank Wattenberg, then at Montana State University, was instrumental in convincing the Consortium to distribute the modules by means of a digital library.

These modules were created as interactive texts using specific computer algebra systems, *Mathcad*, *Matlab*, *Mathematica*, and *Maple*, or in some cases, programmable graphing calculators. Each module is a single-topic unit that can be used for a two-hour lab, or for a shorter supervised period with follow-up on the studentÕs own time, or for self-study. The modules were class-tested with students working in two-person teams in a lab environment. Some modules use an application area to stimulate learning of mathematics, and others went straight to the mathematics. All of these materials use at least some of these powerful tools: hypertext links, Java applets, sophisticated graphics, a computer algebra system, realistic scenarios, thought-provoking questions that require written answers, and summary questions enabling students to see the forest as well as the trees.

NSF support for the Consortium ended in 2003. Although many of the computer algebra files are based on old versions of the software (e.g., most of the *Maple* files are based on version 9.5), the library still exists and is used regularly. (There were over 500,000 views of CCP in 2011.) To see a current list of modules for a differential equations course, visit http://www.math.duke.edu/education/ccp/materials/diffeq/index.html.

Interactive Differential Equations (IDE). In 1994 at a conference in Michigan, Beverly West met Hubert Hohn (Massachusetts College of Art) who had made the most amazing interactive illustrations to aid student learning of concepts. This was different from the open-ended software approach where one could type in any equation. Rather, each illustration used very specific examples to get across a concept, especially the relations between different representations. Each "tool" is well-designed to be simple, direct, uncluttered, and adapted to student needs and questions (not just the instructor's idea of student needs and questions). Every tool allows a great deal of exploration, with point/click on a graph to choose an initial condition, sliders to change parameters, and simultaneous model simulations.

John Cantwell (St Louis University), Jeanie McDill (Cal Poly, San Luis Obispo), Steve Strogatz (Cornell University) and Beverly West collaborated with Hohn at Addison Wesley Interactive to create *Interactive Differential Equations (IDE)*, a package of 97 interactive illustrations and a workbook of 30 labs using these tools to get across concepts.

Suddenly it was not only *students* who saw more light – it was true for *faculty* colleagues too. At various lectures, or at big meetings where the publisher had *IDE* running on a computer display, fellow mathematicians in other fields, who had not been following DEs, would exclaim excitedly at *seeing* a solution evolve in a direction field, or seeing the eigenvectors in a phase portrait.

In 1997 Jerry Farlow (University of Maine, Orono) drafted a text on *Differential Equations & Linear Algebra*, using a lot of these graphical ideas [32]. He and Prentice-Hall asked for some collaboration, which Jim Hall (Westminster College), Jeanie McDill (Cal Poly, San Luis Obispo) and Beverly West provided, on the condition that they could specifically link it to *IDE.* Although the CD was not packaged with the book as was intended, *IDE* is available for free online [39]. The text had a successful second edition in 2007.

Addison Wesley has also bundled *IDE* with ODE texts by Nagle/Saff/Snider [57] and Kohler/Johnson [50]. These texts do not refer directly to *IDE*, but at least the software is supposed to be in students' hands.

Washington State University / IDEA. In the 1990's, Tom LoFaro and Kevin Cooper at Washington State University created *IDEA, Internet Differential Equations Activities*, an interdisciplinary effort to provide students and teachers around the world with computer-based activities for differential equations in a wide variety of disciplines. *IDEA* was sponsored by an NSF grant and is available online [52]. This project is still ongoing; recent additions include a fascinating module on Idaho Wolf Hunting Policy, and another on Insurgencies.

To quote from the *IDEA* home page,

> As with every site on the Web, *IDEA* is evolving. *IDEA* contains a database of computer activities illustrating both mathematical concepts and the application of these concepts in a wide variety of disciplines. The aim is to show differential equations where they live, rather than in a purely mathematical setting.
>
> In addition to the exercises, it provides a variety of software for solving and describing differential equations. These packages include *DynaSys*, a package for Microsoft operating system that can be used in a very flexible way to create displays of solutions of differential equations; Java software that may be used over the Web to develop your own activities, or to work on the ones presented here; and Flash components that again may be used to develop differential equations activities.

The *IDEA* Index of Projects, October 2011 is as follows:

- Basic Ecological Models
- Idaho Wolf Hunting Policy
- Bungee Jumping
- Hydroplane Racing
- Diffusion of Toxics in a Landfill
- Intravenous Drug Administration
- El Nino
- Insurgencies
- Irrigation and Conservation
- Mathematical Models of Neurons
- Chemical Kinetics Fundamentals
- Oscillating Chemical Reactions
- Temperature Programmed Description (TPD)
- Error Analysis
- Quantum Mechanical DE Solved
- Rangeland Ecosystems
- Salmon Migration
- Corporate Fishing
- Small Mammal Dispersion
- Discrete Dynamics on the Real Line and in the Complex Plane(by C.J. Keller)

Boston University ODE Project. From 1993 to 1996, Paul Blanchard, Bob Devaney, and Glen Hall received NSF funding for a large-scale revision of the traditional sophomore-level ordinary differential equations course. They wrote a textbook for a course that emphasizes qualitative theory throughout and has a distinct dynamical systems orientation [15]. Students are expected to use the computer frequently to analyze solutions, and they are exposed to an earlier and more detailed treatment of nonlinear systems. In 1996 and 1997, Blanchard and Devaney received a second NSF grant that funded four four-day faculty enhancement workshops. They were held at Boston University, Oberlin College, Davidson College and DePaul University.

Since 1998, four editions of the textbook have been published, and Blanchard and/or Devaney have given more than 20 workshops devoted to the topic of teaching differential equations with a dynamical systems perspective.

In 2001 and 2005, they worked with Hubert Hohn on a modified version of his *IDE* programs. These modified programs are bundled together in a package that is entitled *DETools*, and the textbook contains exercises that require the use of these tools.

4 Going International

Colleagues from around the world have also been experimenting with the use of technology in the teaching of differential equations as well as designing software packages which contain ODE solvers. Their efforts are not as well publicized as our efforts in the USA. One reason is that many instructors abroad do not use textbooks or publish web pages describing their work. Most of the pioneers already described under "Early Days" are still very actively writing and presenting, and many have expanded their expertise to a broader platform of mathematical learning in general. We list below some additional efforts, specific to ODEs, of which we are aware:

Lebanese American University (LAU) / Beirut. Samer Habre attended the Cornell NSF CODEE Workshop in 1992, before returning to Lebanon from his graduate studies in the U.S. He has become a most valued colleague, returning on sabbaticals (to Cornell 1998-1999, to Cal Poly San Luis Obispo for a semester 2006) and for conferences once a year. During his sabbatical visits Samer conducted extensive research in both mathematics education and dynamical systems, presented at conferences (JMM, ICTCM), made numerous valuable contributions to the text *Differential Equations & Linear Algebra* by Farlow, et. al. [32] and published quite a few papers. Two of his papers, particularly relevant to differential equations, are [36] and [38].

In Beirut, Samer organized two International Conferences on Trends in Mathematics Education (ICTME, 2000; ICTME-2, 2003) that brought together experts from many western countries, with a particular goal of making these discussions available to their Middle East colleagues who are often out of touch with recent developments. As can be seen from their published Proceedings [4], these conferences have been very successful in providing dialog and dissemination throughout the Middle East. Currently (2011) Samer is editing for IGI Global, an international publishing company specializing in scholarly research publications and information science technology, a volume entitled "*Dynamical Mathematical Software and Visualization in the Learning of Mathematics*" [37].

University of South Australia (UniSA) / Adelaide. UniSA has an ODE course that uses interactive modules. That course meets 4 times per week for lectures and one hour per week in a laboratory environment. David Panton and his graduate student Kuva Jacobs describe their success with this approach in a paper published in the CODEE Journal entitled "Learning About ODEs Using Interactive Modules" [47]. This paper discusses the development and implementation of a set of online teaching and learning modules for the modeling and solution of simple linear ordinary differential equations (ODEs). The paper describes how constructivist principles are used in the development of these modules and how built-in assessment has been used to enhance student learning. A discussion of the module structure is provided with particular emphasis on the concept of the module walk-through. Modules developed for the classical ODE problems of the simple pendulum and the mass on a spring are used to illustrate these constructs. The paper concludes with a discussion of the effectiveness of these modules using data obtained from student questionnaires, focus groups and lab session evaluations.

University of Tuebingen / Germany. The Systems Biology Markup Language (SBML) is the standard package for the exchange of biochemical network models. For more about SBML visit http://www.en.wikipedia.org/wiki/SBML. CODEE's *ODEToolkit* is the solver engine which is used in SBML. Andres Draeger (Center for Bioinformatics, Tuebingen) and some international colleagues designed a simulator which makes SMBL easy to use and also convenient to integrate into third-party programs.

Technische Universiteit Delft / Holland. The TU Delft has long offered an ODE course that treats applications from a modeling perspective.

5 ODE textbooks influenced by modern technology

As technology continued to improve and become more convenient to use, textbooks began to change as well. In fact, one can say without fear of contradiction that all the textbooks for introductory ODEs now include mathematical modeling and "labs", and some texts even come with ODE solver packages. Listed below are a few of these textbooks:

John Hubbard and **Beverly West** (both at Cornell) authored two volumes of a series titled *Differential Equations: A Dynamical Systems Approach* (Springer Verlag 1991, 1995), produced and coordinated with *MacMath*. Hubbard still hopes to publish the intended Volume III, but he has several other book projects on the front burner, from his multivariate calculus text to current work with Bill Thurston; he also continues advanced research in dynamical systems. Hubbard's specialty is in the more advanced "second" course in DEs, while West has focused on getting the graphic technology into DE classrooms for undergraduates receiving their first exposure to differential equations. The text *Differential Equations & Linear Algebra* with Farlow, et. al. [32] grew out of that effort when Prentice Hall asked for coauthors, and agreed to link the text directly to *IDE*.

Robert Borrelli and **Courtney Coleman** (both at Harvey Mudd College) authored several books: *Differential Equations, A Modeling Approach* (Prentice Hall, 1987) [18] and *Differential Equations, A Modeling Perspective*, 2nd edition (Wiley, 2004) [19], as well as a workbook with **William Boyce** (RPI), *Differential Equations Laboratory Workbook: A Collection of Experiments, Explorations and Modeling Projects for the Computer* [21]. The main focus of the Workbook is on computer experiments that support and amplify the topics usually found in introductory ODE texts. The experiments do not usually require special features of any particular text or computer platform. Each experiment is a combination of pencil-and-paper and computer work; the work may be straightforward and explicit, but the approach is often open-ended and exploratory. Some experiments are designed for student teams. There are three appendices: Appendix A contains some features to aid students in writing team reports; Appendix B gives some background on mathematical modeling and describes some models that appear in the experiments; and Appendix C is an Atlas of graphs that are frequently referred to in the experiments. With Wiley's permission, the *Workbook* currently appears on the CODEE web site. To see it, visit http://www.codee.org/library/projects/differential-equations-laboratory-workbook-1. All the graphs in these texts and the *Lab Workbook* were produced with the original solver *ODEToolkit* developed at Harvey Mudd College in the early 1980s.

Bill Boyce (RPI) became a real hero – as author of the best-selling traditional DE text, he embraced the ODE technology revolution and went forward with it in subsequent editions. He included *ODE Architect* with the classic text with **Richard DiPrima**, *Elementary Differential Equations with Boundary Value Problems*, 9th edition (Wiley, 2009), and, with **James Brannan** (Clemson University), the text *Differential Equations: An Introduction to Modern Methods and Applications*, 2^{nd} ed. (Wiley, 2011).

Mike Moody, who moved from Washington State to Harvey Mudd to Olin College during and after the CODEE grant, was one of the major movers and contributors on many levels. He gave ingenious and inspiring presentations, authored a lab book, and wrote single-handedly the detailed manual, *A User's Guide to the ODE Architect Software Package*. To see the User's Guide and more on *ODE Architect* visit the web site http://www.math.hmc.edu/resources/odes/odearchitect.

Kent Nagle (University of South Florida) attended the 1992 CODEE workshop at Cornell and incorporated the graphics revolution in his newer "classic" Addison Wesley text with **Edward Saff** (University of South Florida). Later, Addison Wesley (and then Pearson) bundled the solver *IDE* with this text, but Nagle has died, and there has been no mention of *IDE* within the text. **Arthur Snider** (University of South Florida) joined Nagle and Saff for later editions of *Fundamentals of Differential Equations* which is currently in its 8^{th} edition [57].

John Polking (Rice) also attended the 1992 CODEE workshop at Cornell and subsequently developed a front end to Matlab that is an open-ended solver similar to *MacMath*. Polking's *Dfield* and *Pplane* are now Java applets available for free [60]. Polking authored a book *Ordinary Differential Equations using Matlab*, 3^{rd} Ed (Prentice-Hall 2004) and, with **Dave Arnold** (College of the Redwoods) and **Al Boggess** (Texas A& M), authored the text *Differential Equations with Boundary Value Problems*, 2^{nd} Ed [59]. Dave Arnold attended the 1992 CODEE workshop at HMC.

Tony Danby (North Carolina State University) was a mathematician, with a specialty of celestial mechanics, who completely independently wrote a marvelous collection of *Computing Applications to Differential Equations: Modeling in the Physical and Social Sciences* [26]. After some encouragement, Tony published a wonderful update *Computer Modeling: From Sports to Spaceflight: From Order to Chaos* that includes a CD-ROM, with IBM-PC software [27]. Danby has since died (2009), so this is the end of that particular road, but it was a wonderful thing to see him doing all this on his own. His books have been real favorites for sparking student projects; Tony had a knack for presenting applications and explaining essentials, but offering various open-ended opportunities for exploration.

Paul Blanchard, **Robert Devaney** and **Glen Hall**'s (all at Boston University) textbook *Differential Equations* is currently in its 4^{th} edition [15]. In addition to a qualitative, numeric and analytic approach this text also stresses visualizing solutions geometrically, investigating the behavior of solutions, and predicting the behavior of solutions as they apply to models. For animations and other materials useful for a modern ODE course, visit their web site http://www.math.bu.edu/odes/index.html. See Blanchard's CMJ article *Teaching Differential Equations with a Dynamical Systems Viewpoint* for more on the Boston University ODE Project [16].

The textbook *Differential Equations and Linear Algebra* by **Jerry Farlow** (University of Maine), **James Hall** (Westminster College), **Jean Marie McDill** (CalPoly San Luis Obispo), and **Beverly West** (Cornell) is in its improved 2^{nd} edition [32].

C. H. Edwards and **David Penney** (University of Georgia, now retired) produced a textbook in which computing and modeling is stressed throughout [30]. The projects are technology neutral but with illustrative references to *Mathematica*, *Maple* and *Matlab*. There is a *Computing Projects Manual* that accompanies the text. They also published another text combining DEs and Linear Algebra [31].

Dennis Zill, with the assistance of **Warren Wright**, (both of Loyola Marymount University) produced the textbook *Differential Equations with Computer Lab Experiments* accompanied by a *Manual for Differential Equations with Computer Lab Experiments*. The second edition of these books was published by Brooks/Cole in 1998. Zill has several other books which complement these texts by treating applications and solution technique. See [76], [75].

Martha Abell and **James Braselton** (both of Georgia Southern University) authored the text *Modern Differential Equations* which emphasizes applications and technology as well as theory. It is currently in its 2^{nd} edition and was published by Brooks/Cole in 2000. Although this text is technology neutral, the authors have another text which emphasizes the use of *Mathematica* [12].

The delightfully written textbook *Differential Equations and Their Applications* by **Martin Braun** (Queens College, CUNY) was one of, if not the, earliest differential equations textbooks to successfully blend in mathematical modeling in a significant way. It is currently in its 4^{th} edition [22]. The 1^{st} edition appeared in 1976!

Another delightfully written textbook is *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry and Engineering* by **Steven Strogatz** (Cornell). The applications are entertaining as well as enlightening (Westview Press, 2001).

The textbook *Differential Equations: A Modeling Approach* by **Frank Giordano** (USMA West Point) and **Maurice Weir** (US Naval Postgraduate School) is cleverly written at a level that makes the modeling process approachable by a wide audience [34], [35]. In the 1980s and 1990s, the USMA was a very active place for applied mathematics. For many years Frank has directed the COMAP-sponsored Mathematical Contest in Modeling (MCM). The concept of Interdisciplinary Lively Application Projects (ILAPs) was invented by Frank Giordano and colleagues at the USMA. See [46] for a paper on ILAPs by **Michael Huber** (Muhlenberg College).

David Lomen and **David Lovelock** take a very applied approach to learning differential equations which should appeal very much to science and engineering students. Its title: *Differential Equations: Graphics, Models, Data* [53].

E. J. Kostelich and **D. Armbruster** produced the text *Introductory Differential Equations from Linearity to Chaos*. It was one of the earliest ODE texts from the dynamical systems point of view [51].

One of the first differential equations texts for a specific ODE solver was *Differential Equations: Modeling with Matlab* written by **Paul Davis** [29].

C. H. Taubes authored the text *Modeling Differential Equations in Biology* which was one of the earliest texts to emphasize biological models [70].

6 More Recent Activities

As developing technology continues to have an impact on how knowledge is delivered, several recent projects have built on the quantitative, geometric, analytic and modeling approaches introduced earlier with the use of computers. Below are several recent projects:

MIT Mathlets[6]. Haynes Miller (MIT), using an evolutionary variation of a syllabus created originally by Arthur Mattuck (MIT), designed a DE course (18.03) with a carefully developed set of dynamic computer applets called *Mathlets.* (Sometimes the *Mathlets* are called "manipulatives", because of the kinesthetic element they impart to learning, as in Maria Montessori's manipulative approach to elementary mathematics education.) Development of the applets was supported by a grant from The d'Arbeloff Fund. To learn more about the course visit http://www.ocw.mit.edu/courses/mathematics/18-03-differential-equations-spring-2010/.

The general aesthetic and indeed the details of the early applets in this collection grew out of the *IDE* applets by Hubert Hohn, who collaborated for several years with the MIT group in developing many more. Construction of new applets, moving into both calculus and into engineering, continues today. To see an ongoing list of *Mathlets* visit http://mathlets.org.

A nice research study [54] by Haynes Miller and Deborah Upton (Stonehill College) documents the value of these interactive Mathlets in facilitating student learning of key concepts in differential equations. Miller notes that the applets are much more effective as homework than they are as classroom demos, but that actually, the two reinforce each other. Something of a sequel to the Miller-Upton paper has been accepted for publication in the upcoming IGI Global volume *Dynamical mathematical software and visualization in the learning of mathematics.*

In 2000, MIT created the OpenCourseWare (OCW) concept, and the OCW web site was launched in 2002 [7]. OCW currently offers 2000 courses by MIT faculty, including two differential equations courses at the undergraduate level. Miller's course, 18.03, incorporates videotapes of Mattuck's lectures (taped by OCW some years ago), which continue to serve as excellent supplementary material for the class and are consequently incorporated into the OCW publication of the course.[1]

CODEE Consortium (Phase II). In 2007, the NSF funded the *Community of ODE Educators (CODEE), Phase II*, to develop a web site that would be useful for the teaching and learning of ODE's, primarily by encouraging broader use of modeling projects and computer experiments. It is headquartered at HMC with Darryl Yong as PI and Robert Borrelli and Ami Radunskaya (Pomona College) as co-PIs. The project is funded over a five-year period and involves colleagues from a dozen other institutions. Another goal

[1] The OpenCourseWare Consortium was incorporated as an independent non-profit organization in 2008, with support from the Andrew W. Mellon Foundation and also, initially, from the William and Flora Hewlett Foundation. MIT OCW still plays the dominant role in the Consortium, which now is a community of over 250 universities and associated organizations worldwide, committed to openly sharing courseware and its impact on global education. The mission of OpenCourseWare is to advance formal and informal learning through worldwide sharing and use of free, open, high-quality educational materials organized as courses. Collectively, the OCW Consortium members have published materials from more than 13,000 courses in 20 languages, available through the Consortium's web site (http://www.ocw.mit.edu).

of this web site is the creation of an online community of ODE instructors, a site where they can find, share, and discuss resources for teaching ordinary differential equations. The CODEE web site `http://www.codee.org` contains articles, modules and projects, reviews and descriptions of useful ODE software solvers.

Still another part of this project is to develop a Java-based ODE solver to take the place of the ODE solver in *Mathlib* which was command-line-based and ran on VAX machines under VMS. This new solver, also called *ODEToolkit*, 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 functions well even on older equipment. Instead of a command-line interface, the new *ODEToolkit* 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; *ODEToolkit*'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 dynamically 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 on line.

ODEToolkit 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. *ODEToolkit's* code is modular in design and hence will allow for future development. The current version of *ODEToolkit* is always available on a server at Harvey Mudd College and can be accessed at the CODEE web site `http://www.codee.org`. As of this writing (June, 2012), the current version of *ODEToolkit* is Version 1.3.

OdeFactory. Jerry Tutsch, recently retired from Computer Science at University of Wisconsin, has designed an ODE solver package for use by instructors and students [71]. It is free, easy to use and runs on any computer that supports Java. ODEs can be created, explored, documented and saved in galleries which can be shared via the web. Graphics and animation features are supported. `http://www.OdeFactory.com` takes you to the user guide which explains how to obtain and use *OdeFactory*. The user guide also contains links to demo videos for instructors and students. Both the program and the user guide contain many links to free web-based ODE learning resources. This is Jerry's retirement project, and he is anxious for feedback. He has already inserted some of the features (e.g., separatrices, locating equilibria, etc) that ODE instructors have requested. Furthermore, *OdeFactory* includes a direct link to Wolfram|Alpha (discussed below), which gives an analytic solution (when possible) and other insights into whatever ODE is being studied.

MathDL / NSDL. At the 2000 Joint Mathematics Meetings in Washington, the National Science Foundation announced the *National Science Digital Library Program* (NSDL) and issued a solicitation for proposals. [2]

[2] Actually, the original name for NSDL was the National SMETE Digital Library, where "SMETE" stood for Science, Mathematics, Engineering, Technology, and Education. After a few years, "SMETE" was replaced

Jerry Porter (University of Pennsylvania) was interested in creating the *Journal of Online Mathematics and its Application* (JOMA), proposed in the University of Pennsylvania's *Math Across the Curriculum Project* and saw a collection grant within the NSDL Program as a way to do this. Jerry, Gene Klotz (Swarthmore College and founder of an online math site *The Math Forum*), David Smith and Lang Moore (the last two both at Duke) met at the meeting and decided to approach the Mathematical Association of America (MAA) about making a proposal to the new program.

Don Albers (MAA Director of Publications) was contacted and agreed that this looked like a good opening for the Association. With the assistance of Albers and Tina Straley, then the new Executive Director of the MAA, Lang Moore wrote the proposal which was submitted and subsequently funded in the summer of 2000. This led to the creation of MathDL, which debuted in January of 2001 with the help of programming and hosting provided by Math Forum. JOMA became the online journal for the Library. In 2003, MathDL was moved in-house at the MAA.

MAA Pathways Project. NSF ended the NSDL Program in 2011, but the NSDL lives on as an entity with headquarters in Boulder, Colorado, at UCAR (University Corporation for Atmospheric Research). In the last six years or so of NSDL's existence, NSF focused on fairly large *pathway* grants. The pathways are either disciplinary-based or are focused on particular groups of users. There are currently 18 pathways within NSDL. With additional NSF support, MathDL became in 2008 the MAA's Pathway Project in NSDL. MathDL's *Partners* are all mathematics sites. Most of them have resources that can be searched via the Partner Search facility located on the MathDL home page. CODEE is the only one of MathDL's partners which is devoted exclusively to differential equations. Other MathDL partners have some differential equations resources [8]. There is more information on the MathDL Partners program at http://www.mathdl.maa.org/mathDL/?pa=content&sa=viewDocument&nodeId=2004. Lang Moore has been Executive Editor of MathDL since it was founded.

In 2007, JOMA was merged with *Digital Classroom Resources* and *Convergence*, an online publication devoted to the use of the history of mathematics in the teaching of math, to form the single online publication *Loci.* There are 18 articles in *Loci* classified as being about ordinary differential equations. Most of these date back to JOMA and include several versions of projects originating in the CCP collection. An additional two articles are so classified in *Loci: Convergence.*

In 2010, MathDL began a new collection, Course Communities, focused on particular courses. In the summer of 2011, a beginning course in differential equations was added as a new Course Community to the collection. To see it visit http://www.mathdl.maa.org. Currently there are 78 resources recommended for use in the differential equations Course Community in the following categories:

- Analytic Methods
- Numeric Methods
- Collection and Archives

by "STEM". Finally, the name of the NSF Program was changed to "National STEM Distributed Learning". Both the program and the library are still referred to by the initials NSDL.

- Modeling
- Graphics Methods
- Series
- Laplace Transforms

Wolfram Demonstrations Project. The *Wolfram Demonstrations Project* [11] was inaugurated in 2007 as a means of creating interactive visualizations. Each *Demonstration* is a small program created with a standard copy of *Mathematica*. Anyone with a good idea and a copy of *Mathematica* can create a *Demonstration* and afterwards *Mathematica* converts the filled–in template into an interactive Computable Document Format (CDF) object. *Mathematica* is not necessary to play a *Demonstration*; the free Wolfram CDF Player is all the user needs. However, *Mathematica* allows the user to modify any *Demonstration's* source code for their own purposes. To access Wolfram *Demonstrations* go to the web site http://demonstrations.wolfram.com. To download a *Demonstration* or its source code the user only needs to click the link in the sidebar. All *Demonstrations* run freely on any standard Windows, Mac or Linux computer. Mac or Windows users can interact with a *Demonstration* right in their browsers. There are currently 231 *Demonstrations* involving differential equations, and the list continues to grow. For more on the *Wolfram Demonstrations Project* go to the web site http://demonstrations.wolfram.com/about.html.

Wolfram|Alpha (W|A). In 2009, Wolfram Research introduced *Wolfram|Alpha* [10]. This tool uses *Mathematica* as a computing engine and will answer a wide range of questions. For mathematics, it will calculate derivatives, integrals, solutions of differential equations, and much more. It is freely available at http://www.wolframalpha.com and an app for an iPhone or iPad is available for a small price. ($1.99 as of October 2011). Not only does Wolfram|Alpha produce answers to questions, but for mathematics, the program will give a step-by-step description of the solution process. See the article, *Life After Wolfram|Alpha: What You (and Your Students) Need to Know*, by Gizem Karaali (Pomona College) and Bruce Yoshiwara (Los Angeles Pierce College) [49]. However, *Wolfram|Alpha* tends to also give one of the most *un*informative graphs (e.g., of a *single* solution over some interval chosen automatically by the computer) we have ever seen. In some cases (W|A) gives the initial condition used for that solution; other times it does not. For us, this has always been a problem with CAS's in general – they just zap out finished pictures, so one misses entirely the concept of creating something oneself. Dedicated ODE software has been far better for understanding what is going on.

7 Looking Forward

So where are we now? As we commented above, in its current state, *Wolfram|Alpha* does not always list very informative steps or appropriate graphical representations. As of October, 2011, it does not produce slope fields or direction fields. But that will change. We are just beginning to see voice communication between users and their hand-held devices.

Clearly, current students will have extensive tools at their finger tips throughout their professional lives. This means that it is increasingly important to teach students how to use these tools. Indeed, this may be more important than particular mathematical topics. For more discussion of this idea, see "Changing Technology Implies Changing Pedagogy" [56] by Lawrence Moore and David Smith, in MAA notes #69.

With the accelerating pace of technological innovation we have come a long way in the past 40 years in bringing differential equations into the mathematics core and in making it a lot more relevant to all science, engineering and mathematics students. Textbooks with many new features are now available (even online!) and many software solvers can be delivered gratis over the Internet. Classrooms outfitted with WiFi allow students to bring their laptops to class to download solvers and labs, making things as convenient as possible for the instructor. So where do we go from here? There are still some wrinkles to be ironed out.

There are so many solvers available either in the marketplace or online that it is sometimes difficult for inexperienced instructors to decide which will best fit his or her class. Software solvers have their own peculiarities depending on the algorithms employed, the hardware platform, and so on. All solvers allow some adjustment to the solver parameters. Some solvers are menu-driven and interact with the user through elaborate displays (but not so much anymore), 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. Somehow, students need to be made aware that solvers cannot be run mindlessly with the results accepted without question. Visual displays of solutions can be misleading or difficult to interpret unless the basic theory is in hand. It is ironic that the introduction of computer solvers into an ODE course often leads to a heightened interest in the theory rather than less. So this must be kept in mind as the pace of technological innovations accelerates.

Ongoing ODE software development. Many of us have had the experience of designing ODE software tools on a particular platform with a specific operating system only to have the tools inoperable because the operating system changes or the platform goes out of existence. This is what happened to the original *ODEToolkit*, *ODE Architect* and *MacMath*. We have finally learned our lesson and currently use Java code for our tools. Who knows what other alternatives the future has in store? One possibility we are looking into is to run the original software on a virtual machine to make access to it via cloud computing (which ODEA is considering doing). Unfortunately, sophisticated open-source solvers are not written in Java, so less-than-optimal algorithms have to be coded from scratch just as the latest versions of *ODEToolkit* and *ODEFactory* have done. However, *Pplane* and *Dfield*, which are written in Java, call solvers from Matlab. For a review of *Pplane* and *Dfield* see the paper by Alan Zehnder [74]. Visit the web site http://www.codee.org/software/ for more on solvers that can be used in the classroom.

There is hope that this problem can be addressed to some extent by a recent development: Harvey Mudd College and a group of some 18 educational institutions nationwide have created a consortium called LabSTOR which uses Virtual Computing Lab (VCL) for remote access to computing environments that include applications found in campus computing labs. HMC has successfully experimented with making *ODE Architect* available over a cloud computing facility such as CloudShare and found that the 16-bit *ODEA* Graphical User Interface (GUI) and integrated solvers can be accessed no matter what Microsoft decides to do with its new releases of Windows.

Unfortunately the multimedia feature in *ODE Architect* does not seem to work using this approach. See the article by Jeho Park [58] on the future of *ODE Architect*.

The advantages of using the cloud facility LabSTOR include reduced costs and more bargaining power when negotiating license agreements. Harvey Mudd College plans to test *ODEA* in this environment in late 2011. Visit `http://labstor.blogspot.com/2011/11/labstor-moves-from-proof-of-concept-to.html` for more information. So there is some hope that this problem can be solved.

ODE Faculty. Differential equations courses taught by ODE specialists seem to be more or less saturated with technology, but students are taught by many other instructors. Work now needs to be done on convincing *non*-specialists to use software tools in their ODE course. They can make or break the future for students studying ODEs.

Too many younger colleagues have evidently had too much paper-publishing tenure pressure to take up new ways of teaching ODEs. Their big cry seems to be they have no time to fit computer graphics into a ODE course outside their specialty. Perhaps a campaign can be mounted to write and speak at future meetings on how graphics tools and ODE solvers can save time in the syllabus. Jeanie McDill wrote a short piece in the 2^{nd} edition of Farlow et al on exactly that point. See "Notes on the uses of *IDE*" which is included in the Preface ([32], pp xv-xvi) and is available online at `http://www.courses.citcornell.edu/bhw2/pdfs/2$^{nd}$Edition.pdf`.

In recent years at the JMM there has been no specific session for short presentations on teaching or modeling with differential equations. Some of us are retired, with less energy and less chance for interaction with colleagues or students, and some former leaders are gone altogether. The good news is that the 2012 JMM offered three Minicourses and one contributed paper session all involving the teaching of ODEs, but the Minicourses come with a fee and a larger time commitment. Without more sessions at national and regional meetings we will be missing the chance to connect with many nonspecialists. We encourage our younger colleagues to make more of an effort to close this gap.

Institutions using software in ODE courses. A search on the internet will turn up many faculty and institutions who are experimenting in one way or another with the design of software tools for introducing activities such as labs into ODE courses. Obviously, we know well what Harvey Mudd College, Cornell University and Duke University are doing in this regard (see above), but a cursory internet search turned up the following institutions where faculty are also using software tools and experimentation in their differential equations courses:

Agnes Scott College (GA)
Azusa Pacific University (CA)
Benedictine University (MA)
Boston University (MA)
Brown (RI)
Bryn Mawr College (PA)
Cal Poly SLO & Pomona (CA)
Clarion University (PA)
Clemson University (SC)
College of the Redwoods (CA)
Eastern Illinois University (IL)
Georgia Southern University (GA)
Gettysburg College (PA)
Gustavus Adolphus College (MN)
James Madison University (VA)
Johnson County Comm. Coll. (KS)
Keele University (England)
Lebanese American University (LB)
Loyola Marymount University (CA)
Macalester College (MN)
MIT (MA)
Monmouth College (IL)
Montana State University (MT)
Mount Holyoke (MA)

(continued on next page)

Muhlenberg College (PA)
Naval Postgraduate School (CA)
North Carolina State (NC)
Northwestern (IL)
Quest University (Canada)
Raritan Valley Comm. Coll. (NJ)
Rice (TX)
RPI (NY)
Salisbury University (MD)
Salt Lake Comm. Coll. (UT)
San Joaquin Delta Comm. Coll. (CA)
Technische Universiteit Delft (Holland)
Texas A & M (TX)
University of Arizona (AZ)
University of Colorado, Boulder (CO)
University of Maine (ME)
University of Miami (FL)
University of Michigan (MI)
University of North Carolina (NC)
University of South Carolina (SC)
University of South Florida (FL)
University of South Australia
University of Tuebingen (Germany)
Wartburg College (IA)

Communication role of CODEE/MathDL. There are many more institutions using software in ODE courses that we could list, but the pattern is clear: institutions at all levels are involved, and not just here at home but also world-wide. What is not clear, however, is whether they are communicating very well with one another. The hope is that communication among faculty engaged in these activities will improve as the CODEE and MathDL web sites continue to ramp up. It would also help if more sessions on differential equations were held at regional and national meetings.

8 Appendix I: Our Timeline

	Harvey Mudd	Cornell	Duke	Other
1960s	Borrelli & Coleman: modeling; software	Hubbard: fences & funnels theory, with French colleagues	Moore & Smith	
1970s	“monster” text *Applied Analysis:* team projects with *Handbook* Mathlib (on VAX) *Interface*	Hubbard: upper division teaching West: teaching ODE first course to nonmajors		MAA workshop for Modules in Applied Math COMAP CAS: *Matlab*
1980s	B&C text: *DEs, A Modeling Approach* *ODE Toolkit* (Mathlib front end using LSODA)	H&W: *MacMath*, and 2 vol. text *DEs: A Dynamical Systems Approach* European partners	*Project CALC* using *MathCad Derive,* and *EXP*	ICMI Conference, Strasbourg CAS: *MathCad, Mathematica, Maple, Derive*
1990s	CODEE, Phase I (NSF) (1992-97) incl. 7 week-long faculty workshops *ODE Architect* (*ODEA*)	(NSF) CODEE Workshop *IDE* 2^{nd} yr calc grant (1990-92)	M&S: text *Calculus Models & Application* Connected Curriculum Project (CCP)	Opening of Fields Institute, Ontario; ICME-7, Quebec; Organic Math Conf., Vancouver *Dfield, Pplane* *IDEA* NSDL (NSF) CMJ special ODE issue, 1994; MAA Notes #50 Java
2000s	B& C text: *DEs: A Modeling Perspective* CODEE, Phase II (NSF) (2007-12)	Farlow, Hall, McDill and West text: *Differential Equations & Linear Algebra,* linked with *IDE*	MathDL (MAA) MathDL Partners MAA Pathways Project within NSDL (NSF) JOMA, *Loci*	MIT: *Open CourseWare* *Wolfram|Alpha* CAS:*TI-Nspire* ICTME Confs. Beirut
2010s	online *Interface,* new (Java) *ODEToolkit* LabSTOR, with CloudShare for *ODEA*	H&W: continued writing and development	MathDL Course Communities	Tutsch: *ODEFactory* 2012 JMM-3 ODE minicourses

9 Appendix II: Glossary of Acronyms

page		
7	CAS	Computer Algebra System
14,23	CCP	Connected Curriculum Project
14	CECM	Centre for Experimental and Constructive Mathematics
14,19	CMJ	College Math Journal
14	CMS	Canadian Mathematical Society
10	CODEE, I	Consortium for ODE Experiments
21	CODEE, II	Community of ODE Educators
		http://www.codee.org
10	COMAP	Consortium for Mathematics and its Applications
6	CSMP	Continuous Systems Modeling Program
20	CUNY	City University of New York
6	DEC	Digital Equipment Corporation
25	GUI	graphical user interface
2	HMC	Harvey Mudd College
10	ICM	Interdisciplinary Contest in Modeling
14	ICME	International Conference on Mathematics Education
4	ICMI	International Committee on Mathematics Instruction
10, 17	ICTCM	International Conference on Technology in Collegiate Mathematics
17	ICTME	International Conference on Trends in Mathematics Education
15	IDE	Interactive Differential Equations
		http://www.aw-bc/ide
16	IDEA	Internet Differential Equations Activities
		http://www.idea.wsu.edu
20	ILAP	Interdisciplinary Lively Applications
6	IMSL	International Mathematical and Statistical Laboratory
8	IVP	initial value problem
10, 17, 26	JMM	Joint Mathematics Meetings
23	JOMA	Journal of Online Mathematics and its Application
25	LabSTOR	http://labstor.blogspot.com
17	LAU	Lebanese American University
7	LSODA	Livermore Solver for ODEs
14	MAA	Mathematical Association of America
22	MathDL	Mathematics Sciences Digital Library
10	MCM	Mathematical Contest in Modeling
21	MIT	Massachusetts Institute of Technology
22	NSDL	National Science Digital Library
5	NSF	National Science Foundation
21	OCW	OpenCourseWare
		(consortium of over 250 institutions worldwide)
		http://ocw.mit.edu
3	ODE	ordinary differential equations
11	ODEA	ODE Architect

19	RPI	Rensselaer Polytechnic Institute
18	SBML	Systems Biology Markup Language
13	SIR	susceptible, infected, recovered model for infectious diseases
23	SMETE	Science, Mathematics, Engineering, Technology, and Education
23	STEM	Science, Technology, Engineering, and Mathematics
23	UCAR	University Corporation for Atmospheric Research
17	UniSA	University of South Australia
20	USMA	U. S. Military Academy
25	VCL	Virtual Computing Lab
24	W|A	Wolfram|Alpha http://wolframalpha.com

10 Appendix III: Biographical Index

12	Hale	Margie	Stetson University
16, 19	Hall	Glen	Boston University
15, 19	Hall	James	Westminster College
7	Hindmarsh	Alan	Lawrence Livermore Labs
5	Hinkle	Ben	Cornell University
15,17,21	Hohn	Hubert	Massachusetts College of Art
3, 13, 14, 18	Hubbard	John H.	Cornell University
20	Huber	Michael	Muhlenberg College
2	Ince	Edward L.	Liverpool, Cairo, Edinburgh University
17	Jacobs	Kuva	University of South Australia
15	Johnson	Lee	Virginia Polytechnic University
14	Kallaher	Michael	Washington State University
24	Karaali	Gizem	Pomona College
22	Klotz	Gene	Swarthmore College
15	Kohler	Werner	Virginia Polytechnic University
20	Kostelich	Eric	Arizona State University
8	Larson	Ron	Pennsylvania State University
2	Levinson	Norman	MIT
13, 15	LoFaro	Tom	Washington State University, Gustavus Adolphus College
20	Lomen	David	University of Arizona
20	Lovelock	David	University of Arizona
10	Lucas	Bill	Cornell University, Claremont Graduate University
21	Mattuck	Arthur	MIT
14,15, 19, 26	McDill	Jeanie	Cal Poly San Luis Obispo
21	Miller	Haynes	MIT
11, 12, 19	Moody	Michael	Harvey Mudd College, Olin College
1, 8, 14, 22-24	Moore	Lawrence or Lang	Duke University
15, 19	Nagle	Kent	University of South Florida
13, 14	Noonburg	Anne	University of Hartford
17	Panton	David	University of South Australia
13, 25	Park	Jeho	Harvey Mudd College
19	Penney	David	University of Georgia
7	Petzold	Linda	Lawrence Livermore Labs
13	Polina	Ben	University of Hartford
19	Polking	John	Rice University
9	Porta	Horatio	University of Illinois
22	Porter	Jerry	University of Pennsylvania
21	Radunskaya	Ami	Pomona College
2	Rota	Gian-Carlo	MIT
15, 19	Saff	Edward	University of South Florida

3, 11, 13	Shampine	Larry	SMU
12	Siegmann	William	RPI
8, 14, 22, 24	Smith	David	Duke University
15, 19	Snider	Arthur	University of South Florida
5	Sours	Richard	Wilkes University
15, 20	Strogatz	Steve	Cornell University
9	Stroyan	Keith	University of Iowa
4	Tall	David	University of Warwick, UK
20	Taubes	Clifford	Harvard University
22	Tutsch	Jerry	University of Wisconsin
9	Uhl	Jerry	University of Illinois
21	Upton	Deborah	Stonehill College
14	Wattenberg	Frank	Univ. of Massachusetts, Montana State Univ., USMA
20	Weir	Maurice	U.S. Military Academy
1, 3, 10, 12-15, 18, 19	West	Beverly	Cornell University
4	Winkelmann	Bernard	University of Bielefeld, Germany
3	Wood	Ron	Harvey Mudd College
20	Wright	Warren	Loyola Marymount University
21	Yong	Darryl	Harvey Mudd College
24	Yoshiwara	Bruce	Los Angeles Pierce College
25	Zehnder	Alan	Cornell University
20	Zill	Dennis	Loyola Marymount University

References

[1] COMAP (COnsortium for Mathematics and its APplications). URL http://www.comap.com.

[2] CODEE Consortium. Ongoing website. URL http://www.codee.org.

[3] CODEE Newsletters, 1992 – 1997. URL http://www.codee.org/library/newsletters.

[4] *ICTME. Proceedings of the International Conference on Technology in Collegiate Mathematics in Beirut Lebanon.* July 2000. URL http://www.ilnprojects.com/teachersportal/resources-2/Mathematics/Technology%20in%20Mathematics%20Education.pdf.

[5] INTERFACE and Author's Guide for INTERFACE, 1973, 2000. URL http://ccdl.libraries.claremont.edu/col/ija. Author's Guide written by Harvey Mudd student Patrick Costello.

[6] MIT Mathlets, . URL http://www.math.mit.edu/mathlets. Or http://mathlets.org.

[7] MIT Open Courseware, . URL http://ocw.mit.edu.

[8] MathDL Partners Program. URL http://www.mathdl.maa.org/mathDL/?pa=content&sa=viewDocument&nodeId=2004.

[9] *CODEE Consortium. ODE Architect.* John Wiley and Sons, 1998. URL http://www.math.hmc.edu/resourses/odes/odearchitect.

[10] Wolfram|Alpha. URL http://www.wolframalpha.com/.

[11] Wolfram Demonstrations Project. URL http://demonstrations.wolfram.com.

[12] M. Abell and J. Braselton. *Differential Equations with Mathematica.* Academic Press, A Google e-book, 3^{rd} edition, 2004.

[13] M. Artigue and V. Gautheron. *Systèmes Differentiels: Étude Graphique.* CEDEC, Paris, 1983.

[14] Jeffrey Becker, Robert Borrelli, and Courtney Coleman. *Models for Applied Analysis: a Handbook for Projects.* October 1981. URL http://www.ccdl.libraries.claremont.edu/u?/mmm,4688.

[15] P. Blanchard, R. L. Devaney, and G. R. Hall. *Differential Equations.* Brooks Cole, 4^{th} edition, 2011.

[16] Paul Blanchard. Teaching Differential Equations with a Dynamical Systems Viewpoint. *Collegiate Mathematics Journal*, 25(5):385–395, November 1994.

[17] College Board. Calculus AB and Calculus BC course description. URL http://www.apcentral.collegeboard.com/apc/public/repository/ap-calculus-course-description.pdf. Downloaded on October 26, 2011.

[18] R. Borrelli and C. Coleman. *Differential Equations: A Modeling Approach.* Prentice Hall, 1987.

[19] R. Borrelli and C. Coleman. *Differential Equations: A Modeling Perspective.* Wiley, 2^{nd} edition, 2004.

[20] Robert Borrelli and Courtney Coleman. *Mathematical Methods, Models and Applications for Engineers, Mathematicians and Scientists.* 1977. URL http://www.ccdl.libraries.claremont.edu/col/mmm.

[21] William Boyce, Courtney Coleman, and Robert Borrelli. *Differential Equations Laboratory Workbook: A Collection of Experiments, Explorations and Modeling Projects for the Computer.* Wiley, 1992.

[22] M. Braun. *Differential Equations and their Applications: An Introduction to Applied Mathematics.* Springer Verlag, 4^{th} edition, 1992. Texts in Applied Mathematics #11.

[23] Martin Braun, Courtney Coleman, and Donald Drew (editors). *Modules in Applied Mathematics*, volume I. Springer Verlag, 1983. There are three other volumes in this series.

[24] James Callahan, Kenneth Hoffman, David Cox, Donal O'Shea, Harriet Pollatsek, and Lester Senechal. *Calculus in Context: The Five College Calculus Project.* Worth Publishing, Richmond, United Kingdom, 1998.

[25] CODEE Consortium. *ODE Architect Companion.* John Wiley & Sons, 1999.

[26] J. M. A. Danby. *Computing Applications to Differential Equations: Modeling in the Physical and Social Sciences.* Reston Publishing Co., Reston, VA, 1985. A classic – an excellent source for models, with variations on each.

[27] J. M. A. Danby. *Computing Modeling from Sports to Spaceflight: From Order to Chaos.* William Bell, Inc., Richmond, VA, 1997. Includes CD-ROM with IBM-PC software.

[28] Bill Davis, Horacio Porter, and Jerry Uhl. Calculus and *Mathematica* courses taught at Universities of Illinois and Ohio State. URL http://socrates.math.ohio-state.edu.

[29] Paul Davis. *Differential Equations: Modeling with Matlab.* Prentice Hall, 1999.

[30] C. Henry Edwards and David E. Penney. *Differential Equations and Boundary Value Problems: Computing and Modeling.* Prentice Hall, 4^{th} edition, 2008.

[31] C. Henry Edwards and David E. Penney. *Differential Equations and Linear Algebra.* Pearson, 3^{rd} edition, 2010.

[32] Jerry Farlow, James Hall, Jean Marie McDill, and Beverly West. *Differential Equations and Linear Algebra.* Prentice Hall/Pearson, 2^{nd} edition, 2007.

[33] Edwin Freed, Dan Newman, Kevin Carosso, and Kristin Hubner (HMC Students). The *Mathlib User's Guide*, the *Mathlib Reference Manual*, and the *Mathlib Example Set*, 1980-1984. URL http://www.ccdl.libraries.claremont.edu/cdm4/browse.php?CISOROOT=/mmm.

[34] F. Giordano and M. Weir. *Differential Equations: A Modeling Approach.* Addison Wesley, 1991.

[35] F. Giordano, W. Fox, S. Horton, and M. Weir. *A First Course in Mathematical Modeling.* Brooks Cole/Cengage, 2008.

[36] S. S. Habre. Exploring students' strategies to solve ordinary differential equations in a reformed setting. *Journal of Mathematical Behavior*, 18 (4):455–472, 2000.

[37] Samer S. Habre. *Dynamical Mathematical Software and Visualization in the Learning of Mathematics.* In press.

[38] S.S. Habre, J.H. Hubbard, and B.H. West. The convergence of an Euler approximation of an initial value problem is not always obvious. *American Mathematical Monthly*, 108(4):326–335, 2001.

[39] H. Hohn, B.H. West, S.H. Strogatz, J.M. McDill, and J. Cantwell. *Interactive Differential Equations*. Addison Wesley, 2000. URL http://www.aw-bc.com/ide.

[40] J.H. Hubbard and B.H. West. *Differential Equations, A Dynamical Systems Approach, Part I, The One-Dimensional Theory*. Springer Verlag, New York, NY, 1991. Texts in Applied Mathematics #5.

[41] J.H. Hubbard and B.H. West. *Differential Equations, A Dynamical Systems Approach, Part II, Higher Dimensional Systems*. Springer Verlag, New York, NY, 1995. Texts in Applied Mathematics #18.

[42] J.H. Hubbard and B.H. West. *Équations Différentielles et Systèmes Dynamiques*. Cassini, Paris, 1999. Translation and adaptation of [40] and [42] by Gautheron, V.

[43] J.H. Hubbard and B.H. West. *MacMath 9.2*. Springer Verlag, New York, N.Y., 1992.

[44] J.H. Hubbard and B.H. West. *MacMath 10.0, unpublished*. 1999. URL http://www.math.cornell.edu/~dynamics/BenHinkle/index.html. Alas, the package only runs on the Mac Classic operating system, but the programs for 2D differential equations and 2D maps are available online for those with a Mac Classic platform.

[45] J.H. Hubbard, J.M. McDill, A. Noonburg, and B.H. West. A New Look at the Airy Equation with Fences and Funnels. *College Mathematics Journal*, 25(25):419–431, November 1994. URL http://www.cecm.sfu.ca/organics/papers/hubbard/paper/htm/paper.html. This paper was redone in CMS Conference Proceedings, American Mathematical Society, Providence, RI, 1995.

[46] Michael Huber. Teaching differential equations with modeling and visualization. *CODEE Journal*, 2011. URL http://www.codee.org/library/projects/teaching-differential-equations-with-modeling-and-visualization.

[47] Kuva Jacobs and David Panton. Learning about ODEs using interactive online modules. *CODEE Journal*. URL http://www.codee.org/library/learning-about-odes-using-interactive-online-modules.

[48] Michael Kallaher (editor). *Revolutions in Differential Equations: Exploring ODEs with Modern Technology*. MAA, Notes #50 1999.

[49] Gizem Karaali and Bruce Yoshiwara. Life after Wolfram|Alpha: What you (and your students) need to know. *Loci*, January 2010. DOI: 10.4169/loci003365.

[50] W. E. Kohler and L. W. Johnson. *Elementary Differential Equations*. Addison Wesley, 2002.

[51] E. J. Kostelich and D. Armbruster. *Introductory Differential Equations from Linearity to Chaos*. Addison Wesley Longman, 1997.

[52] Tom LoFaro and Kevin Cooper. *Internet Differential Equations Activities*. URL http://www.idea.wsu.edu.

[53] D. Lomen and D. Lovelock. *Differential Equations: Graphics, Models, Data*. Wiley, 1999.

[54] Haynes Miller and Deborah Upton. Computer manipulatives in an ODE course: Development, implementation, and assessment. URL http://math.mit.edu/mathlets/wp-content/uploads/cet-published.pdf.

[55] Michael Moody. Guide to *ODE Architect* tool. URL http://www.math.hmc.edu/resources/odearchitect/docs/guide-to-odearchitect-solver-tool.pdf.

[56] Lawrence C. Moore and David Smith. Changing technology implies changing pedagogy, from *A Fresh Start for Collegiate Mathematics*. *MAA Notes #69*, pages 229–234, 2006. Nancy Baxter Hastings editor.

[57] K.R. Nagle, E.B. Saff, and A.D. Snider. *Fundamentals of Differential Equations*. Addison Wesley, 8^{th} edition, 2011.

[58] Jeho Park. *ODE Architect*: Now and Future. *CODEE Journal*, 2011. URL http://www.codee.org/software/ode-architect/ode-architect-now-and-future.

[59] J. Polking, A. Boggess, and D. Arnold. *Differential Equations with Boundary Value Problems*. Pearson, 2^{nd} edition, 2006.

[60] John Polking. *Dfield* and *Pplane*. URL http://math.rice.edu/dfield/dfpp.html.

[61] Lawrence Shampine. *Numerical Solution of Ordinary Differential Equation*. Chapman and Hall/CRC, second edition, 1994.

[62] Lawrence Shampine. The Savvy Solver I. *CODEE Newsletter*, 1992. URL http://www.ccdl.libraries.claremont.edu/u?/cla.

[63] Lawrence Shampine. The Savvy Solver II. *CODEE Newsletter*, 1993. URL http://www.ccdl.libraries.claremont.edu/u?/cla.

[64] Lawrence Shampine. The Savvy Solver III. *CODEE Newsletter*, 1993. URL http://www.ccdl.libraries.claremont.edu/u?/cla.

[65] Lawrence Shampine. The Savvy Solver IV. *CODEE Newsletter*, 1994. URL http://www.ccdl.libraries.claremont.edu/u?/cla.

[66] A. Skidmore and M. Hale. *A Guided Tour of Differential Equations using Computer Technology*. Prentice Hall, 1998.

[67] David A. Smith and Lawrence C. Moore. *Calculus: Modeling and Application*. D.C. Heath and Company, Lexington, Massachusetts, 1996. The second edition is available online by the MAA. URL `http://www.calculuscourse.maa.org/`.

[68] Keith D. Stroyan. *Calculus: The Language of Change*. Academic Press, San Diego, California, 1993.

[69] D. Tall and B.H. West. Graphic insight into calculus and differential equations. *ICMI Conference Proceedings*, 1986. Also, published in *The Influence of Computers and Informatics on Mathematics and its Teaching*, Cambridge University Press, pp. 107-119.

[70] C.H. Taubes. *Modeling Differential Equations in Biology*. Prentice Hall, 2001.

[71] J.H. Tutsch. OdeFactory. URL `http://www.odefactory.com`.

[72] B. H. West (editor). Special Issue on Differential Equations. *Collegiate Mathematics Journal*, 25(5), November 1994.

[73] Ron Wood and Andrew Flint. ODE solvers for the classroom. *College Mathematics Journal*, 25(5):458–461, 1994.

[74] Alan Zehnder. Review of *Pplane* and *Dfield*. *College Mathematics Journal*, 27(2): 144–148, 1996.

[75] D. Zill and W. Wright. *Advanced Engineering Mathematics*. Jones and Bartlett, 4^{th} edition, 2011.

[76] D. G. Zill and M. Cullen. *Differential Equations with Boundary Value Problems*. Brooks Cole, 7^{th} edition, 2008. 5^{th} ed. is available on Kindle; 8^{th} ed. by Cengage Youbook. Was available in March 2008 at a cost of $255!