

A Tool for Construction of Bode Diagrams from Piecewise Linear Asymptotic Approximations*

ERIK CHEEVER and YUE LI

Dept. of Engineering, Swarthmore College, Swarthmore, PA 19081, USA.

E-mail: erik_cheever@swarthmore.edu

This paper describes a program intended to be an aid for teaching and learning the process of creating Bode diagrams by hand. Bode diagrams are a fundamental tool for understanding the frequency domain behavior of systems. The program, BodePlotGui, is a GUI (graphical user interface) tool written in the MATLAB[®] programming language. This program takes as input a transfer function, splits it into its constituent elements, then draws the piecewise linear asymptotic approximation for each element. This paper is available in interactive form on the IJEE website (<http://www.ijee.dit.ie>).

INTRODUCTION

BODE DIAGRAMS ARE critical to understanding frequency-domain analysis and design of linear systems. Although programs exist that accurately create Bode diagrams given a system transfer function (e.g. MATLAB[®]), students must still learn to draw Bode diagrams using piecewise linear asymptotic approximations to understand fully their relationship to the transfer function. Drawing piecewise linear asymptotic Bode diagrams by hand is a process that is often difficult to learn (and to teach), but it is important for students' understanding of how the locations of poles and zeros affect the shape of the graphs and, hence, the behavior of the system being studied.

Much of a student's difficulty derives from the relative scarcity of examples. Typically, the only resources a student has are a few examples in a textbook and perhaps a few from class notes. The ability to generate additional asymptotic Bode diagram examples to reinforce learned concepts, or to explore a range of 'what if?' scenarios concerning pole and zero placement, would benefit many students. MATLAB does have a function to generate such plots, *BodeAsym*, but it has several shortcomings. Among the shortcomings are: the function graphs do not show the individual terms of the transfer function separately; and do not show resonant peaks in underdamped systems.

The difficulty of teaching Bode diagrams is twofold. First, there is the conceptual problem of the derivation of the asymptotic approximations and a description of just what the Bode diagrams represent. This paper does not address these issues. The second problem, however, is more mechanical

and thus amenable to solution by computer; it is difficult to draw asymptotic Bode diagrams by hand. Drawing semi-logarithmic graphs by hand is inaccurate, whether it be on a chalkboard, whiteboard or on an overhead projector transparency. Paper handouts with examples worked out can be useful. They are not, however, generally useful for answering a student's 'what if?' questions, due to their static nature. Preprinted axes on overhead transparencies are another option, but these can be prone to errors, and also have difficulty answering many 'what if?' questions before the transparencies become cluttered and unusable.

This paper describes a software tool designed to address the difficulties experienced by students and teachers. The tool is a GUI (Graphical User Interface) written in MATLAB that takes as input a system transfer function. The program separates the function into its constituent terms (i.e. poles, zeros, and a constant term) and graphs the asymptotic approximation for each. The user can then selectively choose to exclude (or include) specific elements to simplify the transfer function or to isolate the behavior of a single part (the software is available at <http://www.swarthmore.edu/NatSci/echeeve1/Ref/Bode/BodeFiles.html>).

THE SOFTWARE

A description of the software follows, along with several examples. Due to the length of the code—close to one thousand lines of code comprising nine functions—only a brief outline of the software is given. The algorithm used by the software to draw asymptotic Bode plots closely follows the rules taught to undergraduates.

* Accepted 25 July 2004.

The software is invoked via the MATLAB command *BodePlotGui(Sys)* where *Sys* is a transfer function object in the MATLAB environment. The first function called up by the GUI is *BodePlotTerms*. This function takes as input a transfer function and decomposes it into seven different types of elements: a constant term, real poles and zeros (not at the origin), complex poles and zeros, and poles and zeros at the origin. The function then determines if any poles and zeros have a multiplicity greater than one by checking if they are equal within a predefined accuracy (set by default to 1 part in 10^3). As a last step, the function checks that all poles or zeros are in the left half of the *s*-plane (negative real part). If there are poles in the right half-plane, the program prints a warning and quits. If there are zeros in the right half-plane, it prints a warning that this is a non-minimum phase system, but continues. Poles and/or zeros on or very near the $j\omega$ axis (excluding the origin) result in a warning that the Bode plot is likely to be inaccurate near those frequencies (since the function is going to zero or infinity, which is impossible to show on a logarithmic, dB, scale).

The next function called up is *BodePlotter*, and it does the bulk of the work. It takes each term (from *BodePlotTerms*) and draws a piecewise linear approximation for magnitude and phase. To these diagrams it adds the exact magnitude

and phase functions (using the built-in MATLAB function *bode*). Lastly, it displays the transfer function displayed by calling up another function, *BodePlotDispTF*.

The rest of the functions are devoted to the operation of the GUI and are not discussed here. Any reader who is interested can refer to the MATLAB code.

A SIMPLE EXAMPLE

Consider the transfer function:

$$H(s) = 100 \frac{s}{s+10} = 10 \frac{s}{1 + \frac{s}{10}}$$

This function has three separate terms to be considered when constructing a Bode diagram:

1. a multiplicative constant of 10 (20 dB);
2. a pole at $\omega = 10 \text{ sec}^{-1}$; and
3. a zero at the origin.

The following MATLAB commands begin execution of the GUI:

- `MySys = tf(100*[1 0],[1 10]); %define Xfer function`
- `BodePlotGui(MySys) %Invoke GUI`

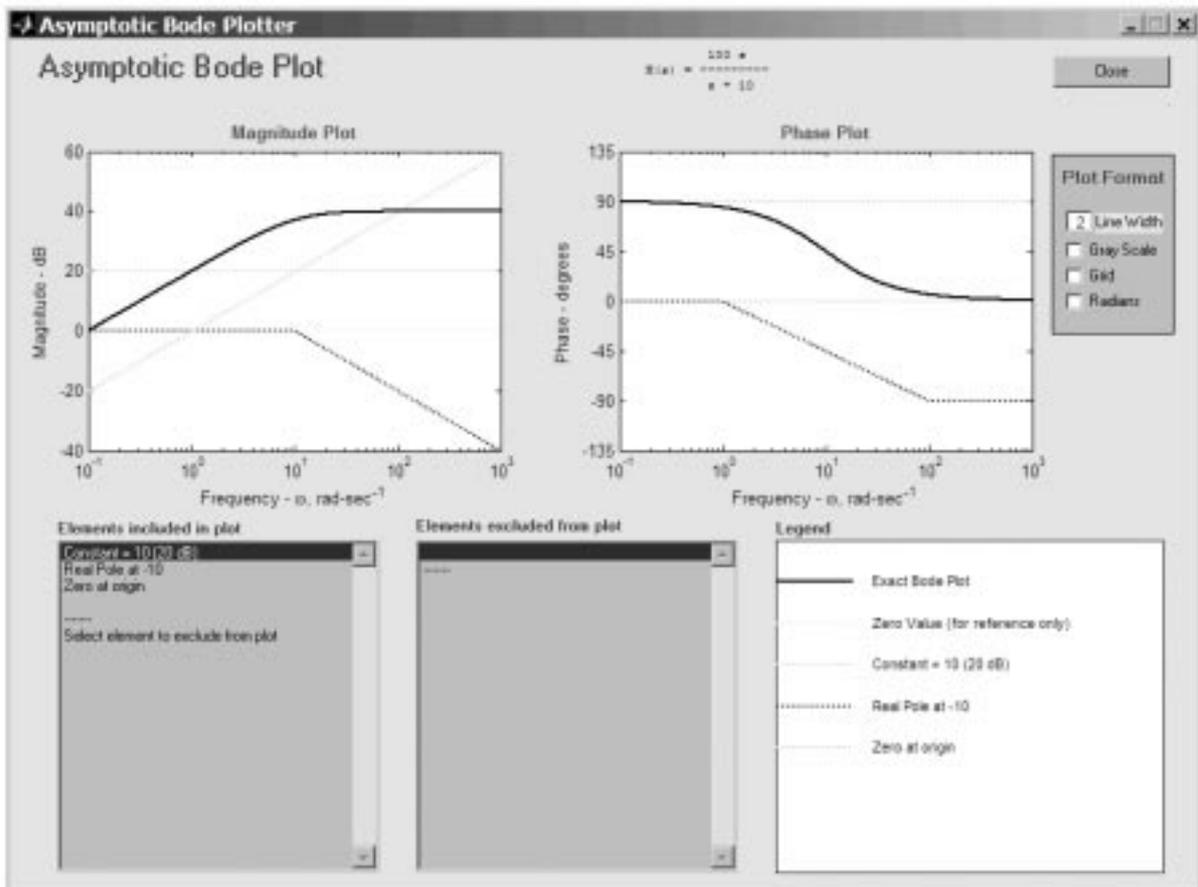


Fig. 1. Display of the GUI for a simple transfer function with a pole at $\omega = 10 \text{ sec}^{-1}$, a zero at the origin, and a constant term of 100.

The GUI generates a window as shown in Fig. 1. Starting at the upper left and going clockwise, the windows show:

- The magnitude plot, showing both the piecewise linear approximation for all three terms as well as the exact Bode diagram for magnitude. Also shown is a zero reference.
- The phase plot.
- A box that allows the user to format the plot by changing the width of the lines, adding grids.
- The legend identifying individual terms on the plot.
- A box that shows elements excluded from the plot. This box is empty in this display because the diagram displays all three elements of the transfer function.
- A box that shows elements displayed in the plot.

Also shown in the upper right-hand corner is the transfer function $H(s)$. Although the grayscale makes the individual terms of the Bode diagram difficult to distinguish, on a color display the differentiation between the plots for each term is much more dramatic.

The function displayed can be manipulated term by term to illustrate the effect of each term. For example, the zero at the origin can be excluded simply by clicking on it in the lower left-hand box.

Figure 2 shows the result, along with an annotation describing the operation performed. Note that the zero term is gone from the display and the transfer function, $H(s)$, and legend have changed to reflect this.

COMPLEX POLES AND ZEROS

When constructing Bode diagrams, underdamped systems with complex poles and/or zeros present an added difficulty because of the resonant peaks. The software described here graphs the piecewise linear asymptotic Bode diagram and also shows the resonance that develops as a deviation from the zero decibel line. For an underdamped pole the peak height is

$$M_r = \frac{1}{2\zeta\sqrt{1-\zeta^2}},$$

and occurs at the frequency

$$\omega_r = \omega_0\sqrt{1-2\zeta^2}.$$

These are well-known equations and occur in a myriad of books on Linear Systems or Control Theory [1].

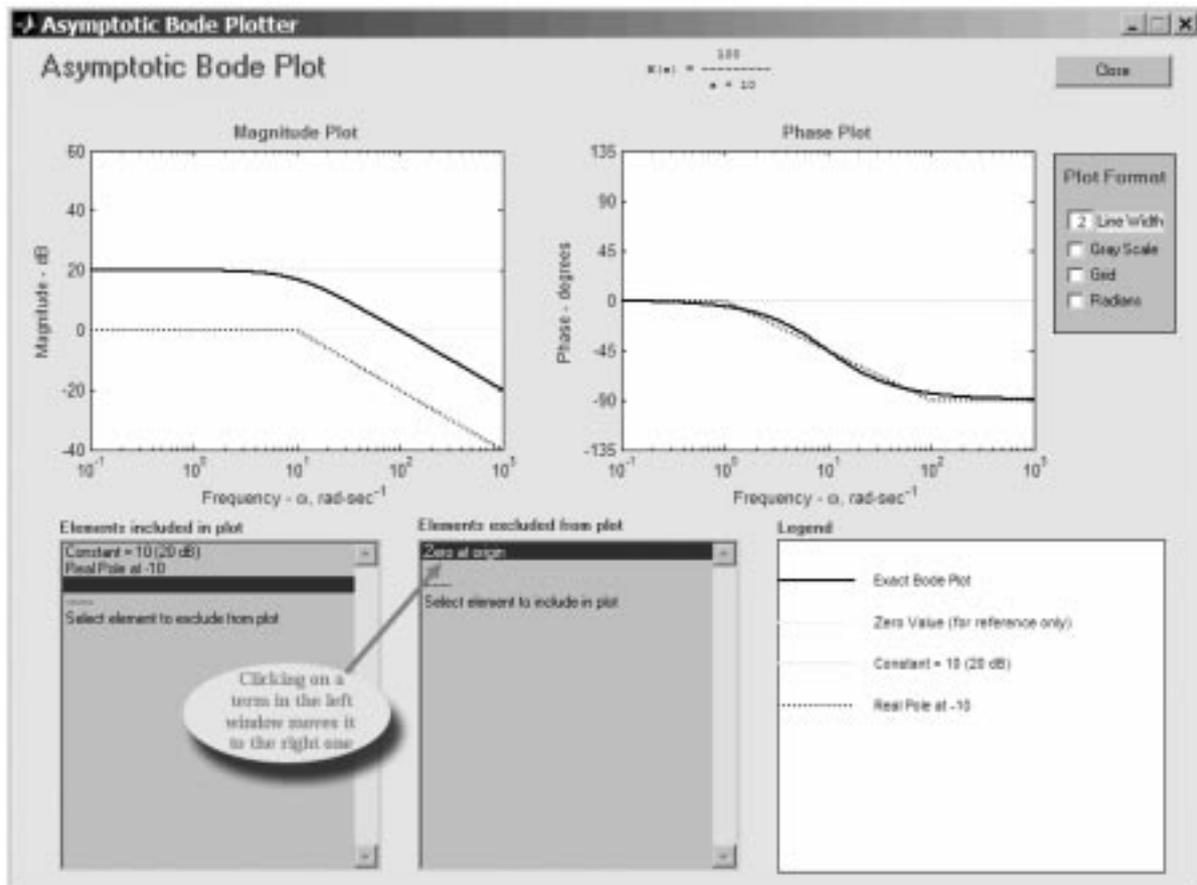


Fig. 2. Identical to Fig. 1, but with the zero at the origin excluded. Clicking on the term when it was in the 'Elements included' window moved it to the 'Elements excluded' window. The legend and transfer function (at top of window) have changed to reflect the change.

Drawing the phase for a complex pole or zero is more complicated. There are no well-known rules for drawing piecewise linear approximations for the phase of underdamped systems. Instead, most textbooks use phrases like ‘There is no easy way to draw such phase plots; refer to the phase-angle plots in Figure . . .’ to guide students. Clearly, this advice is not helpful in the creation of a computer algorithm for drawing phase plots. To develop an approximation to use with this program, the error between the actual phase and a variety of different piecewise linear functions was calculated for a wide range of ζ . It was empirically determined that, for an underdamped pole, a piecewise approximation that is 0° for low frequencies, -180° for high frequencies and linearly varying between

$$\omega = \omega_0 \frac{\log_{10}\left(\frac{2}{\zeta}\right)}{2}$$

and

$$\omega = \omega_0 \frac{2}{\log_{10}\left(\frac{2}{\zeta}\right)}$$

yielded a small error between the approximation and the exact value of the phase. If $\zeta < 0.02$, a straight line is drawn from 0° to -180° at $\omega = \omega_0$.

Figure 3 shows an example of a system with an underdamped pole. In this figure the transfer function $H(s)$ is

$$H(s) = \frac{1}{s^2 + 2s + 100}$$

This transfer function yields two separate terms for the Bode diagram:

1. a dc gain of 0.01 (-40 dB); and
2. a complex pole with $\zeta = 0.1$ and $\omega_0 = 10$.

Note that Fig. 3 shows the resonant peak, and that the legend shows both ω and ω_0 as well as the location of the pole in the s-plane (i.e. $\sigma + j\omega$).

A more complicated example

Now consider a more complicated transfer function with repeated poles. To wit, let

$$H(s) = \frac{s^2 + 2s + 100}{s(s + 100)^2} = 0.01 \frac{\left(\frac{s}{10}\right)^2 + 0.2\left(\frac{s}{10}\right) + 1}{s\left(\frac{s}{100} + 1\right)^2}$$

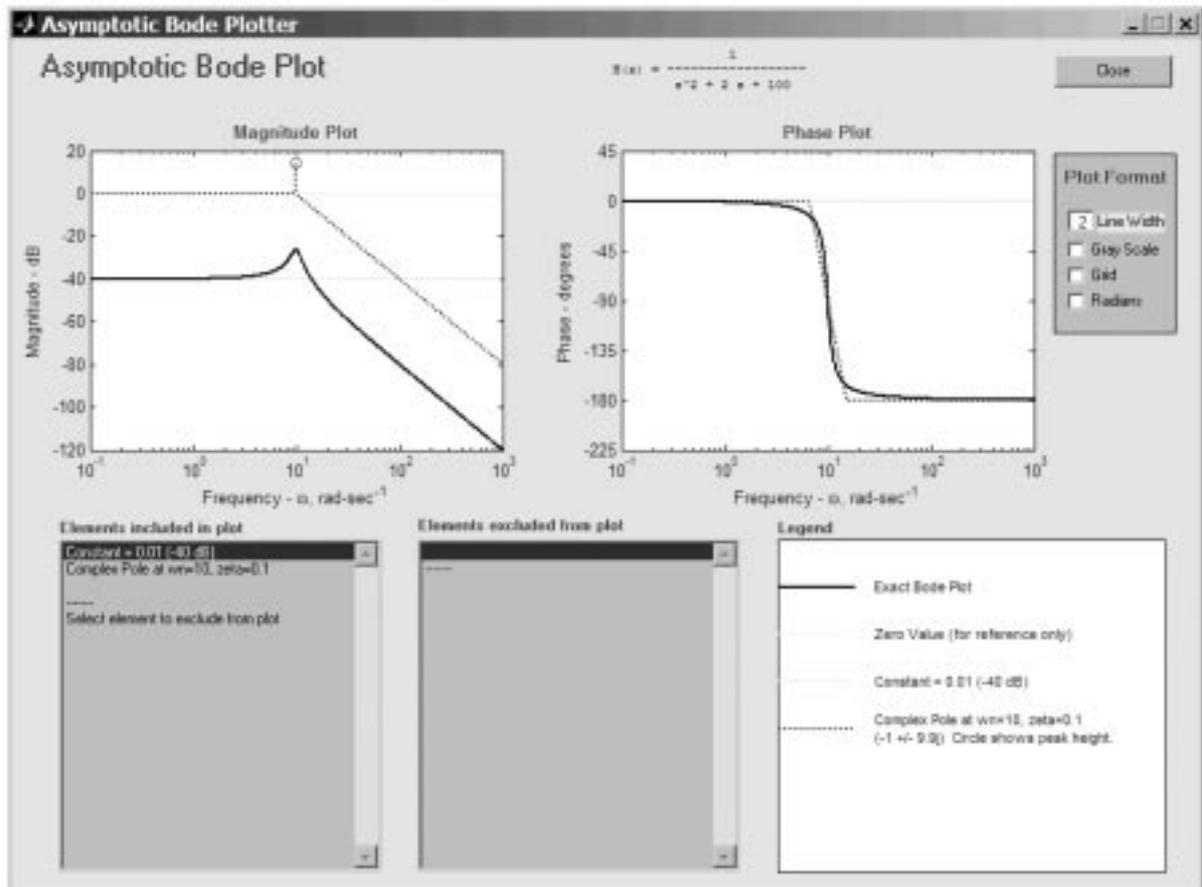


Fig. 3. This diagram demonstrates the display of an underdamped (resonant) term. The magnitude plot shows a displacement from the 0 dB line as a single point. The piecewise linear phase approximation is described in the text. The legend gives the natural frequency and damping ratio as well as the pole location in the s-plane.

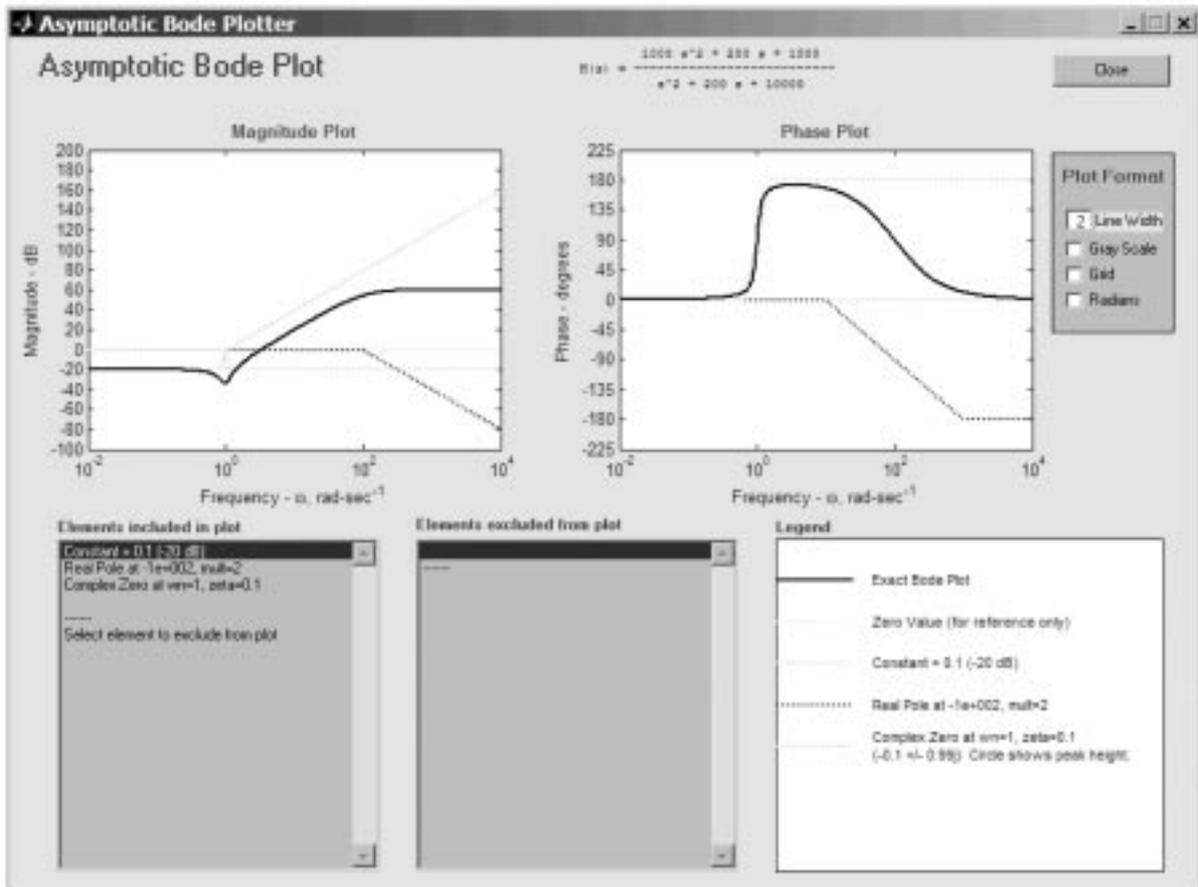


Fig. 4. When a pole (or zero) is repeated the GUI notes this and changes the display accordingly. The legend and 'Items included' windows both show a multiplicity of two, and the magnitude and phase plots reflect the double pole at $\omega = 100 \text{ sec}^{-1}$.

This transfer function has four separate terms to be plotted:

1. a constant of $0.01 = -40 \text{ dB}$;
2. a pole at the origin;
3. a repeated pole at $\omega = 100$; and
4. a complex zero with $\zeta = .1$ and $\omega_0 = 10$.

Figure 4 shows the output from the program. Note that the legend indicates that the pole at $\omega = 100$ has a multiplicity of two.

LIMITATIONS

The software has some known limitations. It does not handle the case of a time delay. It can only decompose transfer functions with up to 20 separate terms, due to the number of colors used. Repeated poles and zeros are treated as a single term (i.e. it is not possible to move only one pole of a repeated pair from the 'Elements included in

plot' window to the 'Elements excluded from plot' window of the GUI).

CONCLUSIONS

The *BodePlotGui* program is a versatile program for teaching and learning the construction of Bode diagrams from piecewise linear approximations.

For teachers, it offers the ability to generate multiple examples in class with neat accurate drawings. Furthermore, it enables the teacher to isolate the individual elements of the Bode diagram to demonstrate how each contributes to the resultant plots.

This software gives students the ability to generate multiple examples to help in the learning process. He or she can take examples from class or from a textbook and reproduce them. They can also break down any example they find, or create themselves, into its constituent parts. Moreover, they can do so at their own pace.

REFERENCES

1. K. Ogata, *System Dynamics*, 3rd edition, pp. 480–481. Prentice-Hall, Upper Saddle River, NJ (1998).

Erik Cheever is a Professor of Engineering at Swarthmore College. He received his B.Sc. in Engineering from Swarthmore College and his Master's and Ph.D. in Bioengineering from the University of Pennsylvania. Professor Cheever's teaching interests include circuits, linear systems and signal processing. His research interests include using embedded web servers for microcontroller user interfaces, and designing genetic algorithms for analog circuit design.

Yue Li is a senior student at Swarthmore College. He majors in Engineering and minors in Environmental Studies. He received all his education in Shanghai, China, before he came to Swarthmore three years ago. He has basic knowledge of CSS and Javascript and is involved in several web design projects at Swarthmore College.