INTERACTIVE APPLETS IN CALCULUS AND ENGINEERING COURSES

HEIDI BURGIEL, CHAD LIEBERMAN, HAYNES MILLER,
AND KAREN WILLCOX

In the fall of 2000, a group in the MIT Mathematics Department began work on a suite of computer applets ("MIT Mathlets") designed to support basic MIT mathematics education. For a report on this work and a review of relevant literature see Miller and Upton [3]. The current state of this collection, along with supplementary material, can be found at http://math.mit.edu/mathlets/. The intent of this chapter is to discuss the recent design and application of two new families of MIT Mathets serving very different audiences:

- Calculus tools designed for self-learners and delivered over the internet through MIT OpenCourseWare.
- Tools designed for use in an aerospace engineering classroom workshop to aid in understanding the stability of certain difference schemes.

We will begin by describing the pedagogical factors guiding the creation of this family of applets, the history of their early development, and the background for the recent additions. The next section is devoted to the design and usage of calculus applets for self-learners, and Section 3 to similar treatment of the Eigenvalue Stability applet built for use in an Aerospace Engineering class. We wrap up each section with some lessons learned and a sketch of future work in this direction.

1. Background

Development of the MIT Mathlets began in the Fall of 2000 under a grant from the MIT d'Arbeloff Fund for Excellence in Education. Initially they focused on material for the large course on Ordinary Differential Equations. In the spring of 2003 they were first introduced in that course as lecture demo material and, more powerfully, as a framework for homework assignments. This initial phase of development is reported on by Miller and Upton in [3] (available online), where a detailed study of the pedagogical impact of the applets can be found. As described in that paper, the initial MIT Mathlets began as modifications of applets created for an earlier project, Interactive Differential

Equations [4], authored by Beverly West, Steven Strogatz, Jean Marie McGill, and John Cantwell. The IDE applets were designed and programmed by Hu Hohn, Director of the Computer Arts Center at Massachusetts College of Art, and Hohn continued the development of this initial group and many subsequent applets for the MIT project.

Several well-defined overarching design elements, described in [3], shaped the creation of these tools and continue to shape the new ones. They may be summarized as follows. Each applet

- responds in real time to slider or cursor movements determining system parameters,
- is narrowly focused, providing maximal insight into important concepts using carefully selected examples, and
- is designed to maximize the visual connections among components, using color coding, layout, and other methods.

In the ensuing years many homework problems incorporating these applets have been written (and many are available at the MIT Mathlets site) and further applets were added to the collection. Some of the new applets were again connected with differential equations, but increasingly attention turned to providing material for use in other courses, especially in courses outside the Mathematics Department. Most of our engineering students will have seen these applets in use in the basic differential equations course and will be familiar with their visual conventions and their philosophy of multiple representation and cursor controls. They are therefore easier for these students to use, and provide an attractive link between courses in different departments and different years of the student's career.

Here is a list of recent transdepartmental projects. They were all programmed by Jean-Michel Claus.

- 2007: With Peter Dourmashkin of the MIT Physics Department, the applet Series RLC Circuit was designed and implemented as part of the TEAL approach to basic instruction in electromagnetism. This applet has subsequently also been used in the Differential Equations course, providing an important and conspicuous link between courses.
- 2007–2008: Julie Greenberg of the Harvard-MIT Division of Health Sciences and Technology designed the applet Discrete Fourier Transform. It is used in the HST course "Biomedical Signal and Image Processing."
- 2008–2009: With John Deyst and Karen Willcox (of the MIT Aero-Astro Department) the applet Nyquist Plot was designed and built. A subsequent collaboration with Franz Hover of the

MIT Mechanical Engineering Department resulted in the applet Bode and Nyquist Plots.

- 2010: With Karen Willcox and Chad Lieberman, the applet Eigenvalue Stability was designed and built, and a worksheet guided students through its use in a class in the course Computational Methods in Aerospace Engineering.
- 2010: With Heidi Burgiel and Jerry Orloff (of MIT Experimental Study Group), a group of applets illustrating basic calculus concepts was designed and built for use with OCW Scholar.

In this chapter we shall report in detail on the last two developments. These two differ greatly in their pedagogical context and in their target audiences, and together they present a case study of how a common technology can address diverse pedagogical needs.

2. Calculus Mathlets for Self-Learners

In the spring of 2011, MIT OpenCourseWare launched OCW Scholar, reorganizing and augmenting the OCW content of a series of foundational undergraduate MIT courses to provide a pathway through the course suitable for use by self-learners. Two of the first set of OCW Scholar courses are mathematics courses: Single Variable Calculus and Multivariable Calculus. Applets were designed and built for use in the first of these courses.

Visitors to these sites are often working alone, so students must be able to operate and interpret these mathlets without instructor or peer support. This isolation results in some surprising requirements. For example, one student remarked on a chat site associated with the course:

> The mathlets are clear, but when I use them for the exercises then I get confused. For example, when they say to find the slope of the tangent line, I don't know if they're talking about the red tangent line or the yellow tangent line.

The student was referring to the Secant Approximation applet; see Figure 1. This student does not yet distinguish between tangent lines and secant lines. The quote illustrates how easy it is for a student to experience a sense of understanding without actually comprehending what the visual is designed to convey. This gap in understanding is only revealed when the student is presented with an exercise that links the visual with a conceptual or computational task. Labels, color coding, and layout do not by themselves induce an understanding of the intent of the applet. This is a universal characteristic of student response to applets, but it is perhaps especially severe for self learners.

All the MIT Mathlets have a Help page which details the various functionalities of the applet. As part of our response to the additional explanatory demands imposed by the OCW Scholar distance learning model, we have prefaced that text with a brief description of the conceptual content and intended use of the applet.

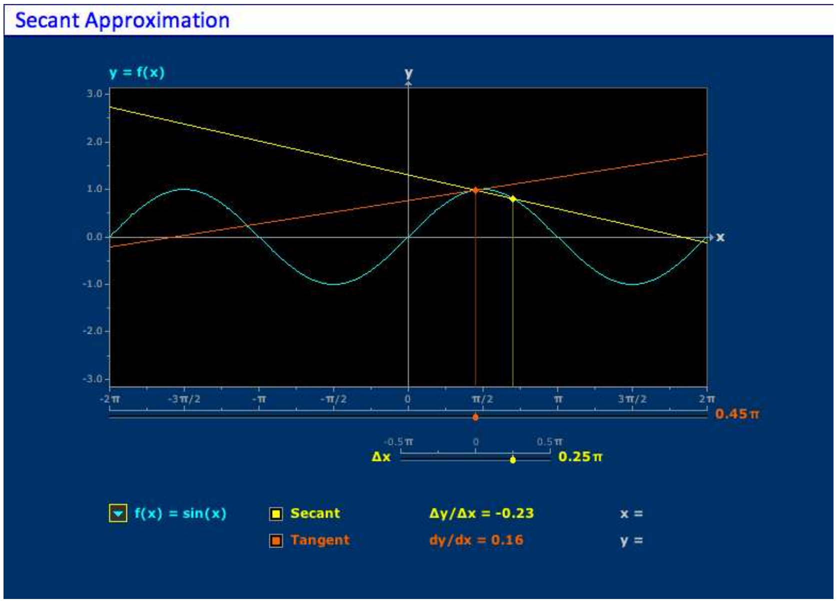


Figure 1. The Secant Approximation Applet.

2.1. **Design of the Secant Approximation Applet.** The applet Secant Approximation (see Figure 1) illustrates many of the design features common to all the MIT Mathlets. This applet is designed to illustrate the convergence of secant lines to the tangent line at a point on a graph; this is a graphical image of the definition of the derivative of a function.

A menu offers a choice of five functions. We argue that a few well chosen functions suffice to illustrate all the aspects of this particular curricular juncture which one is likely to want to explore. The advantage of limiting choice is that we have total control over the visual characteristics of the graph. This principle extends to all the applets in the MIT Mathlets collection.

The size and shape of the graphing window is independent of the choice of function, but the units vary. For trigonometric functions, the horizontal axis is marked off in multiples of π, while the vertical axis is in units. The aspect ratio is correct, however, so that a line of slope one makes a 45° angle with the x-axis. Failing to adhere to this kind of proportion can disrupt student comprehension.

The color cyan is used for three components in the applet: the choice of function, the graph of that function, and the label `y=f(x)` of the graph. These are all representations of the same data, and so are displayed in the same color.

A slider below the graphing window allows the user to select a value of x. The slider position is indicated by a red dot, which is repeated on the graph above and connected to it by a thin vertical red line segment, to drive home the connection between these two marks. There is also a red readout of the x value to the right of the slider. Red indicates the position on the graph.

A yellow checkbox labeled `[Secant]` offers the option of displaying data connected with the secant line, all colored yellow: a slider, centered at the chosen value of x, allowing choice of Δx; a readout of the value of Δx; the secant line itself; a ghosted yellow line segment indicating the connection between Δx and $x + \Delta x$; and a readout of the difference quotient $\Delta y/\Delta x$. The Δx slider is centered at the position of x. Sliders which slide is a novelty of this tool, and JAVA code executing it had to be added to the library.

One of the standing conventions, valid in all the MIT Mathlets, is that clicking on one of the numbered hashmarks sets the value of the corresponding parameter to that value. Important values are also artifically enforced: It may not be possible to station the Δx slider, for example, exactly at $\Delta x = 0$, because the computer screen is divided into discrete pixels. But careful programming guarantees that dragging the slider across $\Delta x = 0$ will result in the display appropriate to $\Delta x = 0$. This is one example of the many ways one needs to "cheat" in order to more accurately reflect the mathematical truth.

When $\Delta x = 0$, the readout for $\Delta y/\Delta x$ reads "`Undefined`." What secant line should be displayed when $\Delta x = 0$? The yellow dot slides behind the red one, and with no second point for the secant line, there is no secant line. It disappears. This is another standard convention of these applets: when a displayed object ceases to have meaning, it disappears.

Now is the time to take the limit. This is achieved by clicking on a second checkbox, labeled `[Tangent]`. Since it pertains to the given value of x, it is red. When it is checked, the tangent line at this point is drawn—in red, of course—and the value of the derivative is displayed, also in red.

A final functionality of this tool, again standard in the MIT Mathlets, is the creation of crosshairs over the graphing window centered at the cursor location (and suppressed by depressing the mouse key) and a readout of the values of x and of y under the cursor. The intent here is to allow the user to perform "experiments"—verifying the displayed value of the slope of the secant line, for example.

The Secant Approximation mathlet incorporates many visual cues to support student understanding of the relationship between secant and tangent lines. It is as simple and easy to use as possible.

2.2. **Use of the Secant Approximation Mathlet.** These applets can be used in a variety of ways. Experience shows that however charming they may seem to the professional, free exploration of them does not work well; the graphical medium is not intrinsically more conceptual than the symbolic one, and students tend to be timid in exploring and unsure what to look for. Individual students need a highly structured and detailed script. For groups one may be somewhat more open ended.

We may use Secant Approximation to illustrate the potential usages of these applets.

Elementary use. A straightforward way to get students involved with the MIT Mathlets is to invite them to make some measurements and then verify them computationally. We illustrate this approach using Secant Approximation. The main point of this applet is to help students grasp the trick needed to make sense of the tangent line at a point on a curve, here the graph of a function: You have to pick a *different* point on the graph, consider the "secant line" through those two points, and then look at the limit as the second point approaches the first. The elementary problems begin with that.

(1) Select $f(x) = 0.5\,x^3 - x$ from the pull-down menu, and be sure that the `[Secant]` checkbox is checked and that the `[Tangent]` checkbox is unchecked. Set $x = 1.0$ (by clicking on the hashmark, for example). Move the Δx slider and observe the secant line. Make a small table of values of $\Delta y/\Delta x$ as Δx ranges from -0.1 to $+0.1$. What is your prediction about the slope of the tangent line? Why is $\Delta y/\Delta x$ "undefined" (and why does the secant line vanish) when $\Delta x = 0$?

(2) Now check `[Tangent]` and uncheck `[Secant]`. (There is no need for the Δx slider anymore, so it disappears!) Drag the x handle and observe that the red line is indeed always tangent to the cyan graph. Use the applet to estimate to within 0.01

the values of x for which $dy/dx = 0$. These are locations of extrema for the function $f(x)$!

(3) Now compute the difference quotient $\Delta y/\Delta x = (f(x + \Delta x) - f(x))/\Delta x$ by hand for this example, and take the limit as $\Delta x \to 0$. Does the calculation agree with the experimental prediction when $x = 1$? Does it agree with the observed values of x for which $dy/dx = 0$?

Intermediate use. One of the attractive features of carefully constructed computer manipulatives, such as the MIT Mathlets, is that they often invite questions which elucidate the educational point at hand but which one might not have been able to ask without them. These can take the form of a game, and can greatly help the student in absorbing what the applet has to offer. Again, the Secant Approximation applet provides some examples.

(1) Continue with $f(x) = 0.5\,x^3 - x$. Check both `[Secant]` and `[Tangent]`. Position the x slider at $x = 1.5$, and observe the range of values of the slope of the secant line as Δx moves across the range allowed by the applet. Record the extreme values. The difference between the extreme values gives an indicator of how small Δx has to be to obtain a good estimate for dy/dx. Now set $x = 0.5$ and $x = -1.0$, and make the same observations.
(a) What features of $f(x)$, near these three values of x, can explain the differences you observe?
(b) Sometimes the slope of the secant for $\Delta x > 0$ is greater than the derivative, and sometimes it's smaller. What features of $f(x)$ account for this?

(2) With the same settings, take $x = 1.0$, where the slope of the tangent line is $f'(1) = 1/2$. Make a table of values of $\Delta y/\Delta x - dy/dx$ for $\Delta x = -0.3, -0.2, \ldots, 0.3$. Use the quadratic approximation of $f(x)$ at $x = 1.0$ and this table to estimate the value of $f''(1)$, and compare this experiment with a hand calculation.

Advanced use. The applets can also be used to explore issues beyond the immediate curriculum for which they were designed. In fact it has been our experience that these applets reveal phenomena we had not previously recognized, even in very elementary settings.

An example using Secant Approximation: Select $f(x) = \cos(x)$ from the pulldown menu, and check `[Secant]`. Select a nonzero value for Δx. What is the maximal value of the slope of the secant line, as you vary x? (This will be a function of Δx.) [Answer: $\sqrt{2(1 - \cos(\Delta x))}/|\Delta x|$.]

2.3. **Other Calculus Mathlets.** We briefly mention several other innovative applets designed along the same principles, also for use in calculus courses.

• Creating the Derivative begins with the graph of a function (chosen from a pull-down menu). Checkboxes toggle displays of the tangent line, the graph of the derivative, the best approximating parabola, and the graph of the second order Taylor polynomial. The graph is drawn from some initial value of the independent variable up to some value which is controlled by a slider or by an animation; the graphs are "created."

• Graph Features uses sliders to control the coefficients of a cubic polynomial, which is graphed. Checkboxes toggle colored indications of regions where the graph is rising, falling, concave, or convex.

• Riemann Sums illustrates the geometric significance of a variety of versions of the Riemann sum, and illustrates the convergence properties of these approximations of the Riemann integral. One selects a function from a drop-down menu, and then selects a method using radio buttons from the options `Evaluation point`, `min`, `max`, `trapezoidal`, and `Simpson's`. In the case of `Evaluation point`, a slider opens by which one can choose where in each interval the function is to be evaluated. A second window illustrates the values of the corresponding sums as n increases. This tool can be very useful in helping students understand these choices, that for continuous functions they don't matter in the limit, and the improvement in rate of convergence offered by Simpson's rule.

2.4. **Lessons learned.** Part of the challenge of assessing the effectiveness of online education is the difficulty of observing student responses directly. Consequently, we have little data on student reactions to the applets designed for self-learners. The majority of our information comes from students using the OpenStudy study group associated with our online Single Variable Calculus materials.

OpenStudy represents an attempt to replicate in distance learning the kind of help one gets from friends or study groups on campus. OpenCourseWare Scholar has provided links to OpenStudy and encouraged formation of OpenStudy groups studying the OCW Scholar courses.

The OpenStudy interaction differs from a campus interaction in many ways, however. On June 18, 2011, there were 418 students enrolled in the Single Variable Calculus study group, but there are rarely more than 3 students online simultaneously. So at any given time the level of activity on this site is less than would be found in a small group exercise in a classroom. On the other hand, questions persist and answers do too, and this somewhat compensates for a lack of immediacy. It also provides a valuable reference for students.

A review of questions posted to the study group reveals that students are more likely to discuss specific homework problems than general concepts A typical question is "What is the linear approximation of $\sin(x)$ at $x = 0$?"; questions like "How are tangent lines related to secant lines?" are rare.

Questions such as the student quote given in our introduction have led to improvements in the MIT Mathlets. For example, in an earlier version of the Secant Approximation applet, there was no toggle invoking the secant line; in fact the term "Secant" never appeared on the applet. Adding the `Secant` checkbox (in yellow) increases the self-explanatory value of the tool, improves its logic, and encourages a better pedagogical sequencing.

No amount of on-screen information will make these tools self-explanatory for all or even most students, however. In their use on campus, it was found that using them in class helped students understand the significance of various elements and made it much easier for students to use them in homework assignments. An easy on-line analogue would be voice-over screen movies illustrating their use, and this is an enhancement which will accompany applets in subsequent OCW Scholar courses.

3. Aerospace Engineering Mathlet

In the spring semester of 2010, we developed an applet specifically for use in the Computational Methods in Aerospace Engineering course in the Department of Aeronautics & Astronautics at MIT. Although the mathlet was developed for this particular use, it is general to the pedagogy of eigenvalue stability (also known as absolute stability), an important concept for numerical methods courses in the advanced undergraduate curriculum [1, 2].

In our experience teaching undergraduate juniors and seniors in aerospace engineering, we have found that many students struggle with eigenvalue stability analysis, often focusing on the procedure instead of understanding the concept. With the applet, students are able to extricate themselves from laborious calculations and obtain real-time responses to variational modifications (e.g., by changing the time step or dominant eigenvalue magnitude for a given scheme). Whereas many of the MIT mathlets are self-contained and can be used to learn a mathematical concept from scratch, the eigenvalue stability mathlet is intended to accompany traditional instruction. Its primary role is to deepen understanding of the topic rather than to teach the process of the analysis. In the Computational Methods in Aerospace Engineering course we used the mathlet as part of an in-class laboratory experiment after students were taught the mathematical procedures of eigenvalue stability analysis.

This section is organized as follows. First, we give some of the basic foundational material on eigenvalue stability analysis. It is necessary to understand the basic theory in order to appreciate the functionality in the applet. In Section 3.2 we discuss the design considerations of the eigenvalue stability applet, emphasizing ease-of-use and maximizing the number of ways a user can interact with the applet. In Section 3.3 we discuss the use of the applet in three levels of difficulty (elementary, intermediate, and advanced) by drawing on lessons learned from an interactive laboratory assignment performed in the Computational Methods in Aerospace Engineering course in the spring semester of 2010. Lastly, we present a few directions for future work.

3.1. **Theoretical Background.** Differential equations provide the language in which the laws of nature are expressed, and, consequently, the mathematics by which we understand the behavior of systems. In aerospace engineering, e.g., differential equations describe the unsteady fluid flow around an airfoil, permitting engineers to estimate lift and drag, quantities of interest that play an important role in aircraft design.

In most realistic applications, the differential equation does not admit solutions that are expressible in terms of elementary functions, and the only way to get an accurate understanding of the solution is via numerical methods. Often, the differential equation has many (thousands or more) degrees of freedom, e.g., the pressures and temperatures at many locations in the computational domain.

There is a well-established family of numerical integration schemes, and understanding their diverse applications is part of any engineering course in numerical methods.

Stability is one important consideration in selecting a numerical integration scheme.[1] A numerical integration method is stable if the numerical approximation to the solution of the system of ODEs does not grow without bound. There are two forms of stability:

[1] Accuracy is another important, but separate, issue. Indeed, accuracy of a numerical scheme does not imply that its numerical solution to an ODE will be eigenvalue stable.

Zero stability: A numerical integration scheme is *zero stable* if the numerical solution is bounded as the number of time steps goes to infinity in the limit of **zero** time step.

Eigenvalue stability: A numerical integration scheme's application to a particular system of ODEs is *eigenvalue stable* if the numerical solution is bounded as the number of time steps goes to infinity for **finite** time step.

Eigenvalue stability depends both on the numerical integration scheme and on the governing equations to be integrated, in particular, through the associated eigenvalues of the linearization about an equilibrium. For a given numerical integration scheme, eigenvalue stability analysis leads to a stability criterion that restricts the choices of time step.

In this section we sketch the analysis of eigenvalue stability for a linear system of equations with the forward Euler integration scheme and the midpoint method. This section is not intended to be a comprehensive primer on eigenvalue stability (see e.g. [1, 2] for this), but rather to give the reader a sense of the mathematics involved and a greater appreciation of the discussion of the applet in the next section.

Consider the autonomous nth-order linear system of ODEs,

$$\dot{\mathbf{y}}(t) = A\mathbf{y}(t), \quad \mathbf{y}(0) = \mathbf{y}_0, \tag{1}$$

where $\dot{\mathbf{y}}(t) = d\mathbf{y}/dt$ is the vector of time-derivatives of the unknown variables $\mathbf{y}(t) = [y_1, y_2, \ldots, y_n]$, and A is the $n \times n$ system matrix. Generic (non-defective) square matrices can be diagonalized. This implies that a change of variables replaces (1) with a decoupled set of scalar equations of the form

$$\dot{x}(t) \equiv f(x) = \lambda x(t), \quad x(0) = x_0, \tag{2}$$

where $x(t)$ is the scalar unknown variable and λ is an eigenvalue (generally complex) of A.

In applying a numerical method, one chooses a time step Δt. Then one computes numbers x_k which approximate $x(k\Delta t)$ inductively, starting from the given value of $x_0 = x(0)$.

Example 1: Forward Euler Method
Let Δt be the time step and x_k be the numerical approximation to $x(k\Delta t)$ for each non-negative integer k. The forward Euler integration scheme approximates the time-derivative at x_k to first-order; i.e.,

$$\frac{x_{k+1} - x_k}{\Delta t} = \lambda x_k; \tag{3}$$

that is to say, $x_{k+1} = (1 + \lambda\Delta t)x_k$. Given x_0 and a time step Δt, one can calculate the approximate value of the unknown x at each time interval by iteration.

The multiplicative factor relating x_k to x_{k+1} is the **amplification factor**. It is a function $g(z)$ of the product $z = \lambda\Delta t$. In this scheme, $x_k = g(z)^k x_0$, which stays bounded provided $|1 + z| = |g(z)| \leq 1$. This imposes a restriction on time step Δt in terms of the eigenvalue λ. For a system of equations, this inequality must hold for all eigenvalues of the system matrix A.

This sets the pattern for more complex numerical schemes. The forward Euler is a single-stage numerical integration scheme. There are also multi-stage schemes that use data from more than one time step to estimate the solution at each step. The midpoint method is an example of a multi-stage scheme.

Example 2: Midpoint Method
The midpoint method for the equation $\dot{x} = \lambda x$ uses the scheme

$$x_{k+1} = x_{k-1} + 2\Delta t \lambda x_k \tag{4}$$

This is a homogeneous second order difference equation. Its solutions will be linear combinations of solutions of the form $x_k = g^k x_0$ for suitable values of g. To find them, substitute this expression for x_k into (4) to see

$$g^{k+1}x_0 = g^{k-1}x_0 + 2\lambda\Delta t g^k x_0.$$

Dividing by $g^{k-1}x_0$ we find a quadratic equation for g:

$$g^2 - 2zg - 1 = 0$$

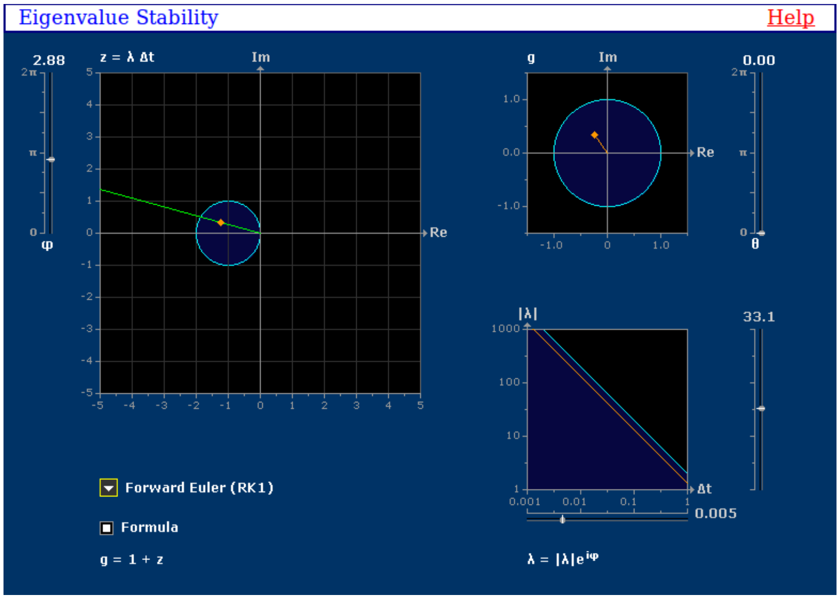
The quadratic formula gives $g(z) = z \pm \sqrt{1 + z^2}$.

The amplification factor in this case is the multivalued function $g(z) = z \pm \sqrt{1 + z^2}$. In applying the midpoint scheme to (1), the time step Δt will satisfy eigenvalue stability provided that $z = \lambda\Delta t$ satisfies $|g(z)| \leq 1$ for both solutions to the quadratic equation and all eigenvalues λ of A. In this scheme, the region of stability is small: it is easy to see that $|g(z)| \leq 1$ if and only if z is purely imaginary of modulus at most 1. This scheme is therefore eigenvalue stable only for systems with purely imaginary eigenvalues. These features of the midpoint method are dramatically represented in the Eigenvalue Stability applet.

The study of the amplification factor for a variety of numerical integration schemes, governing equations, and time steps is a critical component in any numerical methods course in the undergraduate curriculum.

In the next section, we describe the design of the Eigenvalue Stability applet.

3.2. **Design of the Eigenvalue Stability Mathlet.** The primary design considerations for the Eigenvalue Stability Mathlet included the ability to understand the stability analysis from several distinct perspectives, options to toggle between a handful of well-known numerical integration schemes, and also the ability to control the position of one eigenvalue of the governing equation and the time step of the numerical integration.



Forward Euler (RK1)

■ Formula

g = 1 + z

FIGURE 2. The Eigenvalue Stability Applet.

There are three graphical perspectives from which eigenvalue stability can be viewed (Figure 2).

- In the upper right, the g-plane displays the unit disk, i.e., the stability region, and also the complex point z mapped through the amplification factor $g(z)$. The angle θ can be scrolled to outline the unit circle in the g-plane and observe the tracing of the corresponding stability boundary in the z-plane.

- The large graph on the left shows the location of the point $z = \lambda\Delta t$ in the complex plane and the stability region, which is the inverse map of the unit disk in the g-plane. The angle φ is the phase of the eigenvalue λ.
- Finally, in the lower right, a plot of $|\lambda|$ against Δt shows the shaded region corresponding to the points where $z = |\lambda|\Delta t e^{i\varphi}$ lies in the stability region of the z-plane.

All of the perspectives are dynamically integrated so that a modification in one plot is immediately reflected in the others.

The Eigenvalue Stability Mathlet illustrates these three perspectives and the relationships between them for six commonly used numerical integration schemes. For any of these methods, one can choose to show or hide the formula for the amplification factor with a radio button that appears just below the drop down menu for the numerical method.

One approach to using the applet is to begin by specifying an eigenvalue in the governing equations of interest. This can be performed as follows:

(1) Specify the magnitude of the eigenvalue using the scrollbar on the lower right plot. This produces an orange circle in the z plane indicating all of the values of z associated with that magnitude.

(2) Select the phase of the eigenvalue using the φ scrollbar to the left of the z plane. Once selected, an orange dot appears at the intersection of the circle and the ray defined by φ.

(3) Select a numerical integration scheme and find the range of time steps for which the integration will be eigenvalue stable by adjusting Δt until the orange dot moves outside the blue-shaded region.

3.3. **Use of the Mathlet.** In this section we present a set of exercises of varying difficulty demonstrating how the eigenvalue stability applet can be utilized by students. The applet can be used as the primary apparatus in a series of laboratory experiments, can play an integral role in a homework assignment, or can be used in lecture to perform a real-time demonstration of the analysis of numerical methods. Some of the exercises discussed below were inspired by the laboratory worksheet used in class in the spring of 2010.

Elementary use. A critical component of eigenvalue stability analysis, and one that has often been a source of confusion for students, is the mapping of a region or boundary in one complex plane to the corresponding region in another. Eigenvalue stability is determined by the magnitude of the amplification factor $g(z)$, but it is the z-plane that contains the scaled eigenvalues. Thus, one needs to determine the region in the z-plane corresponding to the unit circle in the g-plane. This complex mapping can be explored by students for the six numerical schemes available in the drop-down menu:

(1) Select a numerical integration scheme from the drop-down menu and sketch the stability region in the z-plane by rewriting $g = e^{i\theta}$ and evaluating for several angles $\theta \in [0, 2\pi)$.
(2) Grab the θ slider to the right of the g-plane plot and slide it from $\theta = 0$ to $\theta = 2\pi$. The dots on the g- and z-plane plots turn yellow and the stability boundary is traced out in both planes.
(3) Verify the accuracy of your sketch of the stability region by comparing to the blue-shaded region in the z-plane.

Intermediate use. In the numerical solution of ODEs, it is necessary to balance, or tradeoff, accuracy, stability, and computational cost. Many schemes will be stable and accurate for very small time step, but then more time steps are required and therefore more computational resources will be used. For a set of governing equations and a given numerical integration method, it is therefore necessary to obtain the maximum time step for which the integration is stable.

(1) Calculate by hand the eigenvalues of the system matrix

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -1 \end{bmatrix}.$$

(2) Select `Forward Euler (RK1)` from the drop-down menu.
(3) For one eigenvalue, set the phase φ and magnitude $|\lambda|$ using the scrollbars in the applet.
(4) Determine the maximum allowable time step Δt by dragging the Δt scrollbar in the lower right plot. Observe as z changes magnitude and eventually crosses over the stability boundary.
(5) Does the maximum allowable time step change if you repeat the analysis with the other eigenvalue? Why or why not?

Advanced use. Although eigenvalue stability is often binary (either the scheme is stable or not for a particular problem), the eigenvalue stability applet can be used to demonstrate some additional nuances in the application of numerical methods. Even when a numerical scheme is eigenvalue stable, it still may not be accurate for a given mathematical model. In this example we consider the simulation of a simple harmonic oscillator, e.g., an undamped pendulum.

(1) Using the small angle approximation $\sin(\theta) \approx \theta$, the dynamics of an undamped pendulum (with well-chosen parameters) can be written $d^2\theta/dt^2 = -\theta$. At time $t = 0$, the pendulum is at rest at $\theta(0) = \pi/6$. Rewrite this second-order ODE as a system of ODEs by introducing the two-dimensional state vector $\mathbf{y} = [\theta, d\theta/dt]$.

(2) Implement the trapezoidal scheme and the midpoint scheme to simulate the pendulum, e.g. in MATLAB, from time $t = 0$ to $t = 10\pi$. Determine by hand the eigenvalues of the system of ODEs you derived in (1), and use the applet to choose a time step to preserve eigenvalue stability for each numerical integration scheme. (Do not select a time step that is too small since computational resources are limited.)

(3) What is the difference between the simulated responses? Which simulation is more consistent with the model of an undamped pendulum?

(4) Analyze the eigenvalue stability for the trapezoid and midpoint schemes once again using the applet. In particular, consider the position of the orange dot in the g-plane. What is the effect of the amplification factor in this problem?

3.4. **Future work.** Unfortunately, having used this applet only during one semester, we do not have much feedback from students. It is essential that we better understand students' abilities to learn eigenvalue stability through the use of our applet so that we can design additional teaching tools (lectures, homework problems, and laboratory assignments) and make improvements to the applet itself. In the future, we would like to integrate the eigenvalue stability applet into more of the numerical methods courses taught across other departments at MIT.

4. Conclusion

Carefully designed computer applets can be integrated into course work in a variety of contexts and subjects. We have detailed the design considerations and use of one applet used in a distance learning beginning calculus class and one used in a classroom workshop in computational methods in aerospace engineering. Other examples could easily be given. From OCW Scholar Single Variable Calculus we call attention particularly to Riemann Sums, and from engineering subjects Nyquist Plots and Bode and Nyquist Plots. The JAVA library developed to support these applets makes it quite easy to build new applets.

As is often the case with educational software, the greater challenge is in using them effectively within a course.

References

[1] Uri M. Ascher and Chen Greif, *A First Course in Numerical Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2011.

[2] Michael T. Heath, *Scientific Computing: An Introductory Survey*, 2nd Edition. New York, NY: McGraw-Hill, 2002.

[3] Haynes R. Miller and Deborah S. Upton, Computer manipulatives in an ordinary differential equations course: development, implementation, and assessment, *Journal of Science Education and Technology* 17 (2008) 124–137. Available at http://math.mit.edu/mathlets/wp-content/uploads/cet-published.pdf.

[4] Beverly West, Steven Strogatz, Jean Marie McDill, John Cantwell, and Hu Hohn, *Interactive Differential Equations*, Pearson Addison-Wesley (2008). http://www.aw-bc.com/ide/.

Department of Mathematics and Computer Science, Bridgewater State University, Bridgewater, MA 02325

E-mail address: hburgiel@bridgew.edu

Department of Aeronautics and Astronautics, MIT, Cambridge, MA 02139

E-mail address: celieber@mit.edu

Department of Mathematics, MIT, Cambridge, MA 02139

E-mail address: hrm@math.mit.edu

Department of Aeronautics and Astronautics, MIT, Cambridge, MA 02139

E-mail address: kwillcox@mit.edu