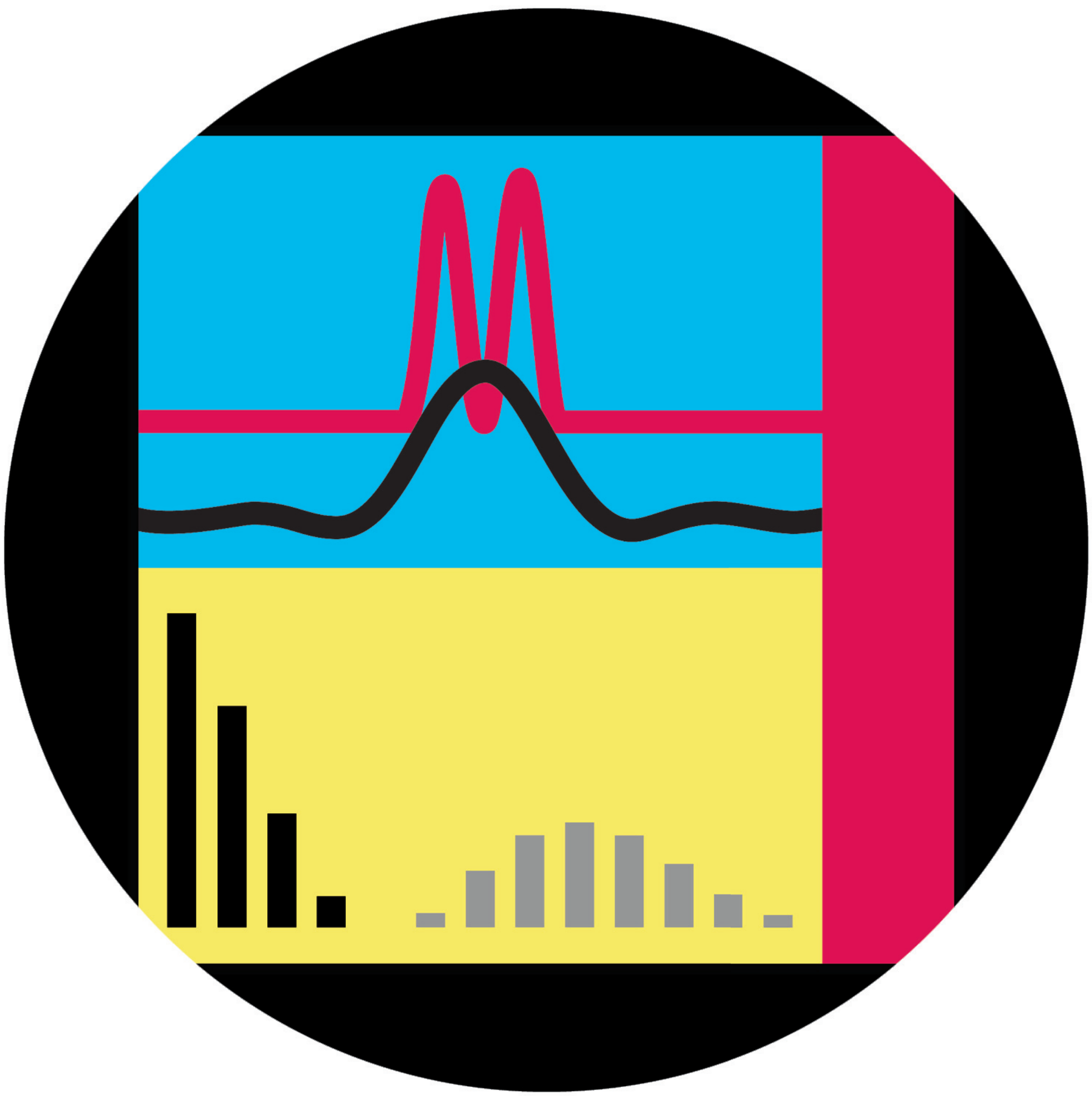
MacScope II



Instruction Manual

The MacScope II program is an audio software oscilloscope for both Mac and Windows which uses the computer's sound input to acquire data. The main features are Fourier analysis, signal averaging and working with triggered curves.

ISBN 0-9770828-0-6

AUTHOR'S INTRODUCTION

In my third year of teaching physics at Dartmouth College, back in the 1960s, I was assigned to teach a large introductory course. The lab for that course had a collection of Heathkit oscilloscopes which I, as a theoretical physicist, found impossible to operate. If you changed any scale, the curve disappeared somewhere for a few seconds, then floated by a few times before settling down somewhere. I refused to teach the course unless we scrounged up some Tektronix scopes that were stable enough for even me to operate. Shortly after that I got a grant to replace all the Heathkits with Hewlett Packard scopes which, remarkably, are still running today.

When I first saw the Macintosh computer in 1984, with its scroll bars for controls, I realized that the computer would make a great oscilloscope. The advantage would be that the data would end up in the computer where it could be immediately analyzed. No more digitizing of Polaroid photographs of HP or Tektronix screens.

Within a couple of years we developed MacScope I, which had an external box that acquired the data and shipped it over to the Macintosh. The external box had its own microprocessor which the Mac controlled. This system was very good at acquiring data down to the microvolt range, on time scales from weeks to milliseconds. Later we used the MacScope software with the Mac connected to a Hewlett Packard computer and could begin to grab data in the microsecond range. This is how we captured the microsecond scale data seen in the Physics2000 text.

There were two problems with the first MacScope. One was that our external hardware box sold for nearly $2000, and the equipment and interface became obsolete when the Mac went to USB.

The main change that led to MacScope II was the great improvement in the sound input of computers. Eight bit sound input appeared in the 1980s, but using that as an oscilloscope input gave very poor results. It is the 16 bit sound input that gives us the excellent results we can get with MacScope II.

Using the computer's sound input capability eliminates the need for expensive external equipment. The cost of external equipment now ranges from $0 to $50. The disadvantage is that we are limited to the audio frequency range of 10 Hz to 2200 Hz.

The MacScope program is the collaborative effort of myself and Chris Sweeney. Chris has stayed up on the internal workings of Mac and Windows computers, and writes sample basic C++ code to handle the fast operations. I expand on Chris's C++ code and then ask him what I did wrong. I then construct the user interface, all the controls and plotting using the program REALbasic. REALbasic immediately compiles for both Mac and Windows, and the people at REALbasic have been very helpful. Several years ago I wrote the field plotting program Charges2000 in order to learn REALBasic before applying that language to MacScope II.

E. R. Huggins
Dartmouth College

MACSCOPE II INSTRUCTION MANUAL

1) HARDWARE

MacScope® II is a software audio oscilloscope written to run on both Macintosh and Windows. The original MacScope, which we designed in the 1980s, worked only on Macs, and required a $2,000 external interface box. With MacScope II, we have considerably reduced external hardware costs, down to as low as $0.

Here we show different hardware inputs. The simplest to use is a built-in microphone input, if your computer has one. In that case no external hardware is required. For computers without a microphone input, you can use the Griffin iMic, shown in Figures 1.2 and 1.3, which converts a microphone signal to a USB output. In Figure 1.4 we show how to go from the BNC cable often found in labs, to the stereo input used by most computers and by the iMic.

Figure 1.1
Some computers have a built-in microphone input. If you use that to record sounds, no external hardware is required.

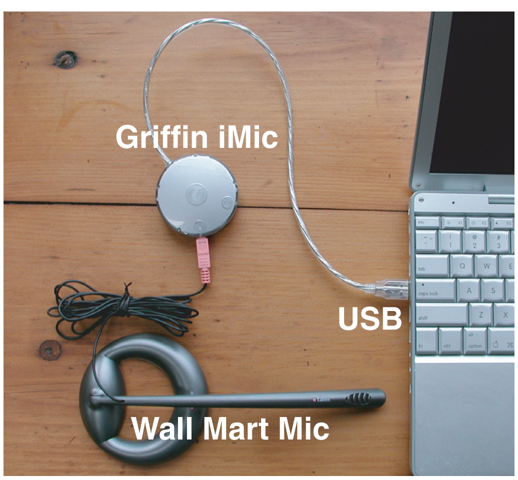


Figure 1.2
If your computer does not have an adequate sound input, you can use the $40 Griffin iMic that converts an audio voltage input to a USB output.

Because MacScope II uses the computer's audio input to grab experimental data, the frequency range is limited to the audio frequency range. Explicitly, computers and the iMic have an AC coupled input that has a low frequency cutoff around 10 Hz, and both digitize the voltage input at a rate of 44,000 points per second. Since the maximum frequency that can be detected requires at least 2 points per cycle, the upper frequency for these audio inputs is 22,000 Hz, which is called the Nyquist frequency. If you try to look for frequencies higher than that in the data, you get spurious results called aliasing.

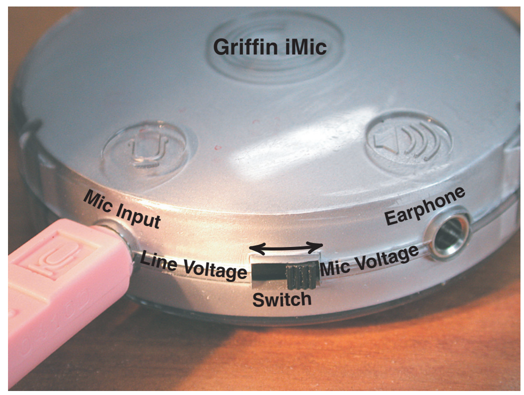


Figure 1.3
The Griffin iMic has an unlabeled switch that changes the input from **Line Voltage** (1 to 2 volt range) to **Mic Voltage** (millivolt range). Keep the input voltage under 2 volts to prevent damage to the iMic. Set the switch to Mic Voltage, as shown, for the greatest sensitivity. (iMic available at www.griffintechnology.com.)

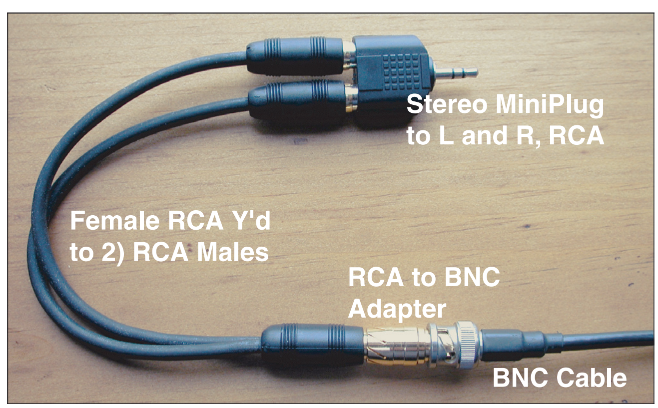


Figure 1.4
Inexpensive connectors that allow us to get a signal from a BNC cable into both sides of the stereo input. (Connectors from www.audiogear.com.)

2) THE MACSCOPE PROGRAM

The two icons associated with the MacScope II are shown in Figure 2.1. The first represents the application itself. The other represents Sound Data Files that are created when you save curves that MacScope has grabbed. You can open MacScope by clicking on either icon, or dropping a Sound Data file icon onto the application icon.

To obtain the Macscope display seen in Figure 2.2, we double clicked on the file named Sound Data - Piano Notes.sdf that is included with the Acrobat version of these notes. (The .sdf extension stands for "sound data file".) This file contains recordings of six different notes on the piano. The first, which we see in Figure 2.2, is middle c, recorded a few seconds after the piano key was struck. This is the note that was analyzed in detail in the accompanying article Teaching Fourier Analysis in Introductory Physics. It is the note that is lacking a seventh harmonic.

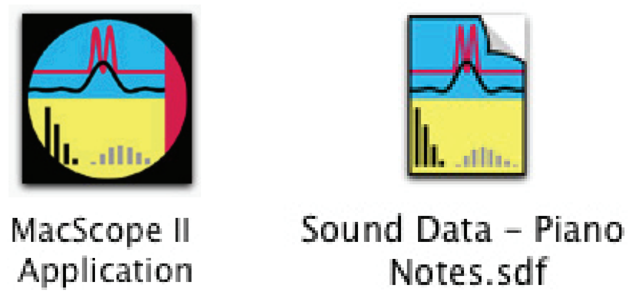


Figure 2.1
MacScope icons. If you click on the application icon, the program opens ready to take data. Clicking on a Sound Data File opens MacScope with all the saved files ready for analysis. It is also ready to take more data.

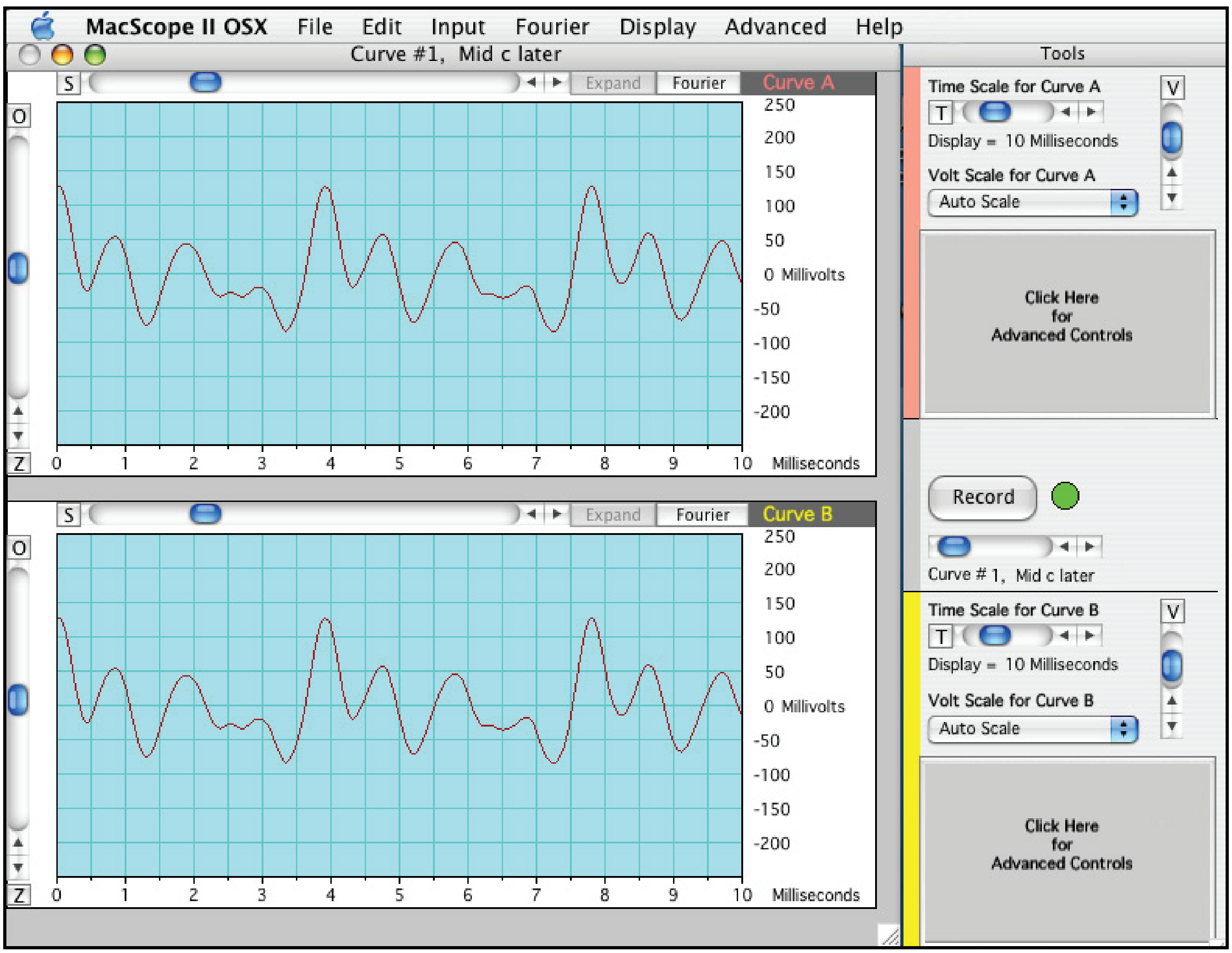


Figure 2.2
MacScope Data window and Tools window. The data, which had been saved in the Sound Data File, is a recording of middle c on a piano. Five other notes are also recorded in the same file.

3) SOUND INPUT

Before we start recording new data, we have to select one of the sound input devices discussed earlier in the hardware section. To make the selection, we go to the Input item in the MacScope menu, and select Sound Input Control as shown in Figure 3.1. What you get depends on the computer, and the computer system you are using.

We see the Sound Input Control panels for Mac OSX, and Mac OS(8 or 9) in Figure 3.2, and Windows in Figure 3.3. In all three panels we are given a choice of the available sound inputs. We chose the USB input, which is from the iMic shown in Figures 1.2 and 1.3.

Recording data with MacScope II is much like using a tape recorder. You should set the input gain or volume so that the input signal level is about 2/3s of the way up the level indicator bar as shown in the two Mac input control panels. If you set the gain too high, the tops and bottoms of your data will be clipped off. Set it too low and you loose sensitivity.

The arbitrary gain control built into computers affects the voltage scale in our plots. MacScope assumes that the maximum voltage range is ± 2 volts. But the actual range depends upon your input volume setting. (If you are using the iMic, the results depend greatly on where the line/mic voltage switch is set. Setting the switch to mic voltage amplifies the voltage by a factor of about 20.)

If the actual voltage values are not critical, and they are usually not for microphone data, then just consider the voltage scales as giving you the relative voltages. If the actual voltages are important, you can use the calibration method we will describe later.

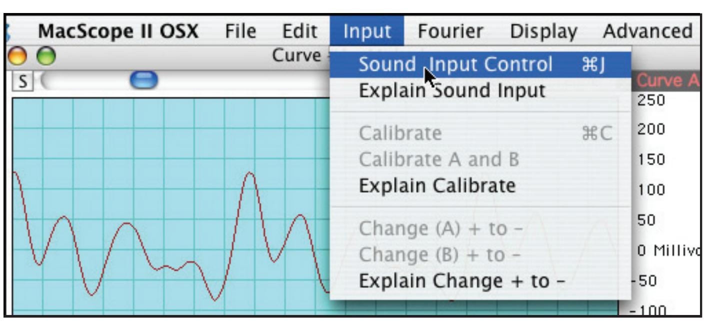


Figure 3.1
Selecting the Sound Input. This brings up the computer's sound input control panel.

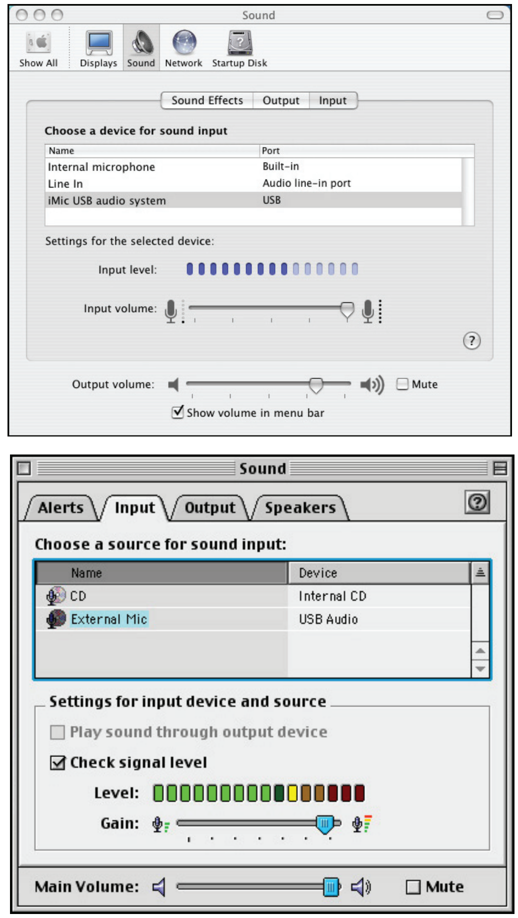


Figure 3.2
Macintosh sound input panels, OSX above, OS(8 or 9) below. We set the input volume or gain so that the level indicator was about 2/3s the way up.

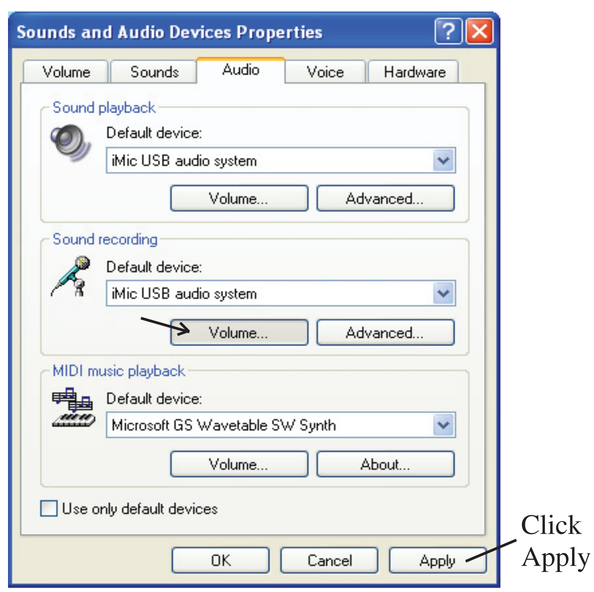


Figure 3.3
Windows GX sound input panel. Clicking on "Volume" in the "Sound Recording" section gave us the gain scroll bar seen in Figure 16.5a.

4) OBSERVING DATA

Once you have selected a sound input, simply go to the Tools window and press the Record button. If you have a repetitive input signal, you automatically get a stable picture of that signal. The reason for this steadiness is that the computer looks through a broad section of data to find the highest point (maximum voltage), and plots that point at (t = 0) on the plotting window. As a result, the computer plots the highest point of a repetitive curve at the same place each time and the curve looks stable.

Figure 4.1 shows the results of saying the vowel sound "oh" into a microphone attached to an iMic as shown in Figure 1.2. Because the microphone is sending almost the same signal to the two stereo inputs, the two plots, Curve A and Curve B are almost the same. On the right side, where a voltage scale will appear when we press Stop, we now see the value of the maximum voltage which was plotted at (t = 0). The voltage scale will be in millivolts, and the time scale is in milliseconds.

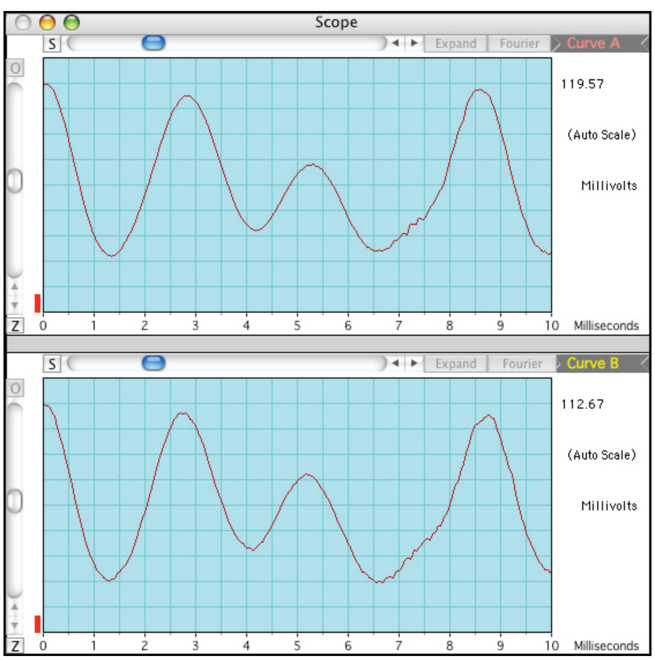


Figure 4.1
Live plot of the vowel sound "Oh". We get nearly the same plot in Curve A and Curve B because the microphone is sending nearly the same signal on both sides of the stereo input.

5) CHANGING TIME SCALES

In Figure 4.1, it is not clear whether the vowel sound "oh" is repetitive or not. We can find out by changing the time scale of the plots. We do this using the Time Scale scroll bars in the Tools Window.

In Figure 5.1, we have changed the Curve A time display to 50 milliseconds (ms) from its original setting of 10 ms. We changed the Curve B setting to 20 ms, and got the results seen in Figure 5.2. Now it is quite clear that the signal is repetitive. (Note that MacScope has the ability to display the same signal simultaneously on two different time scales.)

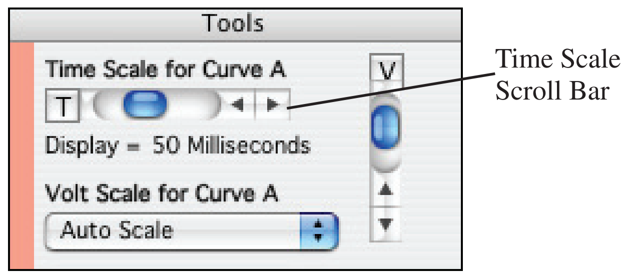


Figure 5.1
We used the time scale scroll bar to change the time scale on Curve A from 10 to 50 milliseconds.

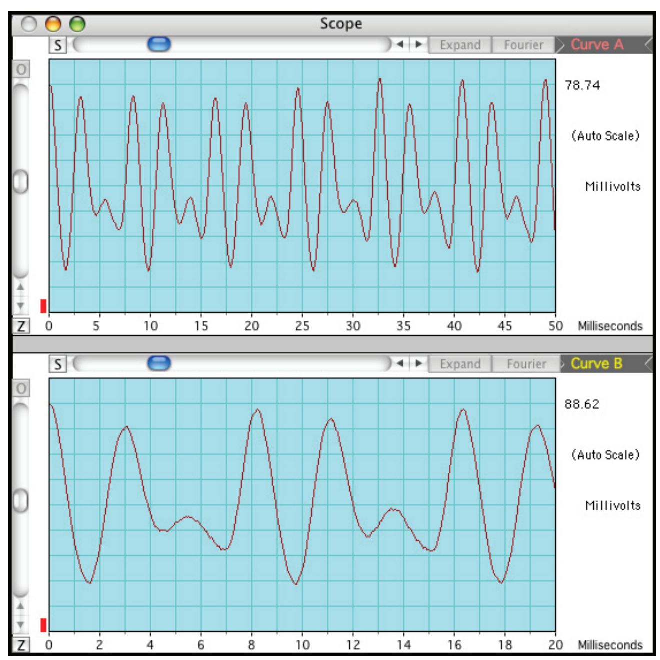


Figure 5.2
Live plot of the vowel sound "Oh" seen simultaneously at two different time scales.

6) VOLT SCALES

When MacScope is running, there are three modes for displaying the voltage scale. They are Auto Scale, Volt Scale-Tracking, and Volt Scale-Fixed, selected by the volt scale menu displayed in Figure 6.1.

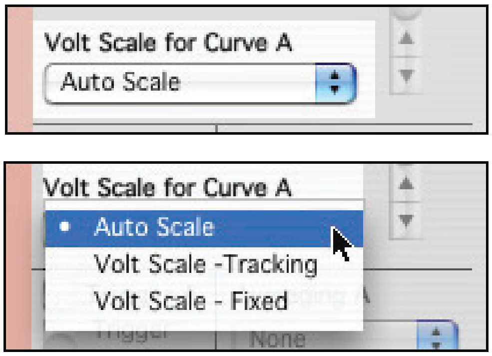


Figure 6.1
Selecting different kinds of voltage scales.

In Auto Scale, the t = 0 voltage value, which is usually the maximum recent voltage value, is plotted one square down from the top, and the corresponding voltage value is printed on the right side of the plot. This has the advantage of starting each successive curve at the same place on the screen which makes the curves look steady. What can continually change is the printed voltage value.

If you want to see a voltage scale, you can select Voltage Scale-Tracking. In this mode, MacScope looks at the t = 0 voltage value and prints the curve using an appropriate voltage scale. For example if the t = 0 voltage were 88 millivolts, a 100 mV voltage scale would be used. If the t = 0 voltage rose to 122 millivolts, MacScope would switch to a 250 mV voltage scale. The advantage of the Volt Scale-Tracking mode is that you always have an appropriate voltage scale. The disadvantage is that the voltage scale can keep changing, which may be annoying.

If MacScope is running and you select Volt Scale-Fixed, MacScope selects a volt scale appropriate for the current t = 0 voltage value and stays there. The volt scale no longer hops around, but the data can go off scale.

When you stop, MacScope always switches to a fixed volt scale that is appropriate for the data being displayed.

7) VOLTAGE SCROLL BAR

You can manually change voltage scales by using the voltage scroll bar shown in Figure 7.1. If MacScope is running in either Auto Scale or Volt Scale-Tracking mode and you make any adjustment to the voltage scroll bar, MacScope automatically switches to the Volt Scale-Fixed mode. This allows you to adjust the voltage scale while MacScope is running.

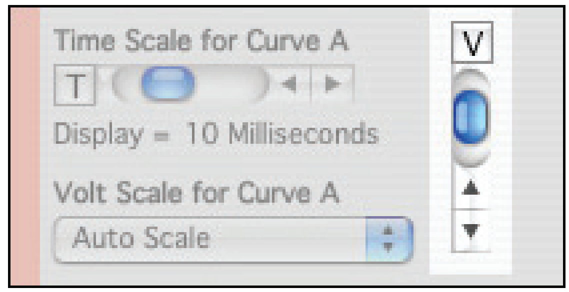


Figure 7.1
The **Voltage** scroll bar amplifies the height of the curve, up or down. When you change the setting of the **Voltage** scroll bar, you automatically get a **Fixed Volt Scale**.

8) THE (RECORD**) BUTTON

If you adjust the voltage scroll bar or make any other significant adjustment like changing the time scale, then when you stop recording, a new button labeled Record ** appears near the Record button. If you want to preserve these adjustments the next time you record, press the Record ** button instead of the Record button. The Record ** button also preserves any adjustments you make when MacScope is stopped, or any adjustments contained in a saved curve you have brought up.

The reason that there is both a Record and a Record** button is that it is easy to make adjustments that you do not want to see when recording again. When MacScope is stopped you may have amplified the voltage scale to look at a piece of low voltage data. If you preserve these settings when you record again, the voltages can be so far off scale that you cannot tell what has gone wrong.

When you press the regular Record button, you essentially do a factory reset to normal settings. The volt scale reverts to Auto Scale, the time scale reverts to 10 milliseconds, the starting time reverts to t = 0, and the voltage value V = 0 moves back to the center of the window. There are times when you want to preserve settings and times you do not. We give you the choice with these two buttons.

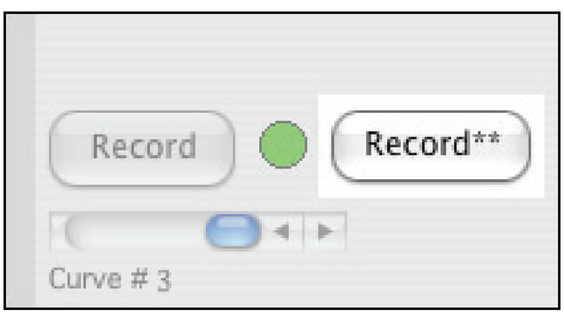


Figure 8.1
When you start recording using the **Record**** button, you preserve your current settings.

9A) TIME OFFSETS

Along the top of both Curve A and Curve B are horizontal scroll bars with a button labeled [T] on the left. These are Time Offset scroll bars that adjust where t = 0 is located. You adjust this scroll bar to look at earlier or later data, as shown in Figure 9-1.

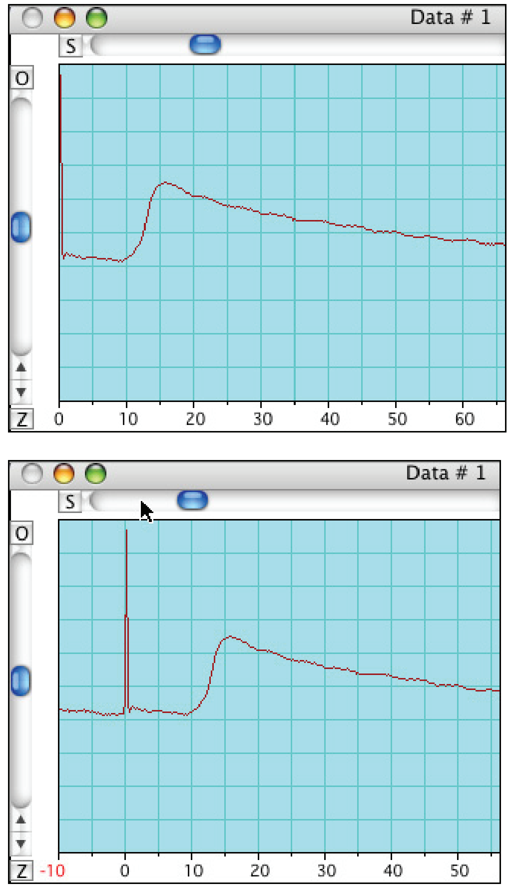


Figure 9.1
Two clicks in the **Time Offset** scroll bar moves t = 0 two squares over so that we can see what happened there.

If you want to get back to having t = 0 at the beginning of the curve, press the button labeled [S]. We call this a reset button. We put reset buttons on almost all our scroll bars in order to quickly get back to some preferred value.

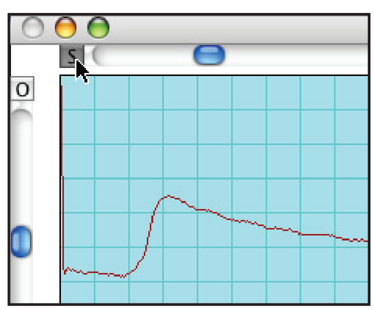


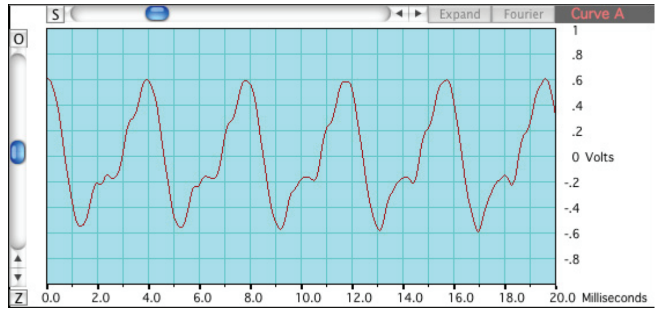
Figure 9.1c
Pressing the reset button moves t = 0 back to its normal position.

9B) VOLTAGE OFFSETS

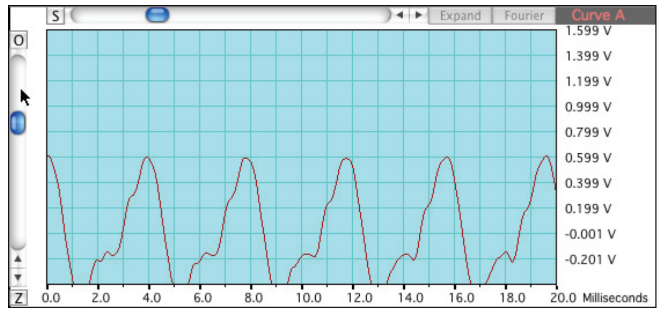
On the left side of each curve is a vertical scroll bar with an reset button labeled [O]. This is the Voltage Offset scroll bar that moves the curve up and down. You can move the curve up until just the bottom tip of the curve is showing as seen in Figure 9-2, or down until just the top tips are visible. The reset button puts the zero of the voltage scale back in the center.

If you make an adjustment in either of these scroll bars for either window, the Record ** button will allow you to preserve the adjustments, while Record will perform resets.

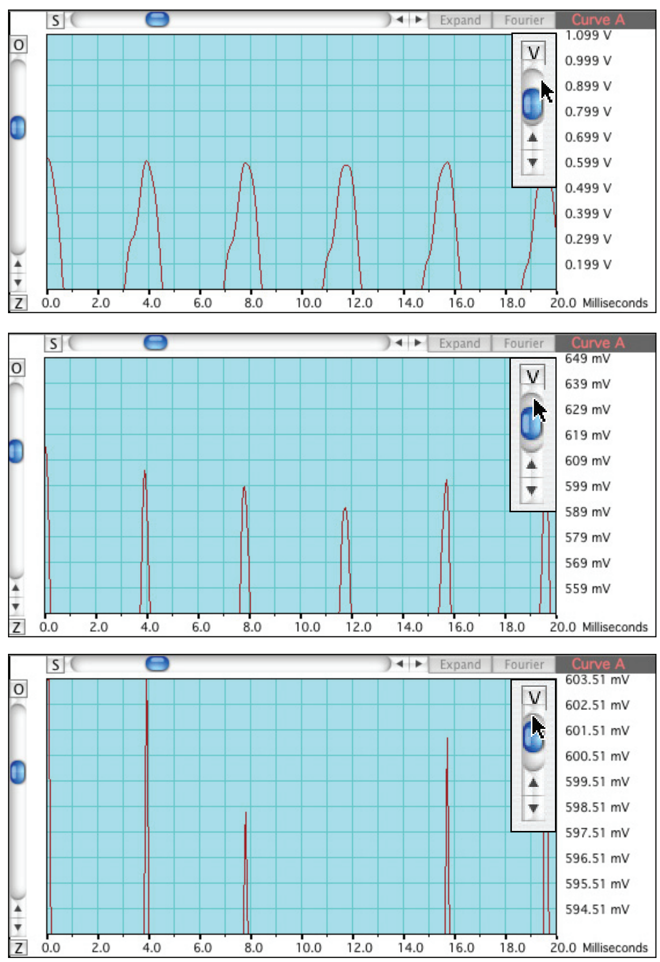
Figure 9.2
How to study curves in detail. Suppose that we want to look very closely at the tops of the middle c curve seen in A). The first thing we do is to use the **offset scroll bar** to move the curve down as shown in B), so that the tops of the curve are at the center of the plotting window. Then when we use the **voltage scroll bar** to amplify the curve, the tops will remain at the center of the window, and we can get an extremely detailed picture of the curve tops.



A) One of our middle c notes.



B) We used the voltage **offset scroll bar** to move the curve down so that the tops were at the center.



C) Then we used the voltage scroll bar to amplify the curve.

10) STOP AND STORE

When you press the Stop button, the current data (up to 177,000 points) is stored in RAM along with previously stored data. As seen in Figure 10.1, the name of the window has changed from Scope to Curve #7. It is Curve #7 because we got six data files when we opened MacScope by double clicking on the "Sound Data - Piano Notes.sdf" file shown in Figure 2.1. That sound data file (.sdf) already contained six files, so that our "Oh" vowel sound file becomes a seventh.

There are several ways to see what data files are stored and ready for viewing. One way, that gives you a complete summary of available files, is to go to Return To in the Edit Menu as shown in Figure 10.2.

To go rapidly between stored files, you can use the Store Select scroll bar underneath the Record button. In Figure 10.3, we started off with Curve #7. Clicking once in the left arrow brings up Curve #6, Octave above a. Moving the thumb all the way to the left brings up the first data file Curve #1 Mid c later.

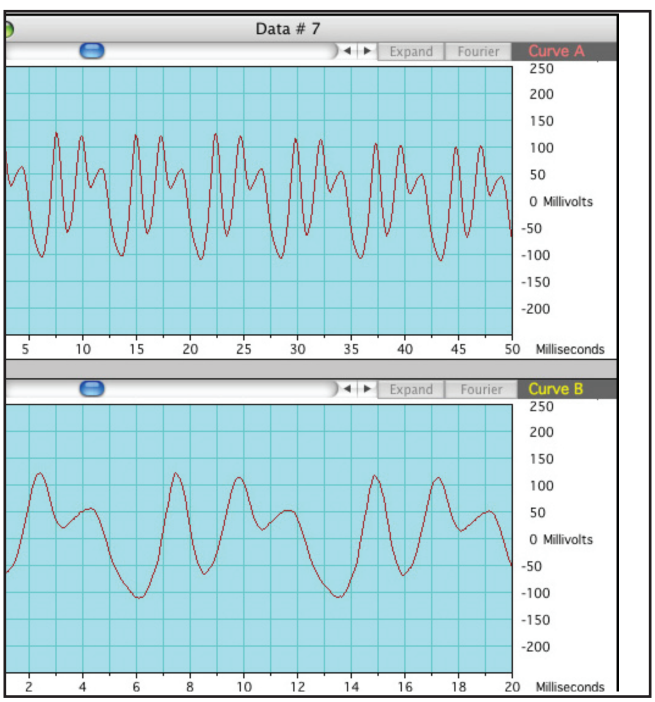


Figure 10.1
Stored plot of the vowel sound "Oh". When you press **Stop**, the most convenient near by volt scale appears, and the curve is adjusted to that scale. Each time you record and then press **Stop** a new curve is stored in RAM (but it is not yet saved to your hard drive).

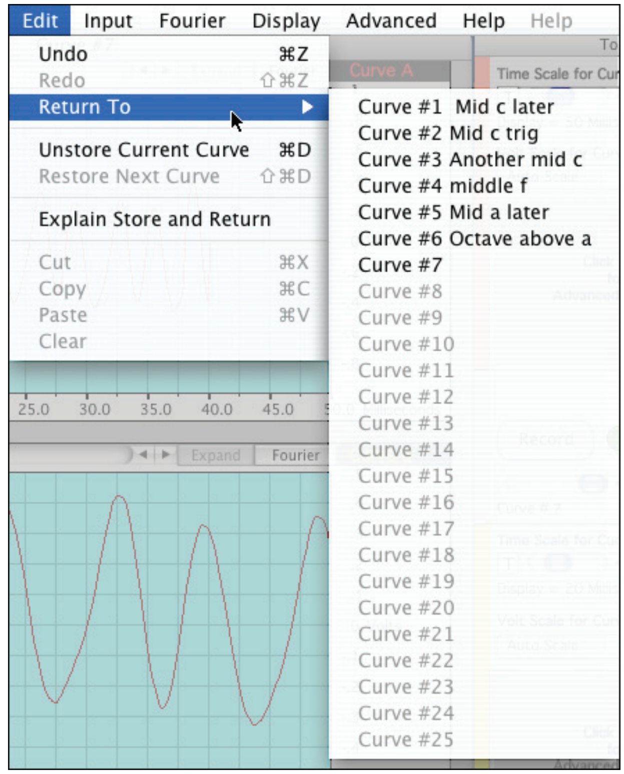


Figure 10.2
The **Return To** menu shows you all the stored files. Up to 25 files can be stored in RAM. If you store a 26th file, it replaces **Curve #1** etc. If you press **Stop** and do not like the results, you can get rid of that file by selecting **Unstore Current**. (**Restore Next** puts it back.) **Undo** takes you to the previous file, and **Redo** to the next file.

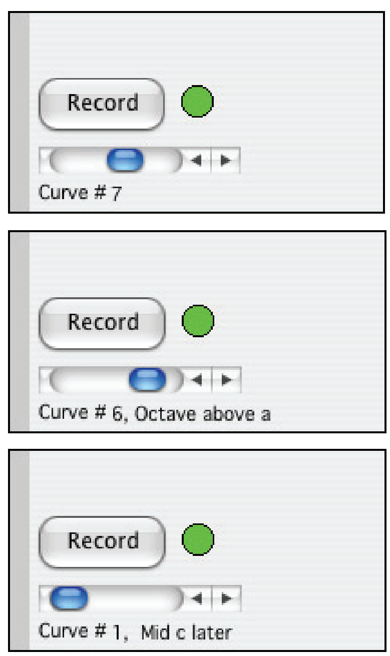


Figure 10.3
Using the **Store Select** scroll bar to rapidly go between stored files.

11) SAVING STORED FILES

The stored files are stored in RAM, the Random Access Memory that disappears when you shut your computer off. Worse yet, these files disappear when you Quit MacScope. You lose them unless you Save them on your hard drive, a CD, or some other storage media.

MacScope gives you the two options for saving data files seen in Figure 11.1. The option Save All Data As... in the File menu, saves all currently stored data files as a single file on your hard drive. The file Sound Data - Piano Notes.sdf that we double clicked back in section 2, was created this way. When you open such a file, the curves are added to any you may have currently stored. (For example, if we already had the seven files shown in Figure 6.2, and we used Open Data File(s) to open the piano notes file again, we would end up with 7 + 6 = 13 stored files.)

With Save Current Data in the File menu, you also have the option of saving just the currently displayed data file. This is useful if you have taken a lot of data, but only one or two of the files are worth saving.

The option Save Plot in the file menu merely saves a picture of your current data window. It is essentially a screen dump of that window. It is not nearly as useful as saving the data, because you cannot manipulate the data later on. If you are working in the lab, we recommend that you save data files, and create plots only when needed.

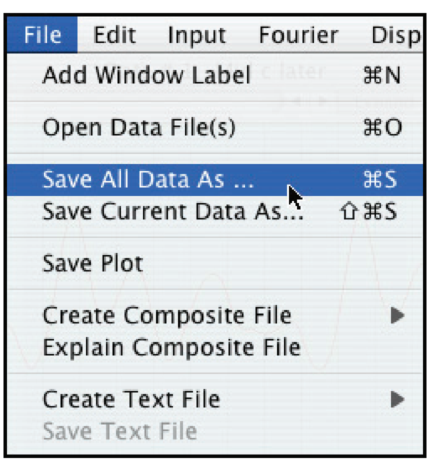


Figure 11.1
Using the **File** Menu to save data files.

12) NAMING STORED FILES

The name “Data #7” is not the ideal name for our vowel sound “Oh” data file. You can see in Figure 10.2 that the other data files have names like “Mid c later” and “Mid c trig” which tell us something about what is in the data file. To add a name to a data file, we use the Add Window Label command in the File menu.

When we make that choice, we get the window shown in Figure 12.1 that allows us to type in a name. After selecting Done, the new window name appears in three places, as the name of the data window (Figure 12.2a), under the store select scroll bar (Figure 12.2b), and in the Return to menu (Figure 12.2c).

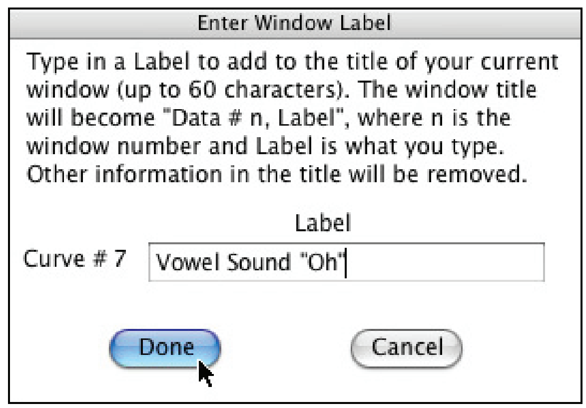


Figure 12.1
Entering a name for Data #7.

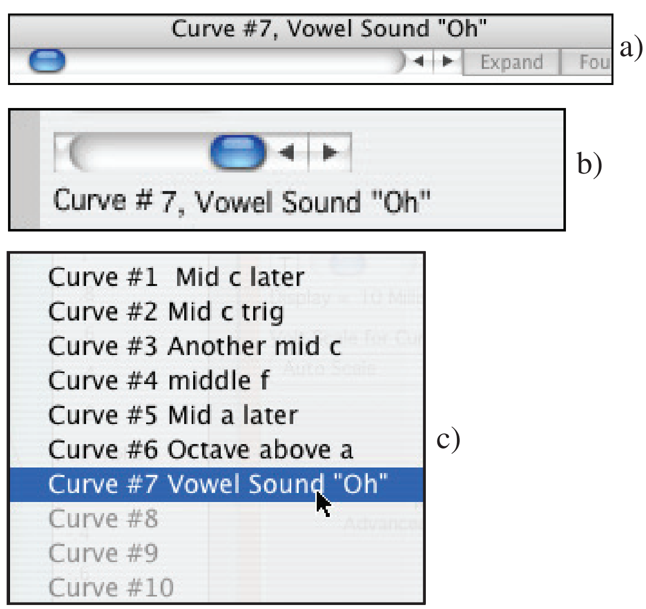


Figure 12.2
The new name appears in three places.

13) CREATING COMPOSITE FILES

At a workshop, one of the participants suggested that he should be able to compare files, by putting one file in Curve A, and a different file in Curve B. We have implemented this feature with the Create Composite File command in the File menu.

We will illustrate this capability by creating a composite file with "Data #1 Mid c later" in Curve A and "Data #2 Mid c trig" in Curve B. The result will become Data #8.

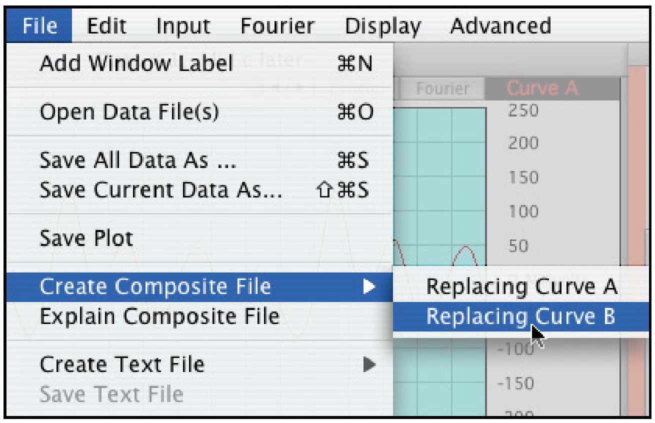


Figure 13.1
After getting the Curve A file we wanted, we choose to replace Curve B.

We begin constructing the composite file by first making "Data #1 Mid c later" the current file being displayed in the data window. Then we go to the File menu and select Create Composite File and then choose Replacing Curve B as seen in Figure 13.1.

That brings up the "Replace Curve B" window shown in Figure 13.2. We have moved the scroll bar so that we will be replacing Curve B with Data #2 of Curve A. The result is the composite curve shown in Figure 13.3. If we do not like the long name the computer gave the file, we can change it using the Add Window Label command in the File Menu.

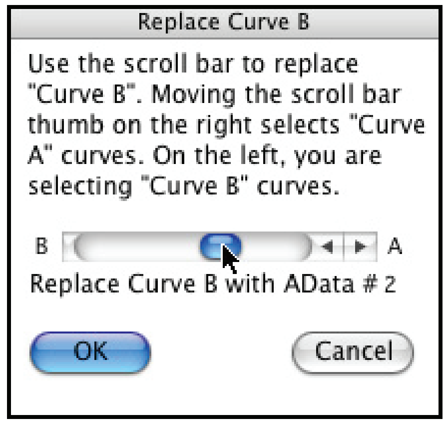


Figure 13.2
Choosing to replace Curve B with the file "A Data #2".

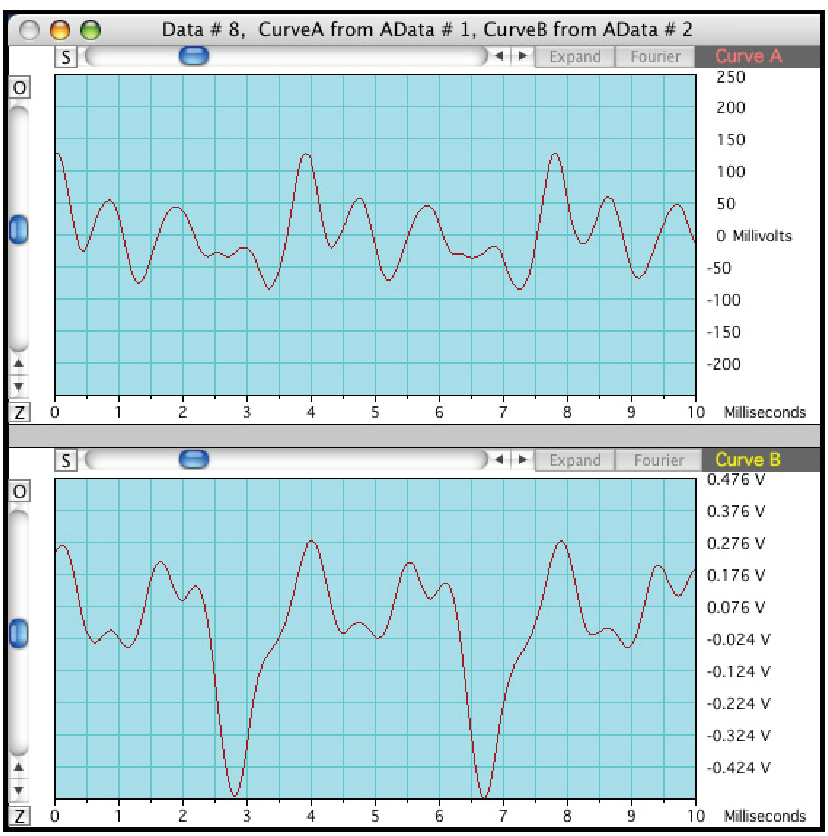


Figure 13.3
The resulting composite file, consisting of "Mid c later" in Curve A and "Mid c trig" in Curve B. This is the composite file we used in the accompanying article "**Teaching Fourier Analysis in Introductory Physics**".

14) SELECTING DATA

The main advantage of using the computer as an oscilloscope is that the data is immediately available for analysis. For example, the curve "Mid c later" which is Curve A of our composite file, is the middle c piano note recorded a short while after the key was pressed. Noting that the curve is repetitive, we can use the selection feature of MacScope to measure the period and frequency of the note.

In Figure 14.1, we are selecting one cycle of the curve by dragging the cursor over the curve. As we make the selection, a new window appears, a window labeled "Curve A values" seen in Figure 14.2. This window tells us that the period t of the selected data is t = 3.9 milliseconds (ms), and the corresponding frequency f = 1/t is f = 256.7 cycles per second (Hz).

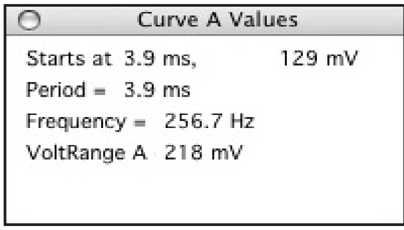


Figure 14.2
Data window for Curve A

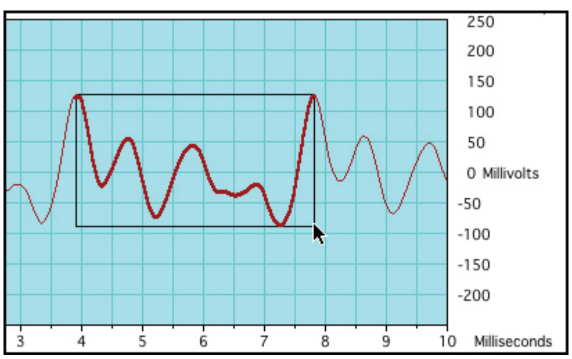


Figure 14.1
Selecting a section of the curve.

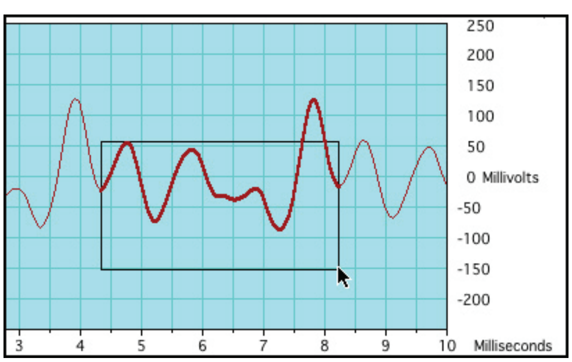


Figure 14.3
Hold down the shift key while dragging and the entire selection rectangle moves with the cursor.

Checking on the web, we found that the middle c frequency should be 262 Hz, which means that our piano is slightly out of tune.

To help position the selection rectangle, we have the feature that if you hold down the shift key while you move the cursor, the selection rectangle moves as a unit with the cursor, as shown in Figure 14.3. This helps you to precisely position the starting position of the selection rectangle. You can then release the shift key and adjust the ending point of the selection rectangle.

Once you release the mouse button, the selected section of the curve is highlighted as shown in Figure 14.4. If you press the button labeled Expand, as we are doing in Figure 14.4, the selected section of curve expands to fill the entire window, as seen in Figure 14.5. This gives us a detailed view of the selected section of data.

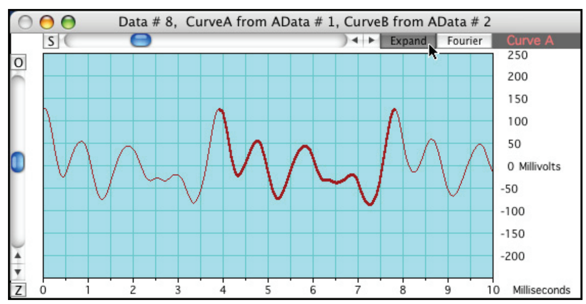


Figure 14.4
The selected section of data is highlighted. Here we are pressing the **Expand** button for a closer look.

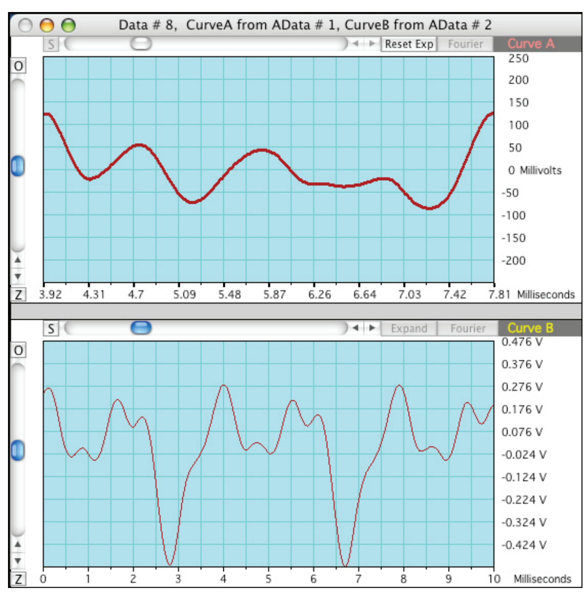


Figure 14.5
Selected data expands to fill window.

15) FOURIER ANALYSIS

One of our main aims in writing the MacScope program was to make it easy to do Fourier analysis of experimental data. The accompanying article "Teaching Fourier Analysis in Introductory Physics" contains a description of not only how to use MacScope to do Fourier analysis, it also shows the mathematics used by MacScope to do the analysis. Here we will briefly summarize some of the results of that article.

To get Figure 15.1, we pressed the Fourier button instead of the Expand button. Curve A data expanded as it did in Figure 14.5, but now the Curve B window is covered by a graph of the harmonics contained in the expanded data.

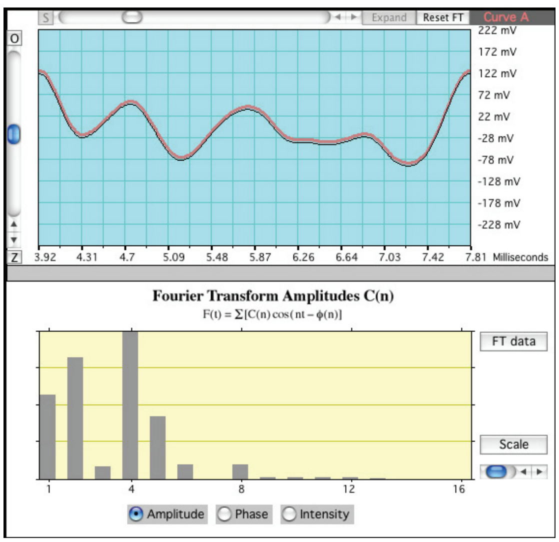


Figure 15.1
Harmonics contained in the selected data.

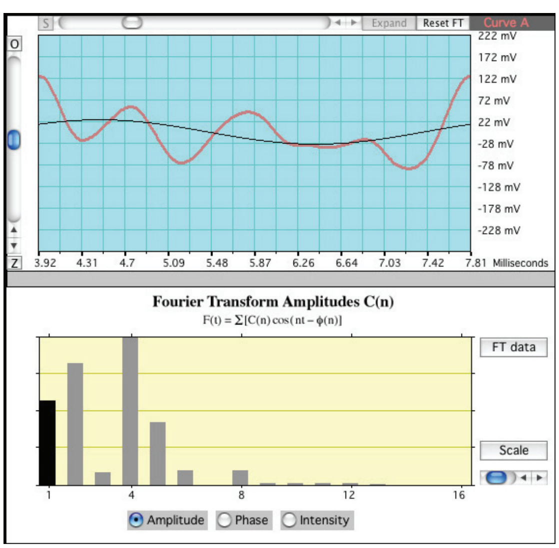


Figure 15.2
First harmonic superimposed on the data.

In Figure 15.2, we clicked on the bar representing the first harmonic, and we see that harmonic superimposed on our expanded data. In Figure 15.3, we selected both the first and second harmonics and see the sum of these two harmonics superimposed on the expanded data. This sum has a few of the general features of the data. When we select harmonics 1, 2, and 4, the sum is quite close to the data.

To choose Fourier analysis, you must first select a section of the experimental data. We made this requirement because the mathematics makes the assumption that the selected section of data repeats indefinitely. If by accident you select data that does not repeat, spurious harmonics appear as shown in Figure 10 of the Teaching Fourier Analysis article.

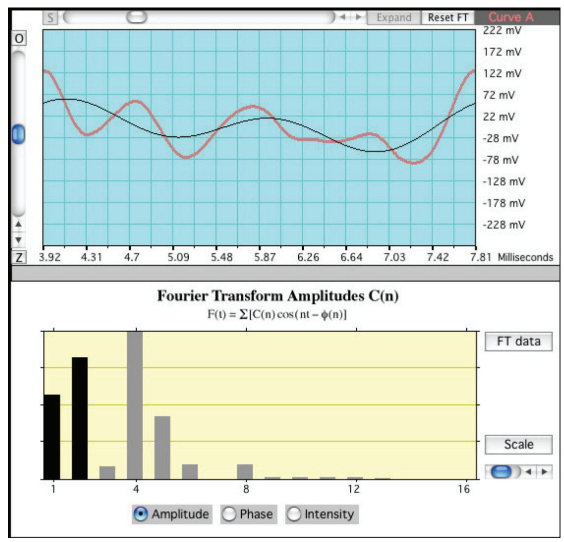


Figure 15.3
Sum of the first and second harmonics.

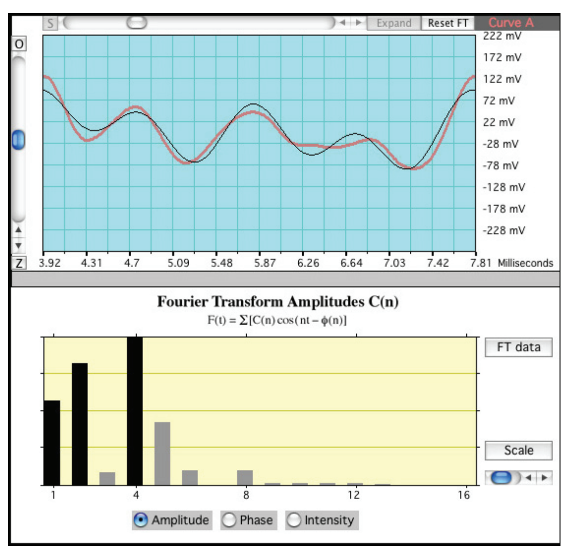


Figure 15.4
We get close adding the three biggest harmonics.

Pulse Fourier Transform

A unique feature we included in MacScope is the Pulse Fourier Transform that allows us to study the harmonics contained in a pulse. Here we show the harmonics in a short one cycle pulse. The reason for studying the harmonic structure of a pulse is that the analysis leads directly to an understanding of the energy-time form of the uncertainty principle, as described in the accompanying article "Teaching the Uncertainty Principle in Introductory Physics."

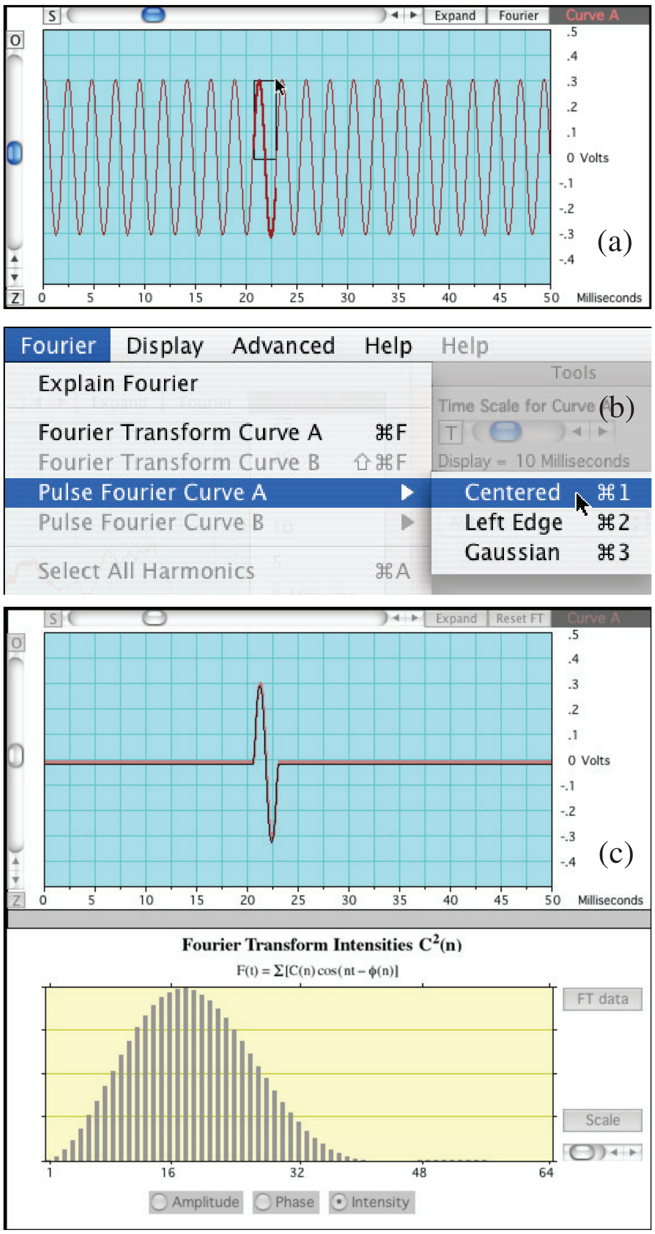
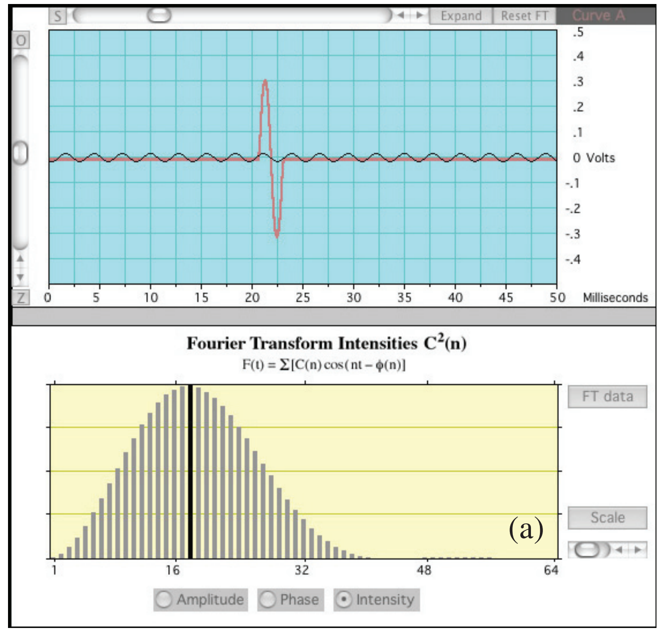
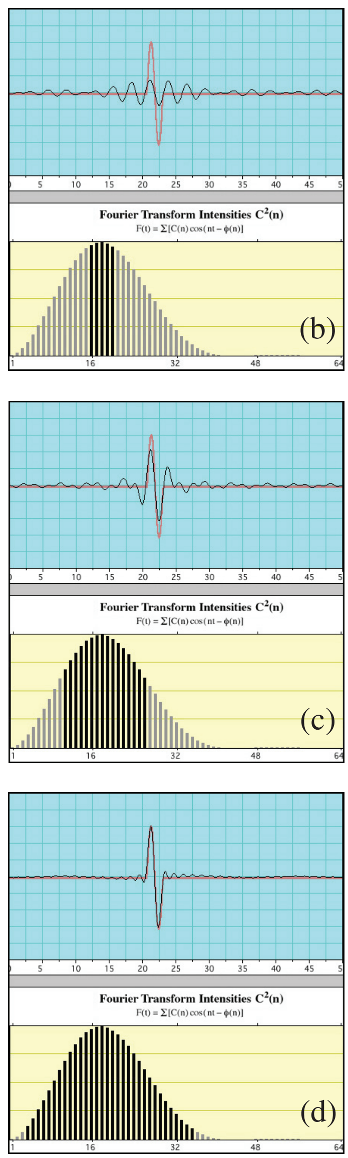


Figure 15.5 Creating a pulse
We can create a one cycle pulse by selecting one cycle of a sine wave as shown in (a), and then choosing the **Pulse Fourier Transform** from the file menu as in (b). In (c), we see that a pulse contains many harmonics.





Figs. 15.6. Fourier analysis of a pulse.
Here we see how a short pulse is constructed from long sinusoidal waves. In (a) we selected the largest harmonic and all it represents is a small sine wave. When we add together the five biggest harmonics in (b), a pulse begins to form. When we add up the 32 biggest harmonics, we get a close representation of the pulse in (d). We need a lot of harmonics to cancel the wave outside the pulse.

Laboratory Work

16) VOLTAGE RANGE AND VOLTAGE SCALES

When we are using MacScope for studying sounds, using either a Wal-Mart type microphone or a computer built-in microphone, we are not likely to damage equipment and we do not really care about the voltage scales on the MacScope windows. This changes, however, when we start plugging other laboratory equipment into MacScope. Then we have to be careful about the voltages we are using, and to get accurate voltage scales we have to calibrate MacScope.

Voltage Range

In the past, a standard calibration signal, produced either by an oscilloscope or signal generator, was a 5 volt amplitude square wave. If you put that signal into an iMic, you could destroy it (we have done that). Perhaps this could also destroy the sound input of a computer (we have not done that). The reason is that the iMic and computer inputs are designed to work in the 0 to 2 volt range, and larger voltages can be damaging. If you wish to study a signal whose voltage range is over a volt, you should use something like the voltage divider circuit shown in Fig. 15.1 to bring the voltage down to about the 100 millivolt (.1 volt) range. Using that circuit, the potentially damaging 5 volt signal is reduced to .05 volts, or 50 millivolts, which is in the ideal range for the computer inputs.

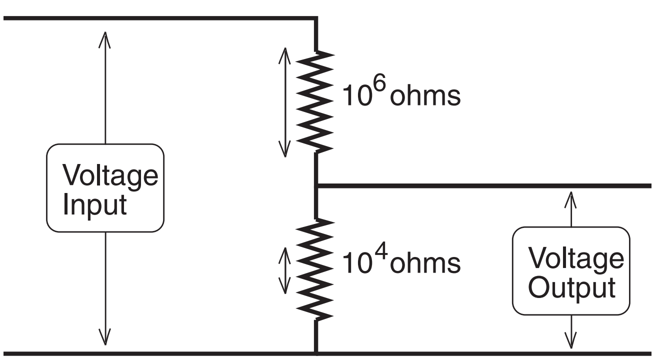


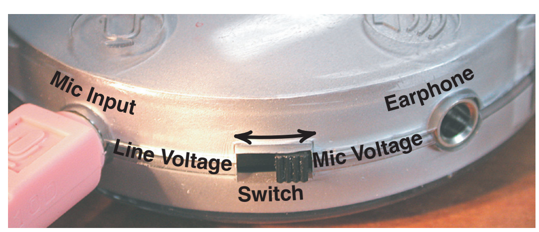
Figure 16.1
Voltage divider circuit. If your input voltage exceeds about one volt in amplitude, you should use a voltage divider to protect your equipment. With the above circuit, the output voltage is reduced by a factor of one hundred. If you replace the 10^4 ohm resistor by a 10^3 ohm resistor, the voltage is reduced by a factor of one thousand.

Voltage Scale

The voltage scale you have been seeing on the MacScope windows is based on the assumption that the full scale input into the computer is in the zero to two volt amplitude range. That is what the computer manuals say, but it is not true in practice because various stages of amplification are inserted before the data gets to the computer.

To see what the different input devices did to our voltage signals, we took an inexpensive sine wave generator, attached it to a standard calibrated oscilloscope, and adjusted the signal to an amplitude of 100 millivolts. This gave us a calibrated 100 mV sine wave signal that we plugged into six different kinds of MacScope inputs. We got four different results which demonstrates the necessity of using MacScope's calibration feature if the voltage scale is important.

Computer sound inputs can be divided into two general categories, line input and microphone input. The iMic, shown in Fig. 1.3 has a switch on the front that switches between these two inputs. The Macintosh G4 computer we are using has an external microphone input, while our PowerBook has an external line input. All these inputs give different results. The only thing consistent was that the iMic values were the same for both Mac and Windows when the Mac amplification was turned all the way down. (More about Mac amplification later.)



From Figure 1.3
Switch on the iMic.

The simplest way to see what these different inputs are doing is to look at what they do to our 100 mV signal.

Input *(100 mV)*	Uncalibrated Voltage Amplitude
iMic microphone input	510 mV
iMic line input	27 mV
G4 external microphone	128 mV
PowerBook line in	15 mV

From this you can see that there is a general trend that microphone inputs amplify the input signal while line inputs reduce the signal. But there is clearly no standardization.

What we have done in the MacScope software is add a calibration routine that corrects the input signal so that the voltage scale reads 100 mV when we put in a 100 mV signal. It can reduce the 520 mV signal or amplify the 15 mV signal to 100 mV.

We get the calibration window by going to the input menu and selecting Calibrate as shown in Fig. 16.2 If you want the same calibration for both curve A and curve B, then select Calibrate as shown. If we want separate calibration factors for the two windows, use Calibrate A and B.

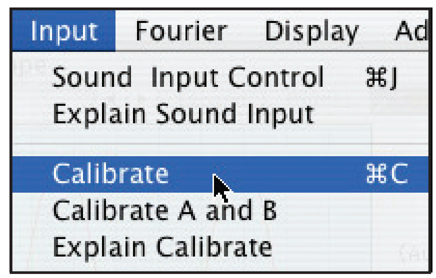


Figure 16.2
Selecting **Calibrate**.

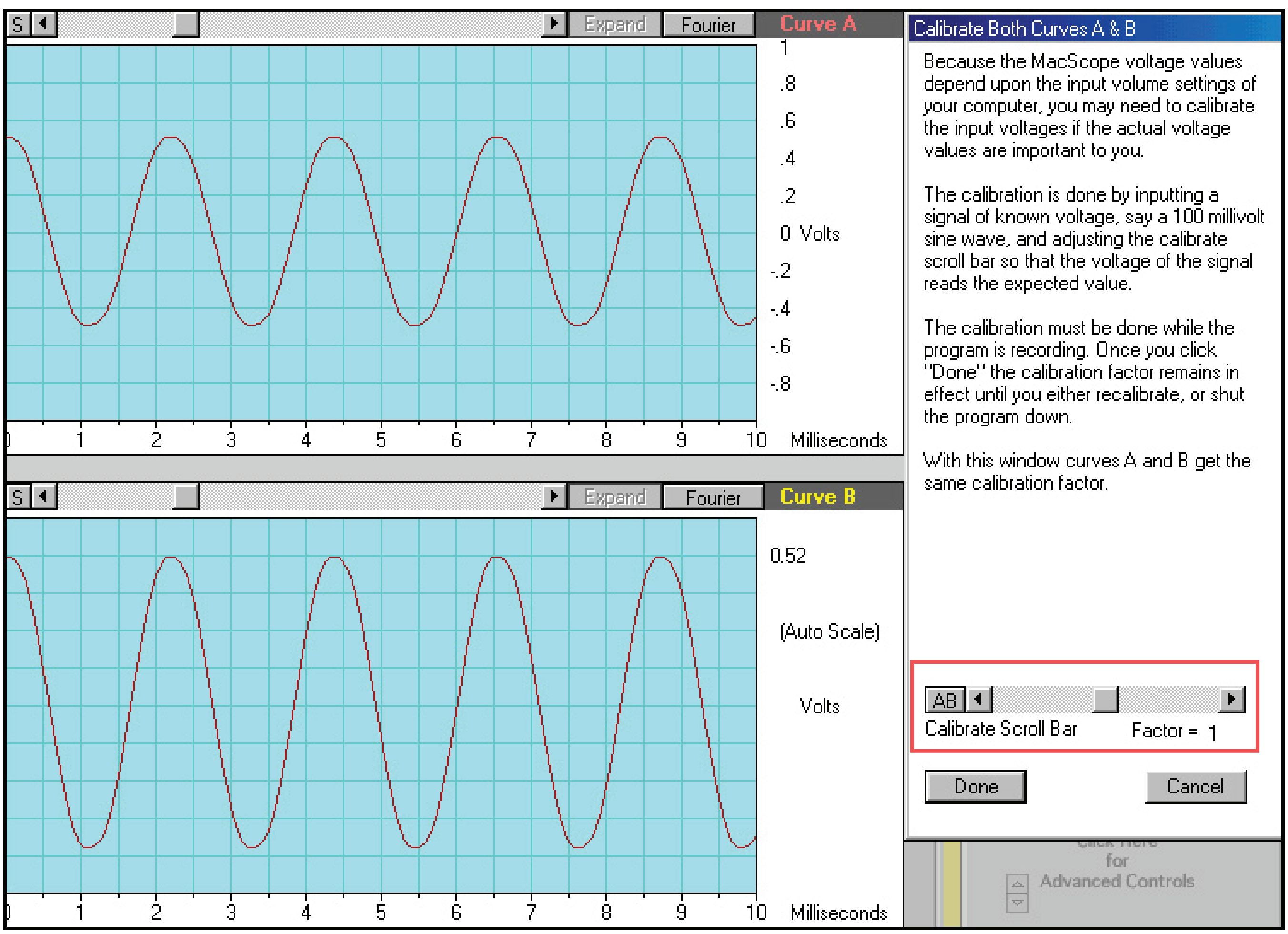


Figure 16.3
Uncalibrated result of a 100 millivolt signal into the iMic microphone input using a Windows computer.

The calibration routine works while MacScope is running. You change the scroll bar in the calibration window until the voltage scale in the scope window has the expected value. For example, in Fig. 16.3 we have put our 100 millivolt sine wave into an iMic attached to a Windows computer and set the line/microphone switch on the iMic to the microphone setting. We see that the voltage scale reads .51 volts (510 millivolts), a factor of 5.1 times too high. In Fig. 16.4, we moved the calibrate scroll bar to the left to get an amplification factor of .199 which reduces the voltage reading to the desired 100 millivolts.

When we calibrate on the Macintosh, there is an extra amplification to consider. As seen in Fig. 3.2, the Mac sound input window itself has an amplification scroll bar that is called Level on Mac OS9 and Input Volume on OSX. If the thumbs of these scroll bars are set all the way to the left, and we send our 100 millivolt signal in through an iMic, we get the same values on the Mac as we got on Windows.

Our Windows 98 machine has no Input Volume scroll bar but the Windows GX machine does. To get at the GX scroll bar, shown in Figure 16.5a, click on "Volume" in the sound recording section of the Audio Input Control Panel.

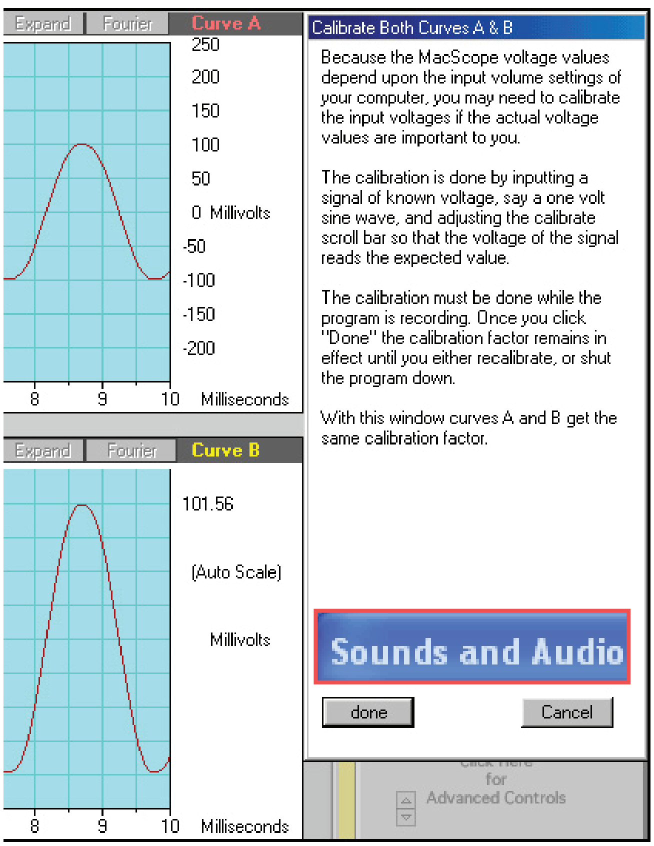


Figure 16.4
Reducing the amplification by a factor of 0.199 gives us the correct reading of 100 millivolts.

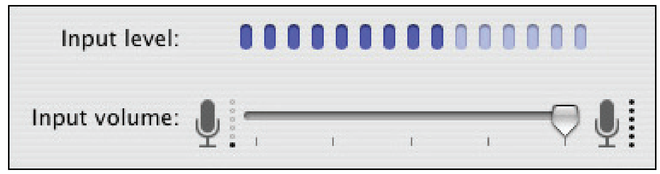


Figure 16.5
The **Input Volume** scroll bar on the Mac OSX sound input window.

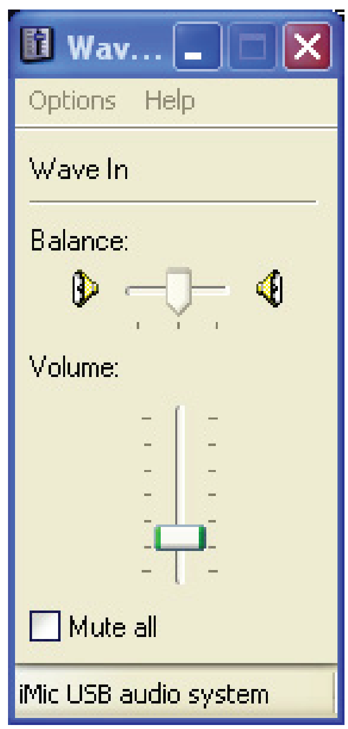


Figure 16.5a
The **Input Volume** scroll bar obtained by clicking on "volume" in the Windows XP sound input window.

When using the Mac, it would be convenient if the range of the Mac amplifier were great enough so that we did not have to use the MacScope Calibrate control. In one case we did this by using the iMic set to the line input and using the Mac amplifier to boost the voltage from 27 mV up to the desired 100 mV. The result is seen in Figure 16.6. However, the MacScope calibrate scroll bar is more sensitive than the Mac one, and you may want to use MacScope's calibration anyway.

When we sent our 100 mV signal directly into our G4 External Microphone Input in order to avoid the iMic, this left us with a 128 mV reading at the lowest Mac amplifier setting. To get the voltage down to the desired 100 mV, we would then have to have used the MacScope Calibrate feature.

17) SIGNAL AVERAGING

In real world laboratory situations, the data we deal with can be inherently noisy, either due to the nature of the object being studied, or to ambient electrical noise in the lab. This is particularly true of biological experiments where the phenomenon being studied continually varies from measurement to measurement, and it is impossible to completely shield the apparatus from outside noise like the 60 cycle radiation generated by the electrical wiring in the building. The cure for both of these problems is signal averaging.

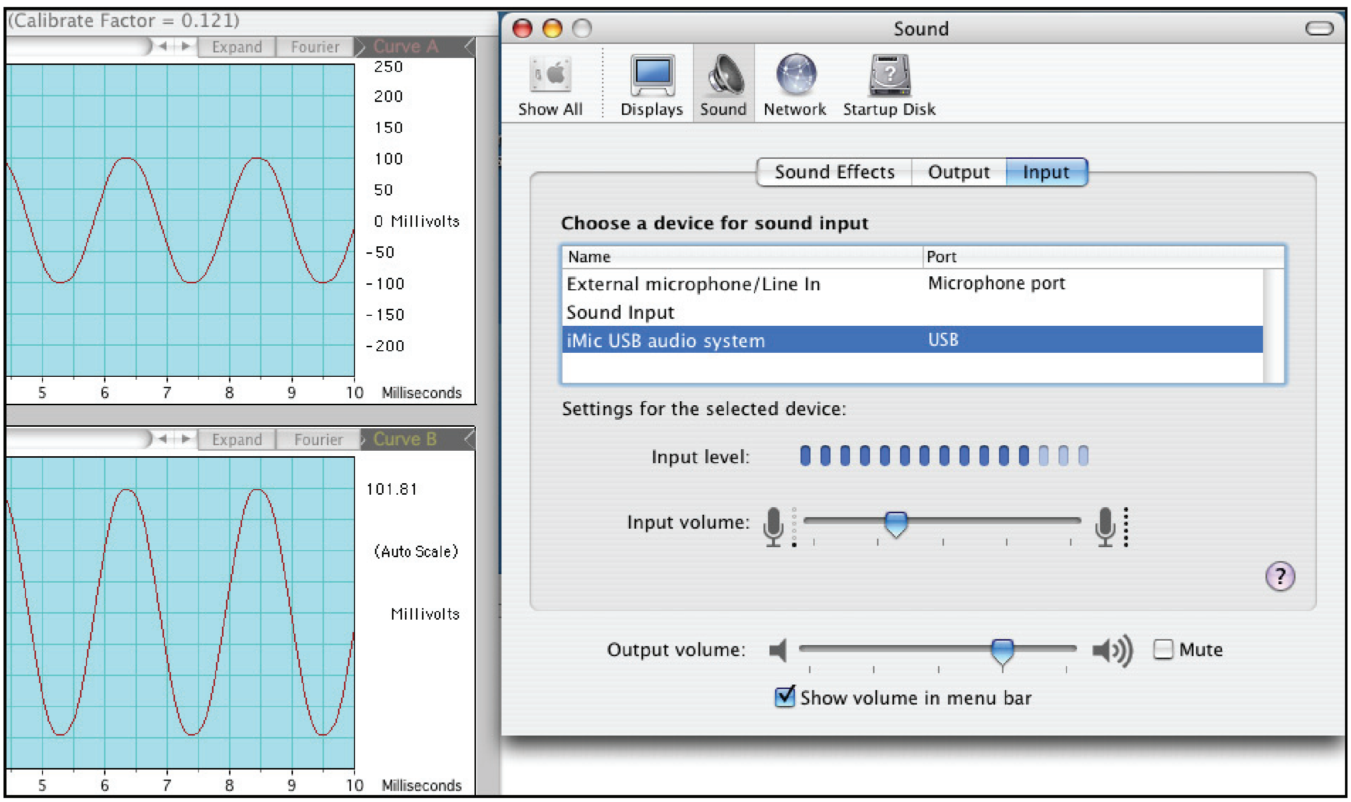


Figure 16.6
Using the Macintosh Input Volume amplifier to calibrate MacScope. Because this amplifier only amplifies, we could not use it to reduce the 128 millivolts we got when our 100 mV signal was plugged into our G4 Sound In port. Here we amplified the 27 mV signal from the iMic Line input.

Synapse Junction Potential

We will illustrate the effectiveness of signal averaging with the measurement of the synaptic potential produced in a muscle fiber when an action potential pulse arrives from an attached neuron. (See Figure 17.1.) The results are shown in Fig. 17.2. The spike on the left is the action potential pulse. Then there is about a ten millisecond delay as chemicals cross the synaptic gap between the neuron and the muscle fiber. After crossing the gap, the chemicals cause the depolarizing potential which is seen as the fairly rapid voltage rise followed by a relatively slow decay.

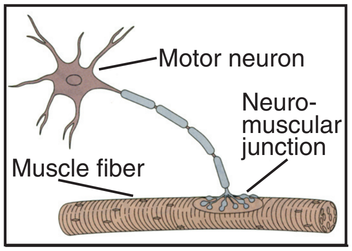


Figure 17.1
Neuron attached to a muscle fiber. (Figure adapted from **Essentials of Neural Science Behavior** by Kandel, Schwartz and Jessell**.**

Curve A in Figure 17.2 is a signal averaged curve. Before signal averaging, the data looked like that shown in Curve B. In this figure, Curve A is the average of 152 sections of raw data.

The way the experiment is done is that an electronic device called a stimulator is attached to the neuron and for this experiment sends out a voltage pulse every tenth of a second (every 100 milliseconds). This stimulator pulse causes the neuron to send an action potential pulse to the neuron-muscle fiber synapse. As we can see in Fig. 17.2, the neuron action potential pulse arrives every 100 milliseconds.

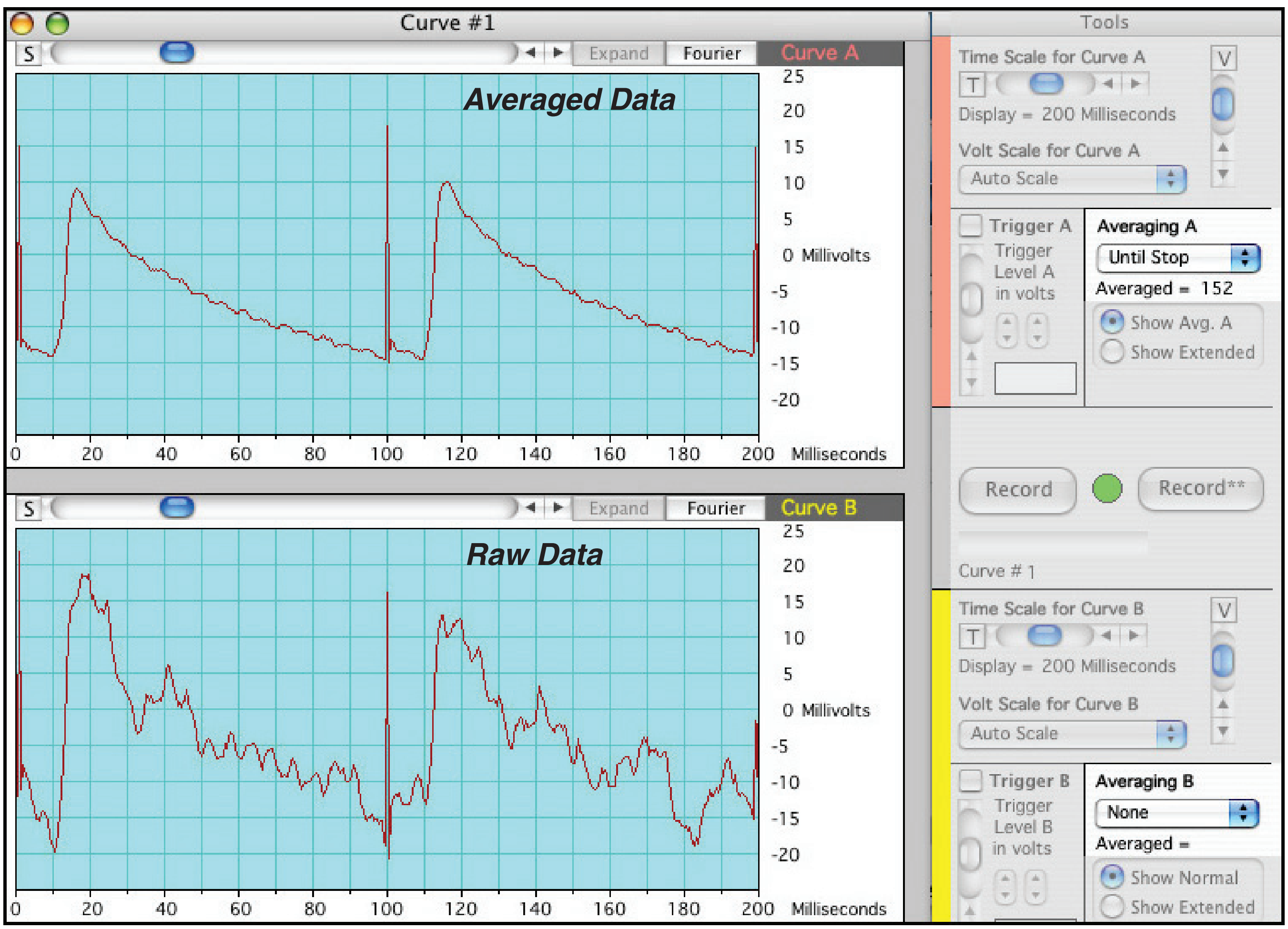


Figure 17.2
MacScope can simultaneously display the signal averaged data (Curve A) and the raw data that is being averaged (Curve B). In the **Tools Window** averaging section, for Curve A we selected **Average Until Stop**, while for Curve B averaging we selected **None**.

A special feature of MacScope is that if you send the same raw data into Curve A and Curve B, you can simultaneously observe signal averaged data and raw data as we did in Fig. 17.2. In the Tools Window, remove the covers over the advanced controls (by clicking on them) and you see the signal averaging control underneath. For curve A we selected Averaging Until Stop. That means that every cycle is included in the average until we press the Stop Button. For curve B we selected None for signal averaging. With that setup we can watch the signal averaged curve smooth out while simultaneously checking that the instantaneous data is still valid.

Instead of averaging Until Stop, you can select to average a given number of curves, as shown in Figure 17.3. For example, if you select 10 curves, then the display will show the average of essentially the last 10 curves recorded. This allows you to observe smoothed data that is slowly changing. The more curves you average, the slower the changes will be.

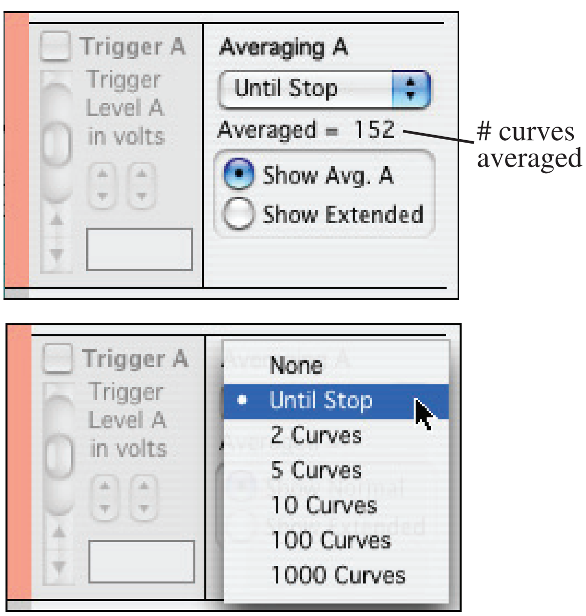


Figure 17.3
Selecting **Signal Averaging**. You can choose to average all the data until you press the **Stop** button, or you can choose to average the latest few curves. If you choose **5 Curves** then an average of the last 5 curves will be displayed in the chosen window. This allows you to watch averaged data that changes slowly. The **Averaged =** tells you the number of curves that have been averaged. Here we see that 152 curves are in the average.

18) SHOW EXTENDED

Once you remove the cover over the advanced controls, three new features appear. One is Signal Averaging which we have just discussed. Another is the Trigger Control that we will discuss in the next section. Here we will describe the Show Extended feature.

When recording, MacScope maintains two separate banks of data. One bank is the raw data which is 177,000 points long and is constantly refilled as sound data comes into the computer. The second data bank, 2,200 points long, holds the current signal averaged data. During recording, MacScope displays from the shorter signal averaged bank of data. When you stop, you can look at the much larger raw data bank by clicking on Show Extended. This allows you to scroll through all 177,000 points, which is just over 400 windows of data. (If you do not select signal averaging, then the signal average data bank simply holds the last 2200 points of raw data.)

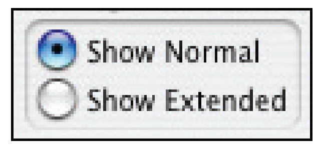


Figure 18.1

When you scroll through the extended data, the time scale is often not useful. To get Fig. 18.2, we started with the smoothed curve A of Fig. 17.2 and clicked on Show Extended to see the same raw data as being displayed in curve B. Then we scrolled back 32 seconds to see the data 160 windows earlier.

If we are interested in the structure of this earlier data, having the time scale start at –32.2164 seconds and stops just beyond –32.0564 seconds, is not particularly convenient. You can zero the time scale and have it go back to milliseconds, by pressing the little button labeled "**Z**" for zero, as shown in Fig. 18.3. Now you can easily see that the cycles repeat at 100 millisecond intervals.

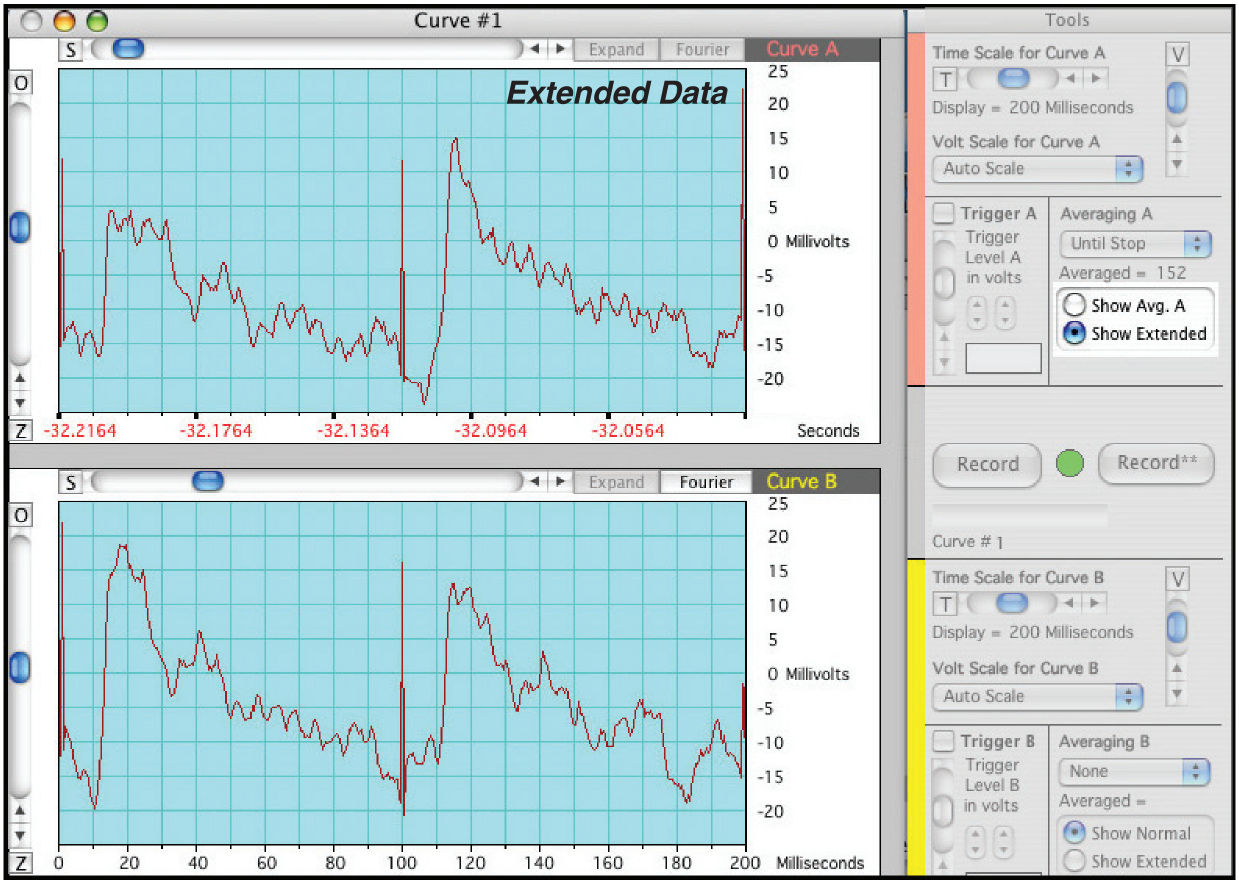


Figure 18.2
Showing extended data in **Curve A**. When we click on the **Show Extended** button, we have access to 400 windows of data, representing the latest unaveraged data. Here we have scrolled back to 32 seconds earlier than the data being displayed in **Curve B**.

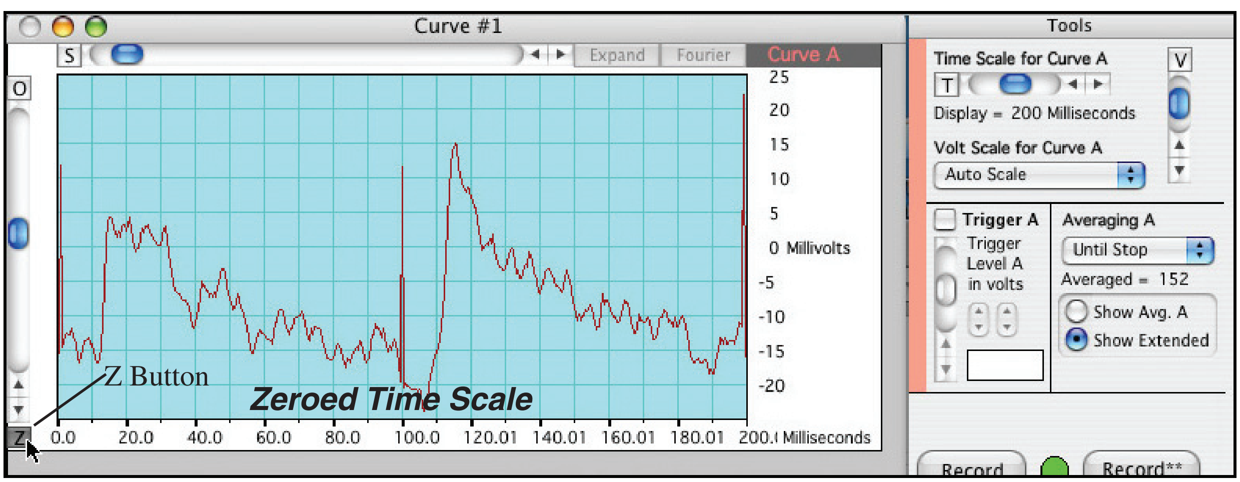


Figure 18.3
Zeroing the time scale for **Curve A**. When you press the **Z button**, you temporarily set the time scale to start at zero. That makes it much easier to see that the spikes are still 100 microseconds apart.

19) AUTOMATIC TRIGGERING

The most convenient feature of MacScope is that in most circumstances it is automatically triggered. The data from the synapse potential experiment is an excellent example of data that is well handled by automatic triggering.

In Fig. 19.1 we are taking a close look at the raw data from one cycle of that experiment. From this picture we clearly see that the sharp neuron action potential spike on the left has the highest positive voltage in the cycle. To draw the curve on the screen, the computer looks for the highest voltage in the next block of data and displays the highest voltage at the t = 0 position on the screen. As a result for our synapse data the action potential spike will be located at t = 0.

Sometimes you run across a nerve where the action potential spike is not as high as the resulting muscle voltage. If this happens then the noisy bump at point (a) in Fig. 19.1 would be the highest voltage, and the computer would move the curve over so that point (a) was located at t = 0. If sometimes the action potential spike was higher than point (a), and sometimes lower, then the curve would flip back and forth and we would say that the curve was not well triggered. This can be remedied using the manual triggering to be discussed shortly.

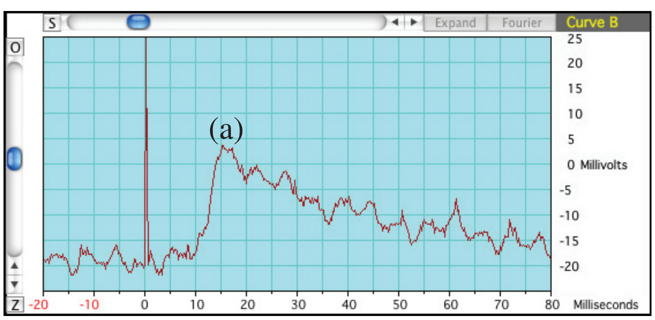


Figure 19.1
With **Automatic Triggering**, the computer looks for the highest point in the next block of data, and plots that at t = 0. Here the high action potential spike is plotted at t = 0.

Change + to –

Another way you can get improper triggering for this experiment is to reverse signal and ground wires so that the action potential spike is down, rather than up. Since MacScope only looks for maximum positive values, it would completely miss the action potential spike. Then the curve would move all over the place, triggered by some noise at the end of the cycle. This problem is quickly remedied by going to the Input Menu and selecting Change + to –. This flips the curve over which is equivalent to reversing the connections to the + and – terminals.

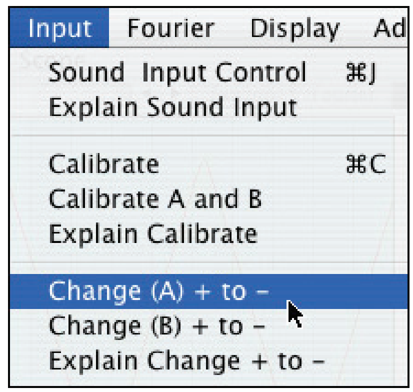


Figure 19.2
Selecting **[Change + to –]** turns the curve over. It has the same effect as reversing the positive and ground terminals.

20) MANUAL TRIGGERING

When automatic triggering fails or does not do what you want it to do, you can use manual triggering where the curve is triggered if the voltage exceeds a trigger voltage level that you set.

In MacScope II, there are two separate trigger modes. One is Stop on Trigger and the other Average Triggered Curves. In Stop on Trigger, the computer stops recording shortly after the first time it detects that the voltage rises above the trigger voltage. It displays the triggered data with t = 0 at the trigger point. The Stop on Trigger mode is most useful for capturing one-time events like the ring of a bell.

The Average Triggered Curves mode allows you to signal average those curves that reach the trigger voltage level, while ignoring curves that do not reach the trigger level. This is convenient when you want to signal average data that is triggered only once in a while and you want to ignore the noise between triggers.

We will discuss more about these two modes after the next section where we describe how to set the trigger voltage level.

21) SETTING TRIGGER LEVEL

Each curve, Curve A and Curve B has its own trigger level setting area. The area for Curve A is shown in Figure (21-1). To set the trigger, the first step is to click on the Trigger A button to enable manual triggering for Curve A. The next step is to use the vertical scroll bar to do a gross adjustment of the trigger voltage. The allowed values are 0 to 2 volts in steps of 20 millivolts. The two sets of small arrows allow fine scale adjustments, the first set in steps of 1 millivolt and the second set in steps of .1 millivolts (100 microvolts).

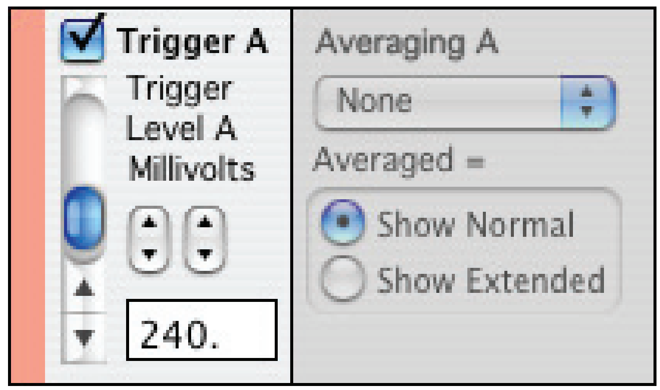


Figure 21.1
The trigger control panel for Curve A.

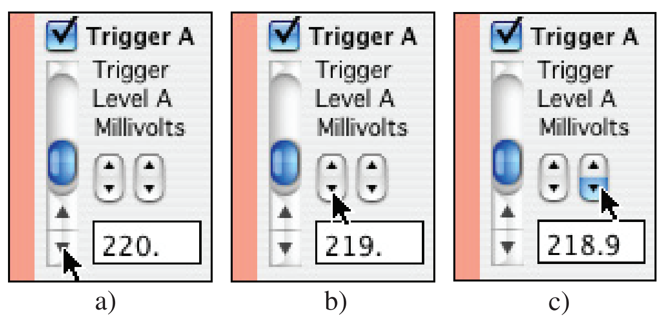


Figure 21.2
Changing the trigger voltage. One click in the down arrow of the main scroll bar (a) drops the trigger 20 millivolts, from 240 mV to 220 mV. Each click in the first small arrows (b) changes the trigger voltage by one millivolt, and each click in the second small arrows (c) changes the trigger voltage by a tenth of a millivolt.

With these three controls, we can set the trigger level anywhere in the range of 0 to 2 volts, to an accuracy of .1 millivolts.

22) STOP ON TRIGGER MODE

In the Stop on Trigger mode, MacScope stops recording shortly after it first encounters a trigger, i.e., the first time the incoming voltage exceeds the trigger voltage. MacScope then discards any old data that may be in the averaging buffer and displays the triggered data. The average data display now contains one window of pretrigger data (before t = 0) and four windows of data after the trigger (after t = 0). If you want to see still earlier data, choose Show Extended to view the extended data. In this mode, no signal averaged data is saved once we get a trigger.

As we mentioned, the Stop on Trigger mode is most useful for recording one time event like the ring of a bell or the call of a frog. The pretrigger window shows you how the event got started.

Figure 22.1
To get this frog to talk, you hold the frog by its front legs, heading away from you, and stroke its back with its mallet. We captured the sound it makes using **Stop on Trigger**. (Frog purchased from **Educational Innovations Inc**, www.teachersource.com.)

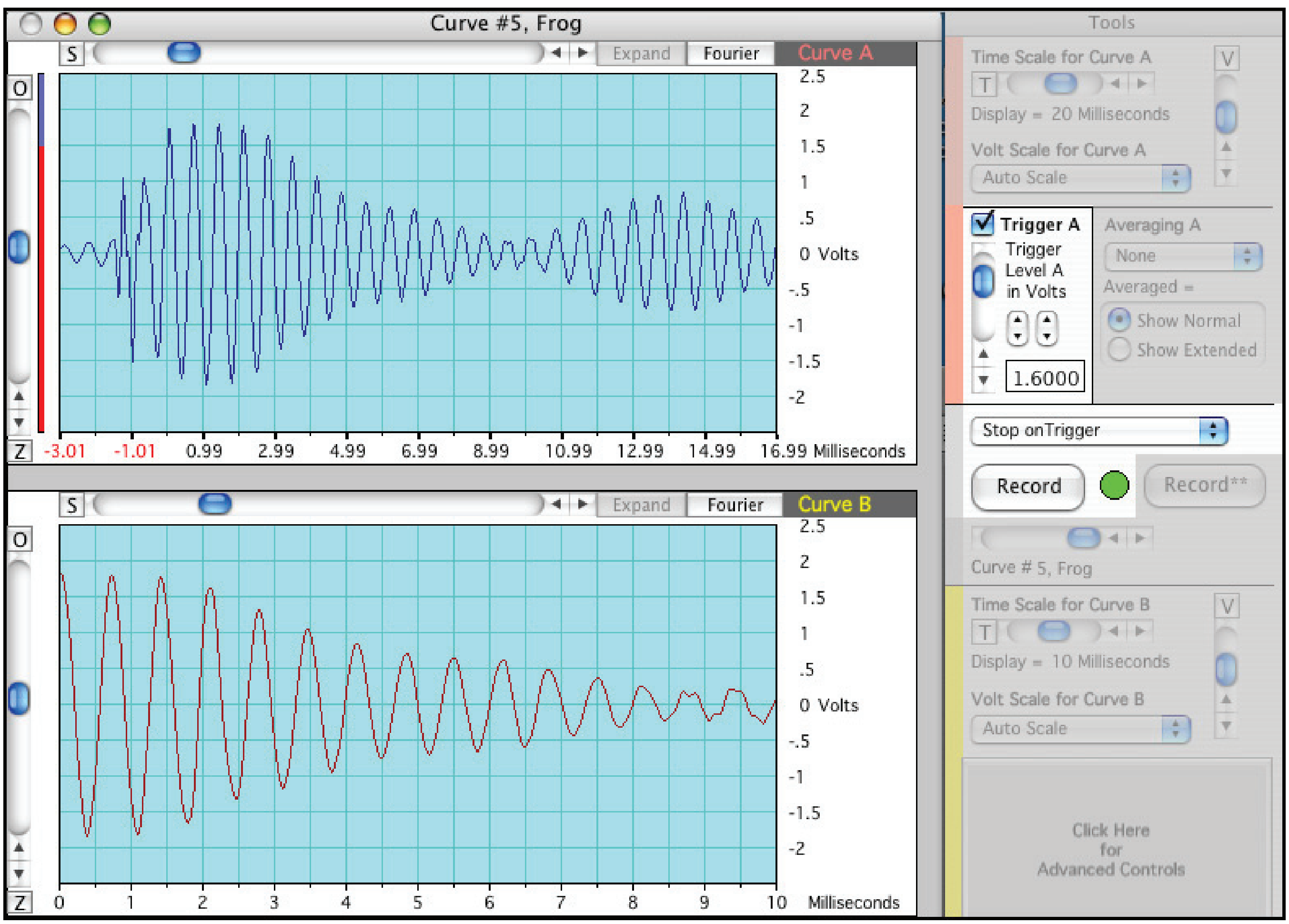


Figure 22.2
Sound of a wooden frog.

23) THE VOLTAGE COLUMNS

When you click on the Trigger A button you will notice a narrow vertical column appears at the left side of Curve A window, as shown in Figure 23.1. The top of the column is blue and the bottom is red. The dividing line indicates the value of the trigger voltage, in the range 0 to 2 volts. For example, if you set the trigger voltage to 1 volt, the dividing line will be half way up, as in Figure 23.1b. As you change the trigger voltage, you will see the dividing line move up and down.

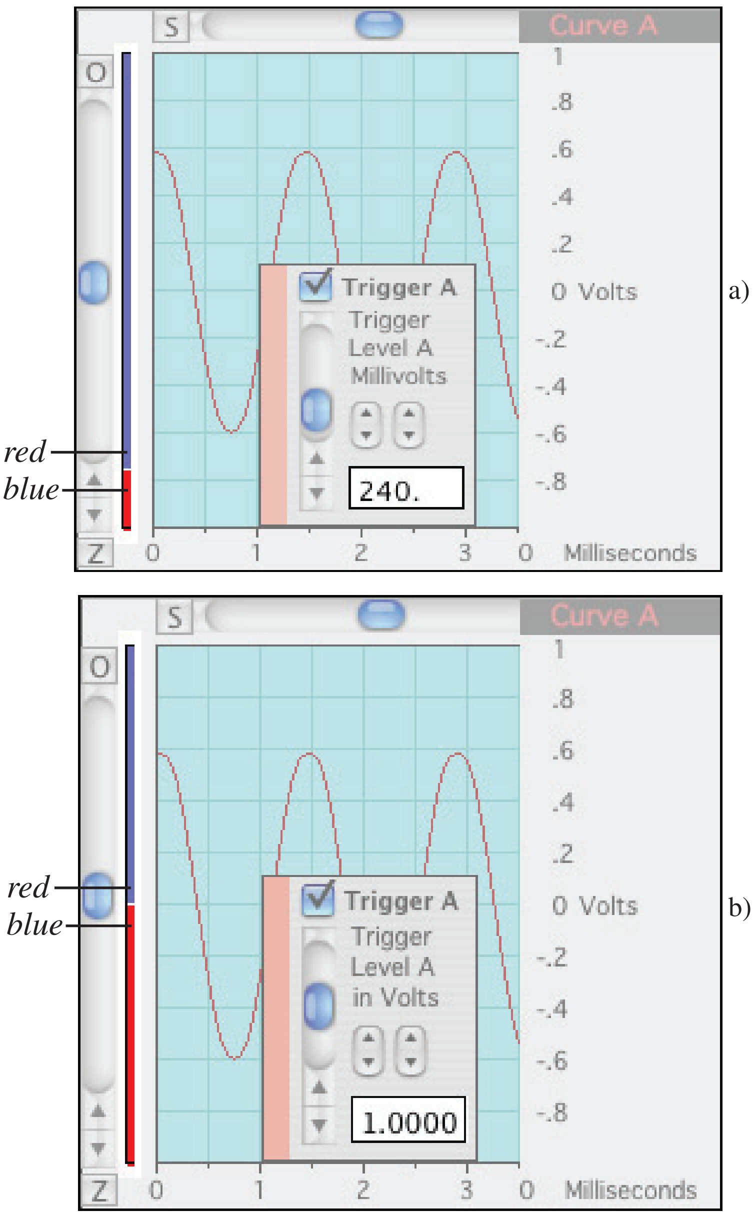


Figure 23.1
The trigger voltage column on the left shows the selected trigger voltage on a scale of 0 to 2 volts. At 240 mV, the column is 1/8 the way up, and half way up at 1 volt.

When MacScope is running, there is another vertical column colored red whose height changes. This voltage column, seen in Figures 23.2 indicates, on a two volt scale, the maximum voltage in the current window being displayed.

When you are using manual triggering and MacScope is recording, you can tell how close the current data is coming to the trigger level by comparing the two voltage columns. In Figure 23.2a, the trigger level column is set at .400 volts and the voltage column is slightly shorter because the sine wave has an amplitude slightly under .4 volts. In Figure 23.2b, the sine wave amplitude exceeds .4 volts, the voltage column is higher than the trigger level column, and we get a triggered curve, which is colored blue.

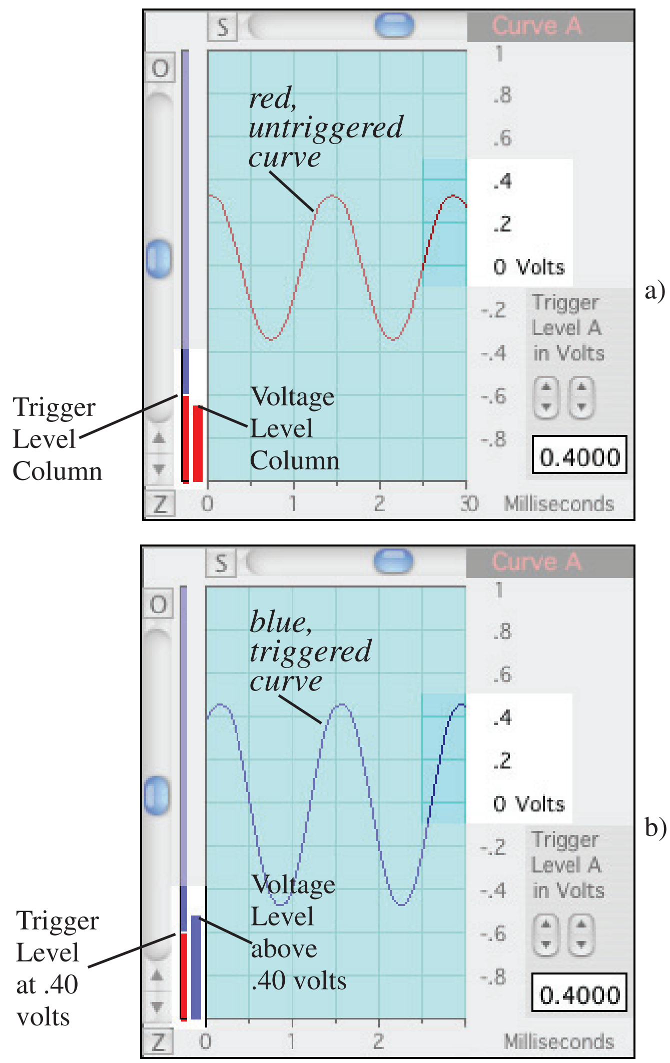


Figure 23.2
Comparing the trigger level and voltage level columns.

24) AVERAGE TRIGGERED CURVES

As we mentioned, in the Average Triggered Curves mode MacScope signal averages only curves which reach the trigger level, ignoring those that do not reach this level. All triggered curves are displayed with t = 0 at the trigger point which, most of the time, is the point where that curve first reaches the trigger level.

If you are running in Average Triggered Curves mode and have not yet gotten a trigger, the scope runs normally, looking like the automatic trigger mode. Once you get a trigger, the triggered curve is plotted in blue and the averaged data from non triggered curves is discarded. From that point on, the display changes only when a new triggered curve is averaged in. If you select None for signal averaging, then the individual triggered curves will be displayed one after the other.

When to Use Average Triggered Curves

There are two main uses of the Average Triggered Curves mode. One is where the trigger occurs infrequently and you want to avoid averaging in the noise between windows.

If your curves are triggered at a regular time interval, and you can set the time scale so that a possible trigger occurs regularly within one or two windows of data, then the automatic trigger will not add in untriggered curves and you do not need to use a triggering mode.

There are cases, even when the triggering occurs regularly, that you may need to use the trigger mode. Consider Figure 24-1, a repeat of Figure 18.2 where the junction potential curve is triggered by the action potential spike. In this case, the action potential spikes at t = 0 is higher than the top of the next junction potential curve, but the spike at t = 100 is lower than the following junction potential.

If this change happens fairly often, which it can for some nerves, and we trigger on the highest point as we do with automatic triggering, the trigger point will move back and forth between the spike and the curve. On the screen you will see an unaveraged curve move back and forth and the signal averaged curve will not be accurate. To see if this is happening, it is good to look at both a signal averaged curve and a non averaged one at the same time. If the non averaged one is hopping around you should fix the problem before trusting the averaged one. You can fix this problem by using the Average Triggered Curves mode and making a good choice of the trigger level.

To set the trigger level, first stop the recording so that you have a couple of good examples of curves you want to trigger, as we did in Figure 24.1. You may want to look through the extended data to find good examples. Then choose a trigger level that is high enough to avoid most of the noise, but low enough to always catch the action potential spike. Because the action potential spike always comes just before the junction potential, the curve will reach the trigger level at the action potential first, and that will almost always be the trigger point.

Once you have adjusted the trigger level to the height you want (I would start with 1 millivolt for Figure 24.1), check that you have signal averaging, choose Average Triggered Curves, and then press Record ** to preserve these settings. If you make a mistake and press Record, you will have to select Averaging again. You can change the trigger level while the program is running to search for the best trigger level. If you do that, stop and start again to get rid of unwanted data.

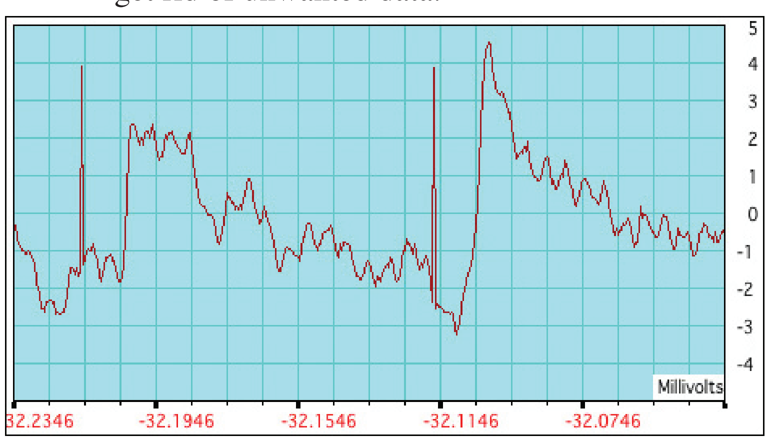


Figure 24.1
Automatic Triggering would not work here because the action potential spike is not always higher than the following synapse potential. Here **Manual Triggering** is needed.

25) MATH WAVES AND SIGNAL GENERATOR

It is easy to get a sine wave into MacScope, just whistle into a microphone. This is how we made the sine waves we used in our discussion of the uncertainty principle.

Because of MacScope's ***Fourier analysis*** capability, it is worthwhile getting other kinds of wave structures into MacScope so that one can study their harmonic structure. In a traditional electronics lab this would be done by recording the output of a device called a ***signal generator***. A typical signal generator outputs sine waves, square waves and triangle waves at frequencies and amplitudes which you can dial.

What we have done with MacScope is to include math waves, so that MacScope can create its own waveforms. Then the fact that MacScope can play selected harmonics or repeated sections of a curve, allows MacScope to act as a signal generator, as well as an oscilloscope. And the math waves are immediately available for analysis.

The math waves we have included are the sine wave, square wave, rectangle wave, triangle wave, and two ramp or sawtooth waves, all shown in Fig.(25.1). The square wave alternates between two voltage levels, spending equal lengths of time at each level. The rectangle wave can spend different lengths of time at the upper and lower voltage levels. We introduce the rectangle wave so that we could have a wave pattern that matched the slit structures used in our study of ***Fourier optics***.

To get the waveforms shown in Fig.(25.1), first go to the ***File*** menu and select ***Create Math Wave*** as shown in Fig.(25.2). This brings up the Create Math Wave window shown in Fig.(25.3). The first thing you do is select the frequency you want for the wave, and then use the drop down menu to select the kind of wave that you want. In Fig.(25.3) we have selected a frequency of 440 Hz and will be choosing a square wave. (If we had chosen a sine wave and played the result, we would have A above middle C.) The resulting square wave is shown in Fig.(25.4).

For the rectangle wave there is an additional step. After selecting the rectangle wave, you get the additional window shown in Fig.(25.5) asking you to choose the ratio of the width of the tops, to the widths of the bottoms. In the background of that figure is a 440 Hz rectangle wave whose tops are 33% of one cycle.

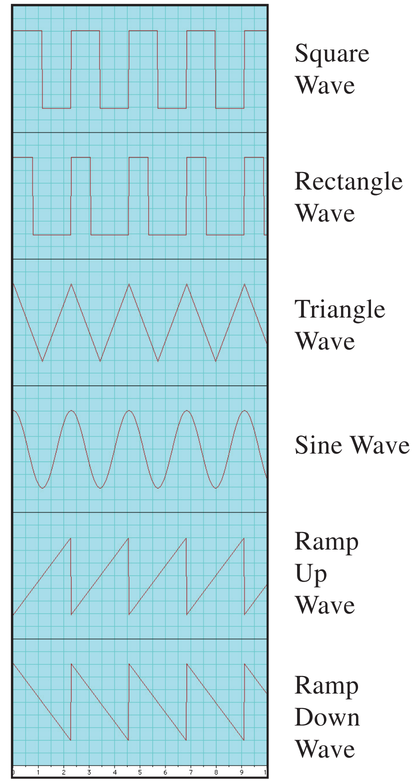


Fig. 25.1 ***The math waves.***

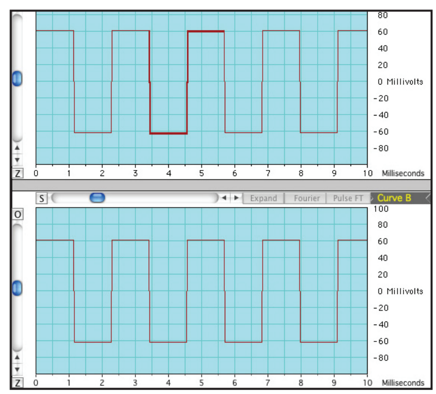


Fig. 25.4 ***Resulting square wave.***

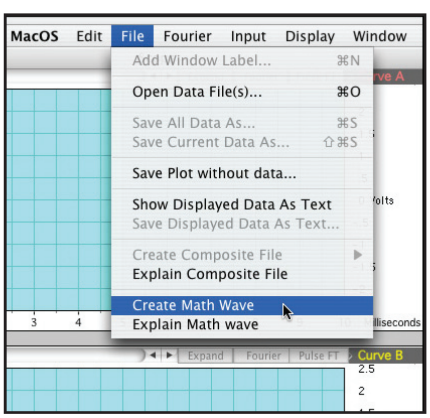


Fig. 25.2 *Selecting* **Math Wave.**

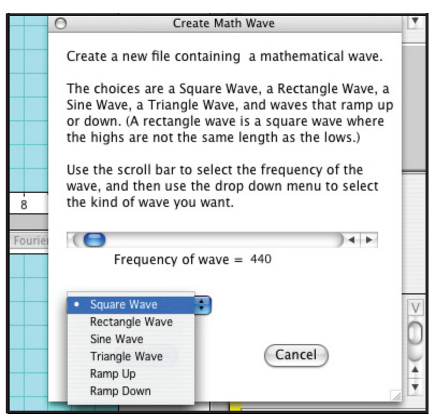


Fig. 25.3 ***Choosing square wave.***

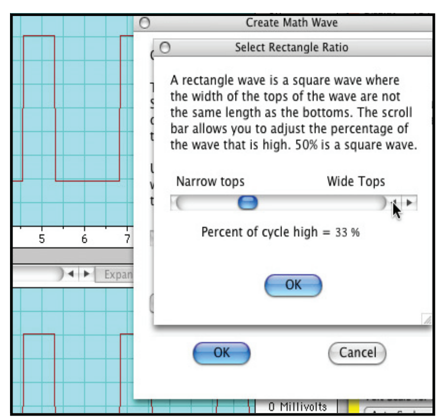


Fig. 25.5 ***Selecting rectangle ratio.***

Fourier Analysis of a Square Wave

In textbooks that introduce Fourier analysis, it is traditional to use a square wave as the first example to show how a complex waveform can be constructed from a harmonic series of sine waves. This is a bit ironic, because a square wave is discontinuous at each jump, while Fourier's theorem tells us that any ***continuous*** wave shape can be constructed out of harmonic sine waves. The result is that when you try to construct an ideal square wave from sine waves, you end up with a small blip at the discontinuity, a blip that will not go away as you add more and more harmonics. This blip is called the ***Gibbs effect***.

Despite the blip, looking at the harmonic structure of a square wave can be quite instructive. In Fig.(25.6) we are selecting precisely one cycle of a square wave. Pressing the ***Fourier*** button we get the analysis shown in Fig.(25.7). The first thing we notice is that all the even harmonics are missing. The square wave is composed entirely of odd harmonics 1, 3, 5, etc.

There is also a smooth progression in the way the amplitudes of the harmonics decreases. If you look closely, you will see that the 3rd harmonic has 1/3 the amplitude of the first; the 5th harmonic has 1/5 the amplitude of the first, and so on. You can see this result more clearly if you click on the button labeled ***FT Data*** and look at the numerical values of the harmonics.

In Fig.(25.8) we look at how the square wave can be reconstructed from its harmonics. The first harmonic is clearly the sine wave that most closely fits into the square wave. The second harmonic, when added in, lowers the peaks in the sine wave. As we go up through harmonic 13, it is clear that each harmonic added in brings the composite wave closer to a square wave.

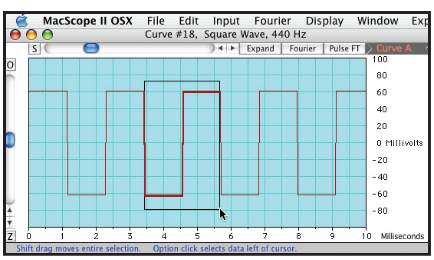


Fig. 25.6. ***Selecting one cycle of a square wave.***

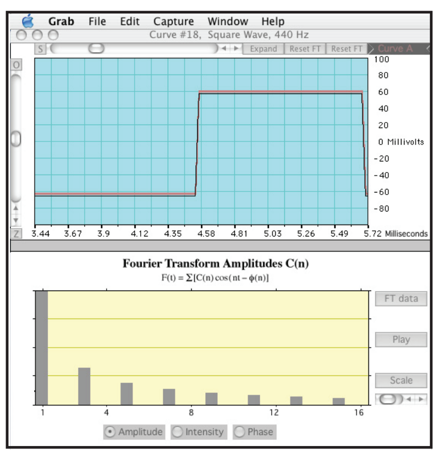
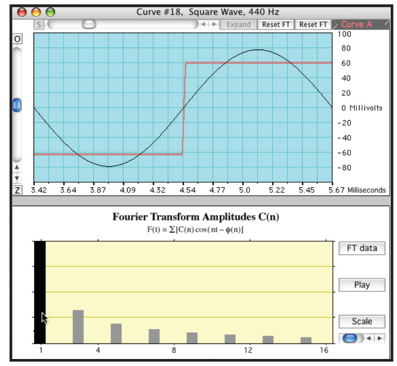
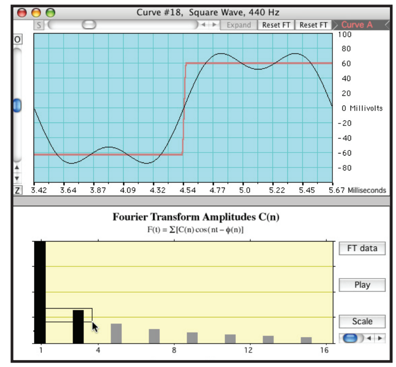


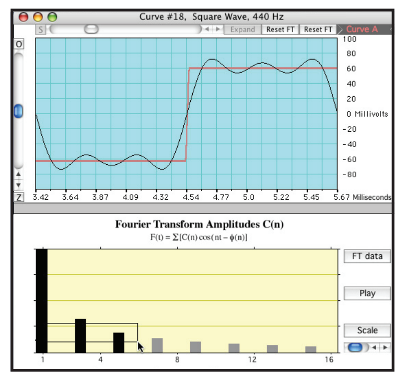
Fig. 25.7 ***The square wave has only odd harmonics.***



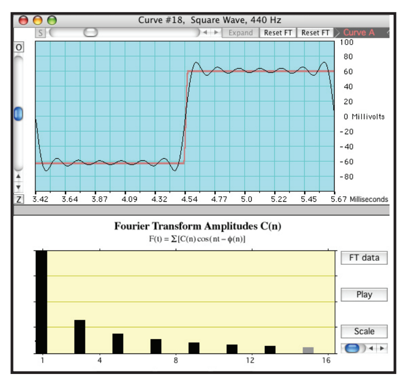
Viewing the first harmonic



Combining harmonics 1 & 3



Harmonics 1 + 3 + 5



Through harmonic 13

Fig. 25.8 ***Building up a square wave from its harmonics.***

Frequency Response

In Fig.(25.9), we added all harmonics up to #50. We chose to add the first 50 harmonics for the following reason. With our basic frequency at 440 Hz, the 50th harmonic has a frequency of 50*440 = 22,000 Hz. But 22,000 Hz is the maximum frequency that can be recorded by the computer's audio input. (The computer records 44,000 points per second, and you need at least 2 points per cycle to recognize a frequency.)

As a result, if you tried to record a perfect square wave with MacScope, or any audio computer input, no frequency component above 22,000 Hz would be recorded, and the result would look like Fig.(25.9).

When we were first testing MacScope, we had our electronics technician build a 440 Hz square wave generator. When we looked at the output of the generator on his high frequency lab scope, we saw a smooth square wave. We then took his square wave generator home and looked at its output on MacScope, and saw bumps at every transition. At first we wondered what had gone wrong. Why couldn't MacScope give as good a plot of the square wave as the technician's lab scope?

We took the square wave generator back to the technician to see if it had been damaged on the way home. The answer was NO—the generator still produced a clean square wave on his lab oscilloscope. It slowly began to dawn that MacScope's limited frequency range was the problem. In effect, MacScope was showing only the first 50 harmonics of the square wave being put out by the square wave generator.

Standard computer audio inputs cut off at both the high and low frequency ends. By sampling the voltage 44,000 times a second, no frequencies higher than 22,000 Hz can be recorded. (This maximum frequency for sampled data is called the ***Niquist*** frequency.) There is also an electronic filter that prevents frequencies below about 20 Hz from being recorded. (The result is that MacScope is called an AC oscilloscope, i.e., one that records only "alternating current" signals and ignores low frequency voltage changes.)

We had our electronics technician drop the frequency of our square wave generator from 440 Hz to 60 Hz, and got the result shown in Fig.(25.10). The upper window shows three full cycles of the wave, while the lower window shows one cycle in more detail. (1/60 sec = 16 2/3 milliseconds.) In these diagrams we see the effects of both the high and low frequency cutoffs. The little bumps due to the high frequency cutoff are still there, while the sags in voltage at the top and bottom of the square wave are due to the low frequency cutoff.

The lesson we learned was that we were not going to get a good square wave into MacScope through the frequency cutoffs used by computer sound inputs. That is when we decided to create square waves inside MacScope with a ***Math Wave*** option.

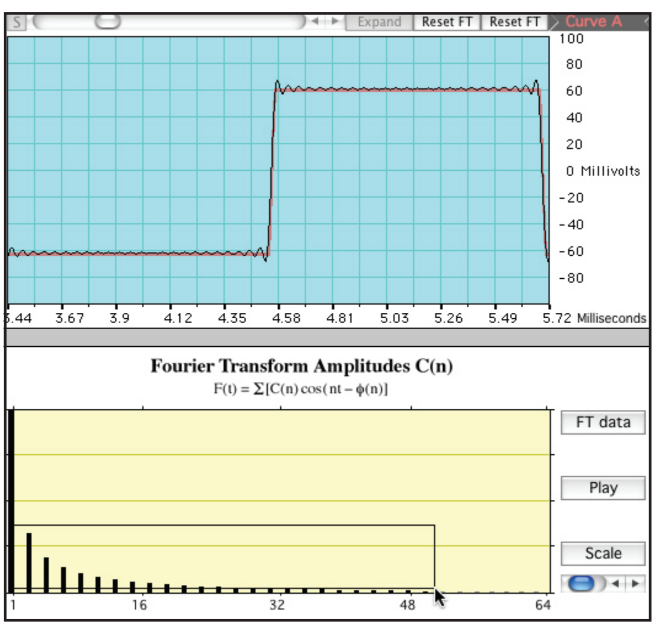


Fig. 25.9 ***Selecting harmonics up to a frequency of 22,000 Hz, the computer's Niquest frequency.***

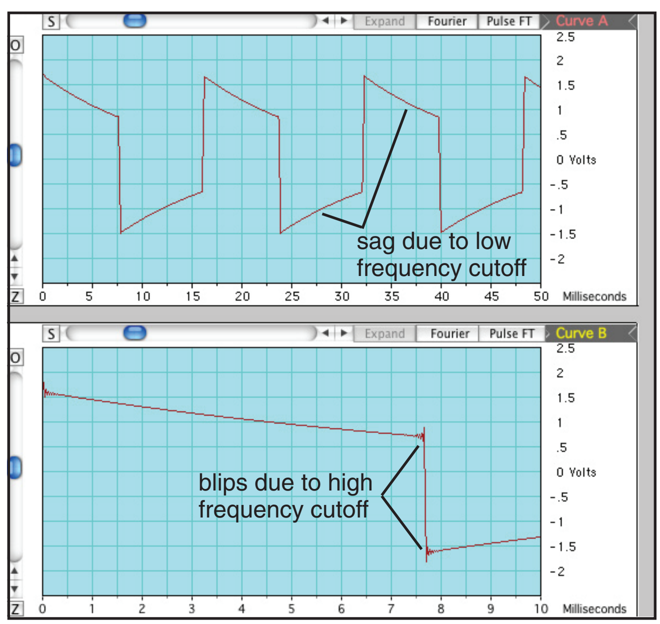


Fig. 25.10 ***Recording of a 60 Hz square wave. The low frequency cutoff causes the curve to sag, the high frequency cutoff causes the blips.***

26) PULSE FOURIER TRANSFORM

To do a Fourier transform with MacScope, you first select a repetitive section of a curve and press the ***Fourier*** button. MacScope expands that section of the curve to fill the display window and then calculates what harmonics are needed to reconstruct the curve out of sine and cosine waves. The plot displayed in the Fourier transform window shows the amplitudes of the required sinusoidal waves.

An explicit mathematical assumption is that the expanded wave pattern you see in the display window repeats forever. This is a mathematical requirement because the sine and cosine waves out of which you are reconstructing your curve are, themselves, infinitely long.

When we say that a sine wave is infinitely long, we mean that if you chop off a sine wave at a finite length, it is no longer a pure sine wave. To see what a chopped off sine wave actually is, we have introduced the ***pulsed Fourier transform*** into MacScope.

As we mentioned, when you press the ***Fourier*** button, the selected section of curve expands to fill the display window. If, instead, you press the ***Pulse FT*** button, the selected section of curve stays where it is, and the rest of the curve is zeroed.

As an example, in Fig.(26.1a) we used ***Math Wave*** to create a sine wave. (We could get nearly the same result whistling into a microphone.) We then selected one cycle of the sine wave near the center of the window.

To get Fig.(26.1b), we pressed the ***Fourier*** button. That one cycle of the sine wave expanded to fill the display window. If that one cycle were repeated indefinitely, we would get a pure sine wave, which is indicated by the fact that we see there is only one harmonic present.

To get Fig.(26.2c) we pressed the ***Pulse FT*** button instead. The non selected part of the curve was zeroed, and the selected cycle remained as a pulse. You may think of this one cycle as a sine wave, but that is not what the Fourier analysis window is telling us. We see that the chopped off sine wave is a complex mixture of many waves. It takes a lot of sine waves to make up a short pulse.

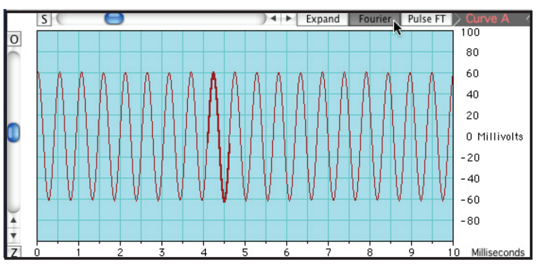


Fig. 26.1a. ***We selected one cycle of the sine wave and pressed the* Fourier *button.***

Fig. 26.1b
The* Fourier *and* Pulse FT *buttons.

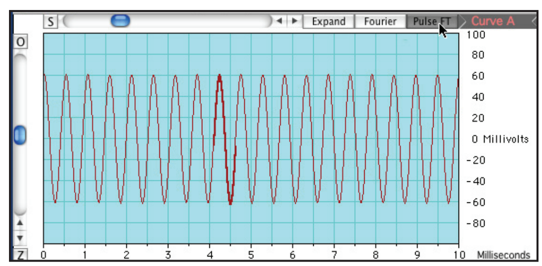


Fig. 26.2a ***We selected one cycle of the sine wave and pressed the* Pulse FT *button.***

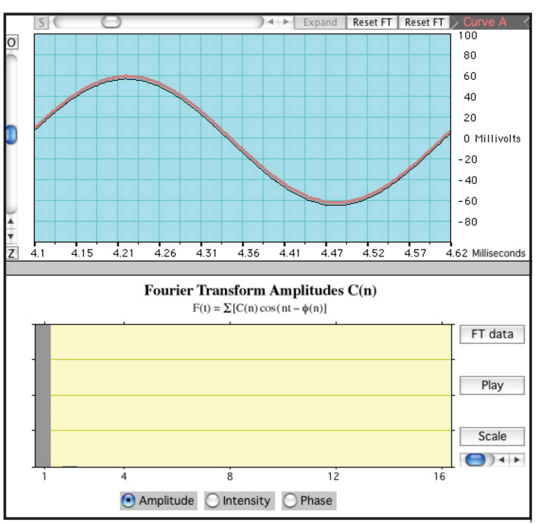


Fig. 26.1c. ***If the cycle repeats, we get essentially only one harmonic.***

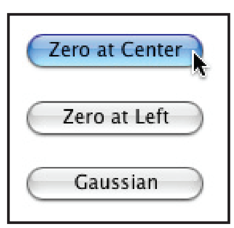


Fig. 26.2b
When you press the* Pulse FT *button, a window comes up asking where to locate the zero line. We chose the zero to be at the center of the pulse.

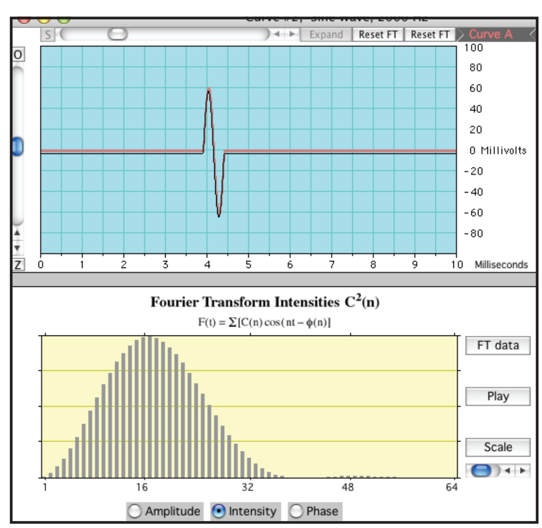


Fig. 26.2c ***Chop off the sine wave and you get many harmonics.***

To see how we build up a pulse out of sine waves, we have in Fig.(26.3a) selected one of the center harmonics and see a feeble sine wave in the display window. This wave has about the same frequency as our original sine wave, but a much smaller amplitude.

In Fig.(26.3b) we selected 5 of the center harmonics and see that we are beginning to build a pulse at the center and reduce the waves at the edges. In Fig.(26.3c) we selected about 2/3 of the center harmonics and see that the pulse has been mostly reconstructed and the waves out from the pulse have been mostly cancelled out. By Fig.(26.3d) where we have selected the center bulge of harmonics, we see that we have an almost complete reconstruction of the chopped off section of a sine wave. The lesson is that a pure sine wave is mathematically infinitely long in both directions. Whenever you cut off a sine wave, the result is no longer a pure sine wave but, instead, a mixture of sine waves.

Our analysis here is not just some abstract mathematical theory. There are important physical consequences which can be directly seen in the behavior of laser beams.

A laser beam is famous for the purity of the light in the beam. The typical laser beam is essentially a single wave with a unique frequency (color) and amplitude. A laser beam comes about as close as we can to a physical representation of a mathematical sine wave. For example, a red laser beam one kilometer long contains nearly two billion similar wavelengths.

The continuous lasers you are familiar with are not the only kind used in research labs. Chemists, for example, need lasers that emit a very short pulse in order to study the behavior of atoms during a chemical reaction. There are now infrared pulsed lasers, where the pulses are only a few wavelengths long. When you create a short laser pulse, you are essentially chopping the light wave off in much the same way we chopped off the sine wave in Fig.(26.2c). Instead of getting a beam of a pure color (a single frequency or harmonic), you get a beam with a spread of colors (or frequencies).

Figure (26.4a) shows the electric field waves in an infrared pulsed laser beam about 12 cycles long. Figure (26.4b) shows the spread in wavelength (spread in colors) of the light in the pulse. Notice that the spread in harmonics contained in our chopped off sine wave closely resembles the spread in wavelengths in the laser pulse. (1 nanometer[nm] = 10^{-9}meters).

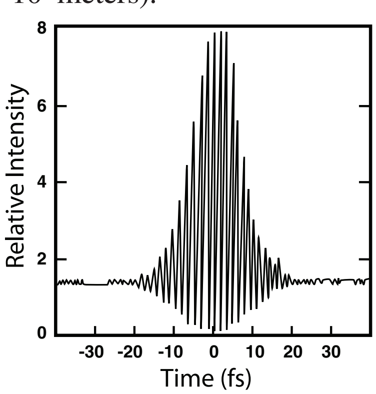


Fig. 26.4a ***Electric field intensity of a 12 cycle long infrared laser pulse. Because the intensity is the square of the field, there are about 24 peaks.***

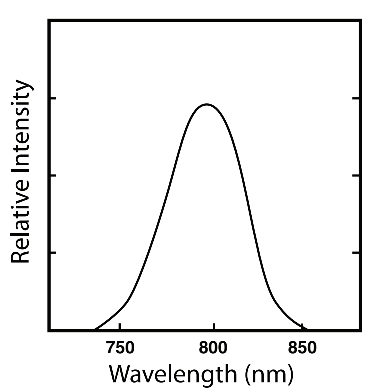


Fig. 26.4b ***Spectrum of light in the laser pulse. The spectrum is nearly as wide as the visible spectrum whose wavelengths go from 400 to 700 nm.***

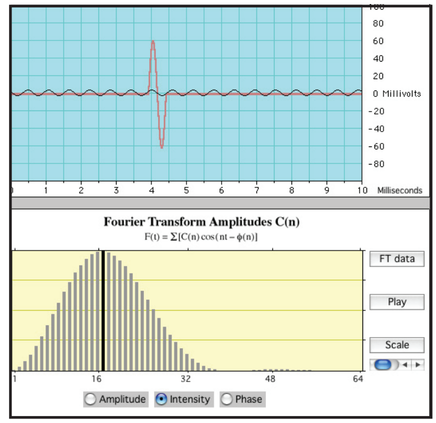


Fig. 26.3a ***The biggest harmonic.***

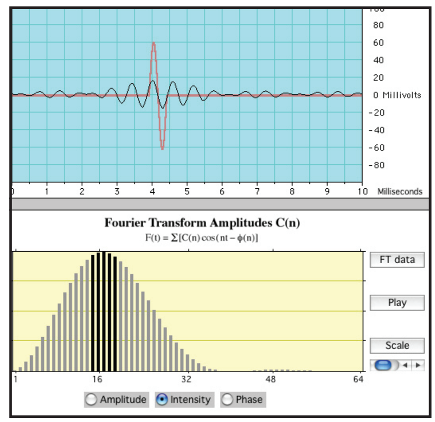


Fig. 26.3b ***Five of the harmonics.***

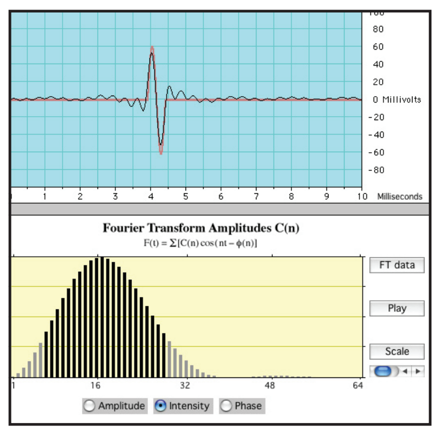


Fig. 26.3c ***2/3 of the harmonics.***

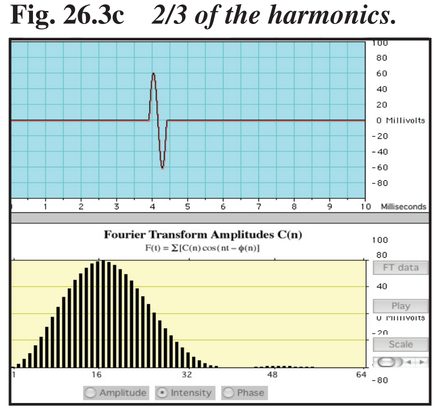


Fig. 26.3d ***Most of the harmonics.***

27) TECHNICAL DETAILS, PULSE FOURIER TRANSFORM

With the pulse Fourier transform, we can study different shaped pulses by selecting different parts of any wave we can record, or get using math waves.

Here we wish to discuss some different ways we have to create a pulse, and to look more carefully at how we interpret the results. First the interpretation.

Repeated Pulses

When we do a pulse Fourier transform, the only difference between that and a regular Fourier transform is that we zero out the non selected part of the curve rather than expanding the selected section to fit the display window. After that everything else is the same.

Explicitly what you see in the display window is assumed, mathematically, to repeat forever. If we display one complete cycle of a sine wave, and that repeats indefinitely, the result becomes a pure sine wave that shows up as a single harmonic in Fig.(26.1b). If we created a pulse as in Fig.(26.2c), MacScope still assumes that the window is repeated indefinitely. As a result the Fourier analysis window shows the harmonic structure of the repeated series of pulses as indicated in Fig.(27.1b).

Our analogy to a pulsed laser still holds fairly well. A pulsed laser sends out not a single pulse, but a steady series of pulses like that in Fig.(27.1b). The main difference between the pulsed laser and our Fig.(27.1b) is that the repeated laser pulses are much farther apart, i.e., separated by many more wavelengths. This wider separation, however, does not appear to make much of a difference in the resulting spectra of harmonics. Our pulse Fourier analysis spectra has essentially the same shape as the spectra of the laser pulse.

Creating the Pulse

For convenience, MacScope provides three different ways of constructing the pulse. When you press the ***Pulse FT*** you get the window shown in Fig.(27.2) allowing you to choose between zeroing the curve 1) at the center of the pulse, 2) at the left edge, and 3) using a Gaussian envelope. In Fig.(26.2c) we zeroed the curve at the center of the pulse, leaving as much curve above as below the zero line. This gave us the symmetric pulse we wanted to study.

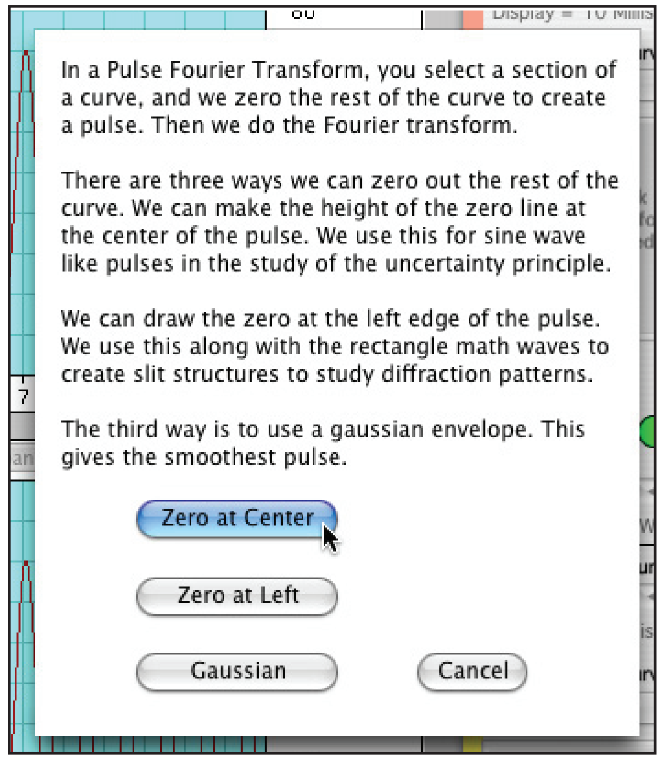


Fig. 27.2 ***Where to put the zero line.***

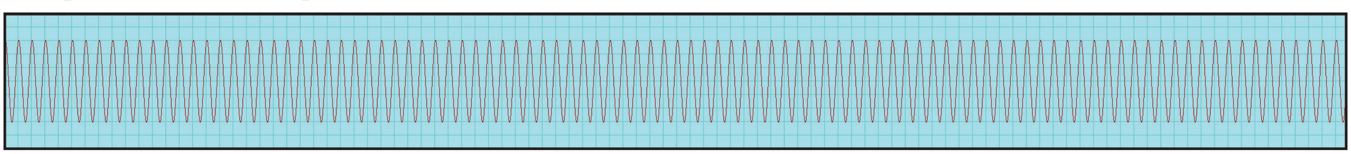


Fig. 27.1a ***Part of a continuous sine wave.***

Fig. 27.1b ***Repeated set of pulses.***

Creating Slits

In our paper on ***Fourier Optics***, we formed a pulse using the left edge option for zeroing the curve. The point was to create what looked like the set of slits that we used to study the diffraction of a laser beam. That allowed us to demonstrate that the Fourier transform of a slit structure was similar to the diffraction pattern produced by a laser beam passing through the slits.

This study began when we helped with a student project by creating the slits using *Adobe Illustrator*™. First we drew parallel lines spaced so that the distance between line centers was three times the thickness of the lines. We then used Illustrator's ***Scale*** command to reduce the size of the plot until the thickness of the lines was 50 microns (fifty millionths of a meter). The students took the program over to a high resolution printer at the medical school and printed the lines in reverse on a transparent medium. The result was a diffraction grating with slits 50 microns wide spaced 150 microns apart. These slits and the resulting diffraction patterns are shown in Fig.(27.3).

To create corresponding slits with MacScope, we used the ***Rectangle Math Wave***, with the top of the waves 33% of one cycle as shown in Fig.(25.5). To create three slits, we selected three cycles, pushed the ***Pulse FT*** button, selected to zero the curve at the ***left edge*** and got the result shown in Fig.(27.4).

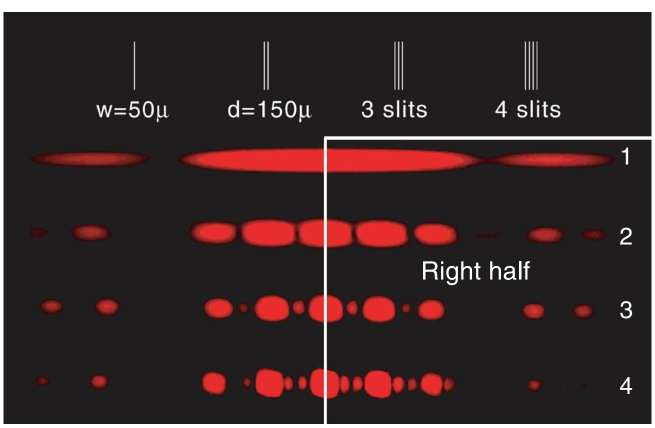


Fig. 27.3 ***The slits we created using Adobe Illustrator, and the corresponding diffraction patterns.***

In Fig.(27.4) we have plotted the laser diffraction pattern below the graph of the harmonics contained in the slit pattern. The only adjustment we made was to scale the diffraction pattern so that the bumps in the diffraction pattern match the bumps on the Fourier harmonic plot.

When you photograph a diffraction pattern, you are observing the intensity or energy density of the light pattern. The intensity is proportional to the square of the amplitude of the light wave.

In MacScope there are three ways to look at the Fourier transform harmonics. You can look at the amplitudes by selecting the ***Amplitude*** button, the phases by selecting the ***Phase*** button, or the intensities by selecting the ***Intensity*** button. With the intensities we are simply plotting the square of the amplitudes. Figure (27.4) shows the intensities of the harmonics contained in the three slits, intensities that closely match the diffraction pattern.

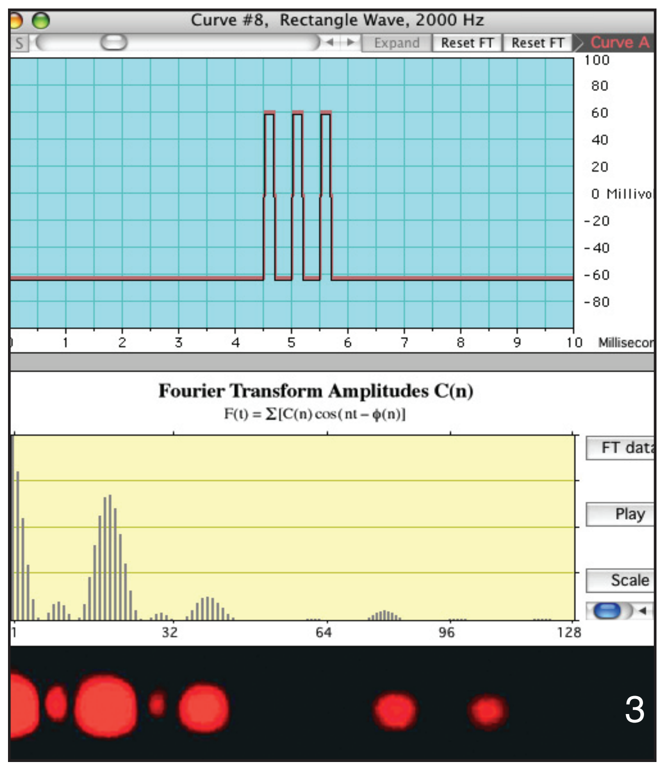


Fig. 27.4 ***Comparison of the right half of a 3 slit diffraction pattern with the Fourier transform of the slit structure.***

Gaussian Pulse

Here we started with the sine wave of Fig.(26.2a) and chose the ***Gaussian*** envelope to more closely represent the experimental data. In Figs.(27.5) we see the result of making a Gaussian pulse from one, two, four, and eight sine wave cycles.

(Mathematicians like the Gaussian pulse because of the theorem that the Fourier transform of a Gaussian pulse is a Gaussian distribution of harmonics.)

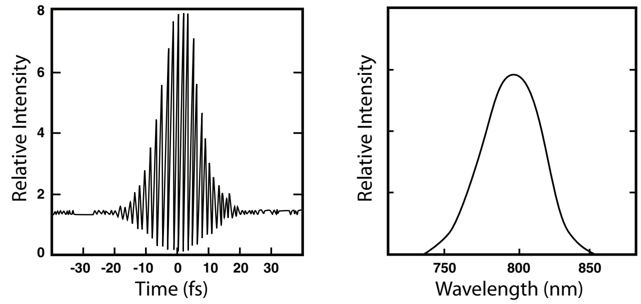


Fig. 26.4 (repeated)
The experimental pulse is more like a Gaussian.

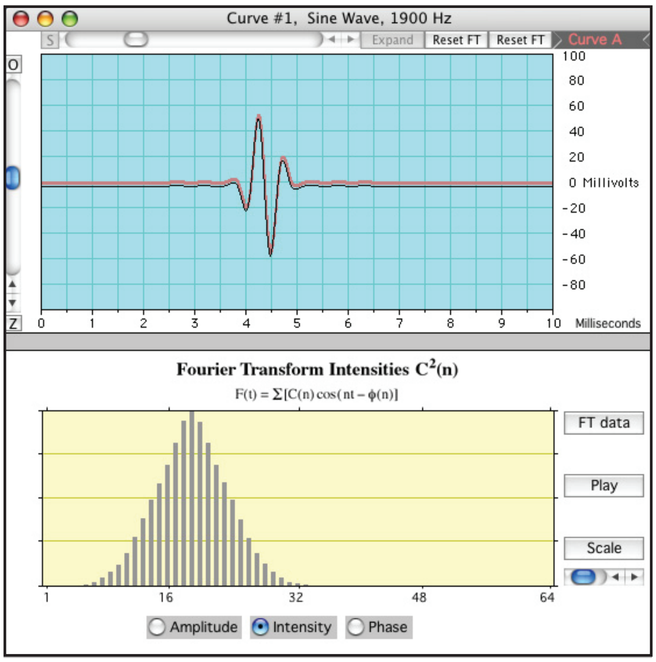


Fig. 27.5a ***Gaussian pulse from 1 cycle***

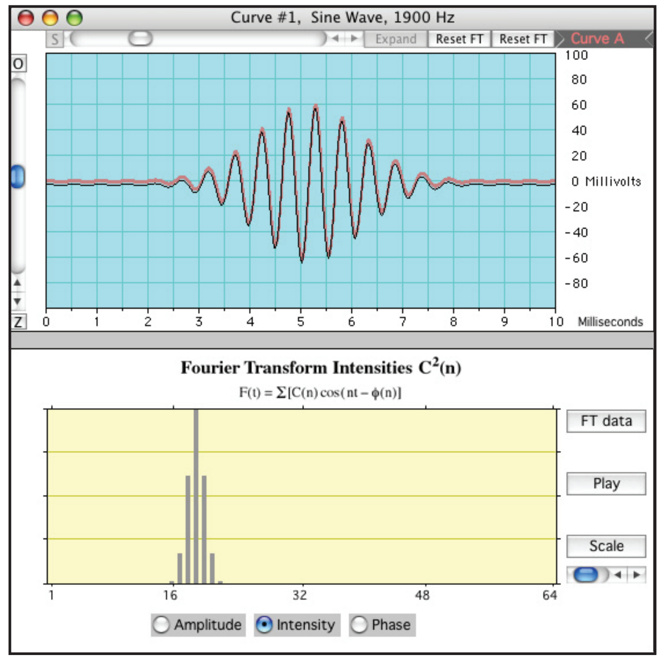


Fig. 27.5c ***Gaussian pulse from 4 cycles***

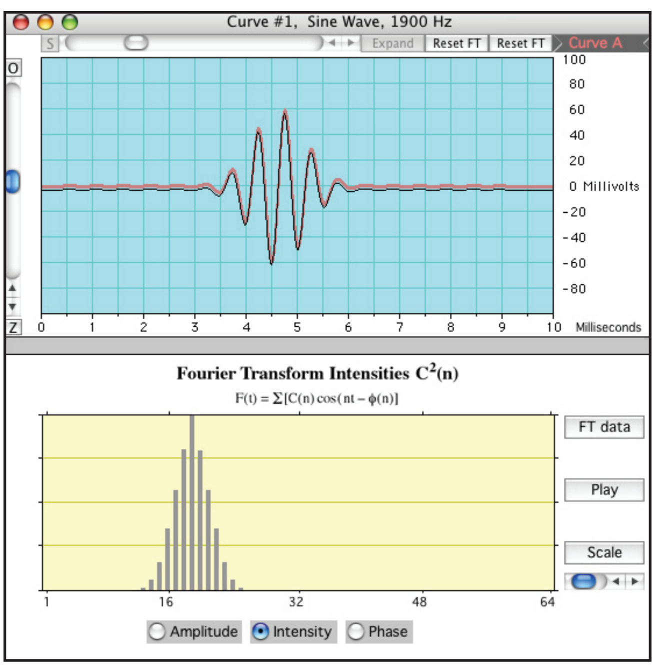


Fig. 27.5b ***Gaussian pulse from 2 cycles***

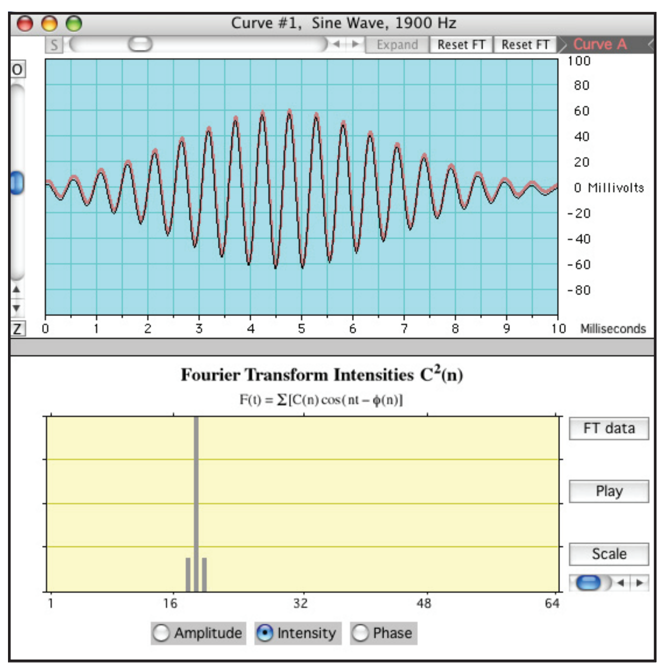


Fig. 27.5d ***Gaussian pulse from 8 cycles***

28) PLAY SOUND & SIGNAL GENERATOR

MacScope has the ability to play selected harmonics. There are two reasons for doing this. One is to hear how a note is constructed by listening as you add harmonics. The other is to use MacScope as a signal generator.

To demonstrate what we get when MacScope acts as a signal generator, we used ***Math Wave*** to create a 256 Hz ***Ramp Up*** wave seen in Fig.(28.1a). Selecting one cycle, we pressed the ***Fourier*** button and got the transform seen in Fig.(28.1b). After selecting the first six harmonics, we pressed the ***Play*** button and got the ***Start Playing Sound*** window. If you have set your computer to play sounds, you will hear a note composed of the first six harmonics of the ramp wave.

What we did instead, is to take the voltage signal that would have gone to the loudspeaker, and use it as the input to MacScope running on an older OS9 Macintosh. Fourier analyzing that signal on the old Mac gave us the results seen in Fig.(28.3). We see that the six harmonics came through quite well.

For Fig.(28.4), we created a 256 Hz ***Ramp Up*** wave on a Dell Windows computer, played six harmonics, and recorded them a Mac. We still see the six harmonics, but the relative amplitudes appear to be a bit more distorted by the Dell's output amplifier.

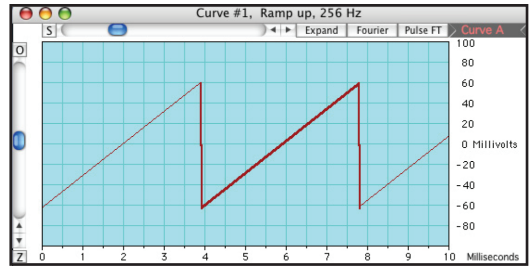


Fig. 28.1a ***We used Math Wave to create a 256 HZ* Ramp Up *wave. One cycle is selected.***

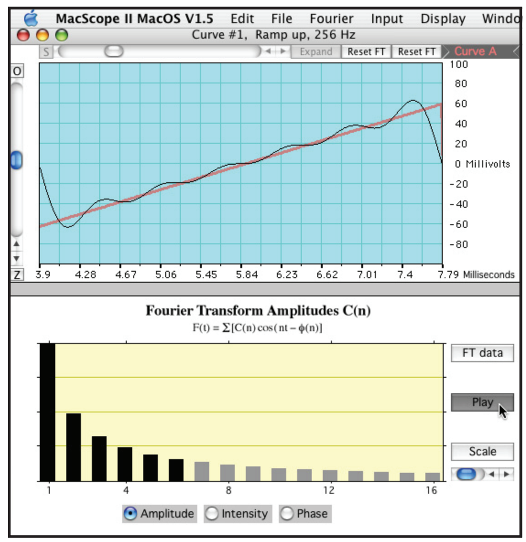


Fig. 28.1b ***Fourier transform of our* Ramp Up *wave. We selected the first 6 harmonics and pressed the* Play *button.***

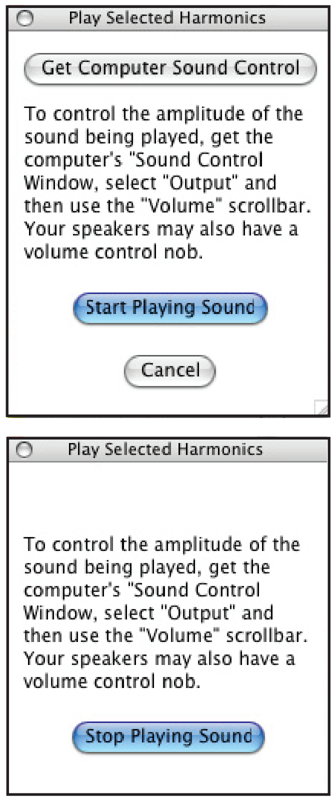


Fig. 28.2 ***Start and Stop* Playing Sound *windows.***

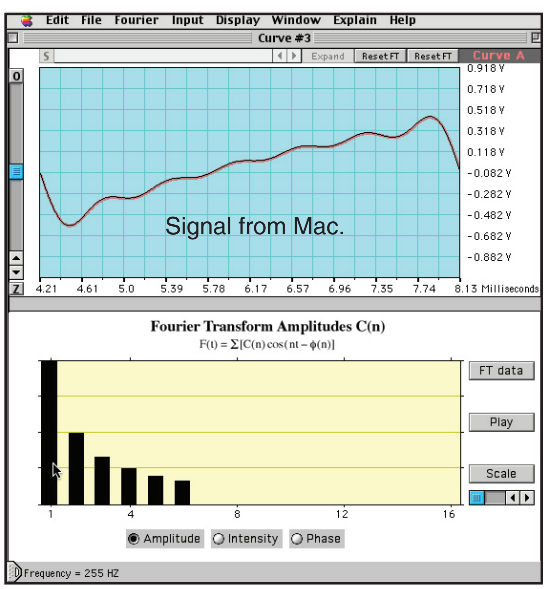


Fig. 28.3 ***Signal played on the newer OSX Mac recorded on an older OS9 Mac.***

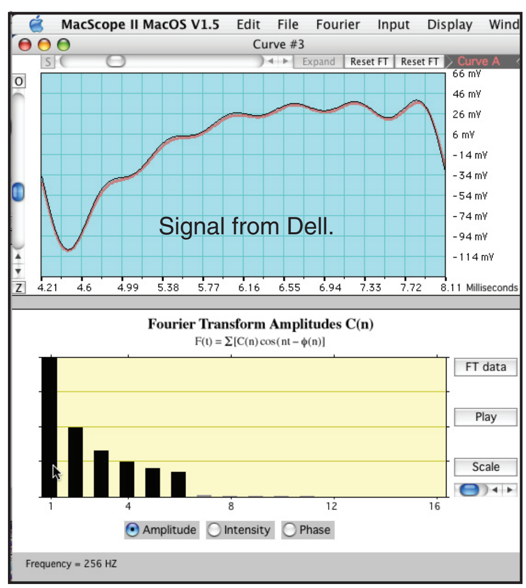


Fig. 28.4 ***Same signal, played on a Windows Dell computer recorded on the OSX Mac. Still six harmonics, but more distortion.***

INDEX

D

E

F

G

H

I

J

K

L

M

T

U

V

W

Y

Z

Physics2000 Textbook

In the special theory of relativity, Einstein combined two apparently simple ideas. One is that you cannot detect uniform motion. That means that when you are riding in a jet aircraft, traveling at three quarters the speed of sound, things behave about the same way they do when you are sitting in your living room. The second idea is that the speed of light is the same to all observers. We are not as familiar with this idea because light travels too fast to easily measure its speed.

We begin the Physics 2000 text by looking at the consequences of these two ideas. Using the Pythagorean theorem, we find that moving clocks run slow, moving lengths contract, and perhaps what is most interesting, even the order of two events can be different depending upon the point of view of the observer. To see that this is the way nature actually behaves, we have included on the CD the 36 minute movie showing the experiment on the lifetime of muons raining down on Mt. Washington. We see that muons traveling at .995 the speed of light live longer than muons at rest – nine times longer. From the point of view of the muons, however, Mt. Washington has contracted to 1/9 of its published height.

By introducing the concepts of special relativity in the first chapter of the text, we can use these ideas throughout the text to present physics from a twentieth century, not nineteenth century, point of view. Early on we discuss the relationship between mass and energy. Magnetism is introduced as a relativistic effect of Coulomb's electric force law. After discussing Maxwell's theory of light waves, we can go directly to the particle nature of light and the simple mechanics of a zero rest mass particle. With this background, we introduce the general particle-wave nature of matter and the basic ideas of quantum mechanics. This approach avoids the great divide that usually occurs between classical and modern physics.

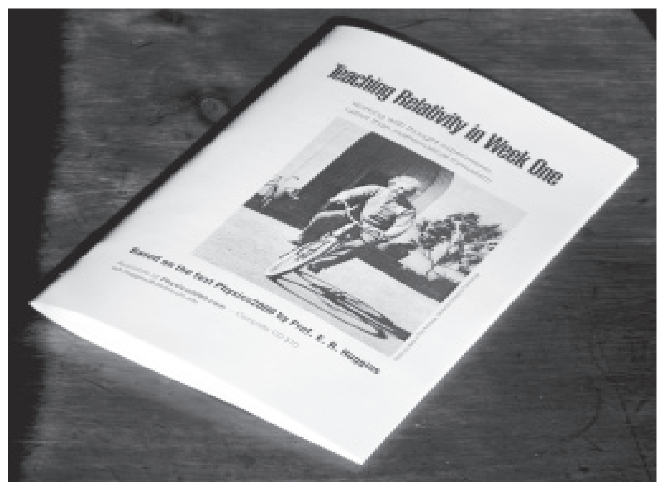
Teaching Relativity in Week One

Working with thought experiments rather than mathematical formalism

Aside from the fact that it is an exciting way to begin the course, the advantage of teaching Special Relativity at the beginning of an introductory physics course is that you can fit twentieth century concepts into the course as you go along.

The Teaching Relativity pamphlet provides a detailed outline of how to teach special relativity in week one using Chapter 1 of the Physics2000 text and the Muon Lifetime Movie that is on the Physics2000 CD.

Even if you do not use the remainder of the Physics2000 text, you will see the advantage of starting with special relativity no matter what text you use or what mathematical background your students have. All they need is the Pythagorean theorem.



Teaching Relativity Pamphlet

The first 23 pages of this 64 page pamphlet are a detailed outline of how to teach the basic concepts of Special Relativity in five 45 minute lectures. The remainder is a printed version of Chapter 1 of the Physics2000 text. Add to this the 36 minute Muon Lifetime Movie on the Physics2000 CD and you have all you need to start your course with Einstein's Special Theory of Relativity.

"The Physics2000 textbook introduces relativity early in the curriculum. This major difference from the usual introduction to physics provides an alternative to the "frozen curriculum" that hasn't changed despite all the changes in physics over the past century."
From the announcement of the Spring Meeting of the Michigan Section of the AAPT.

Teaching Relativity in Week One

Working with thought experiments rather than mathematical formalism

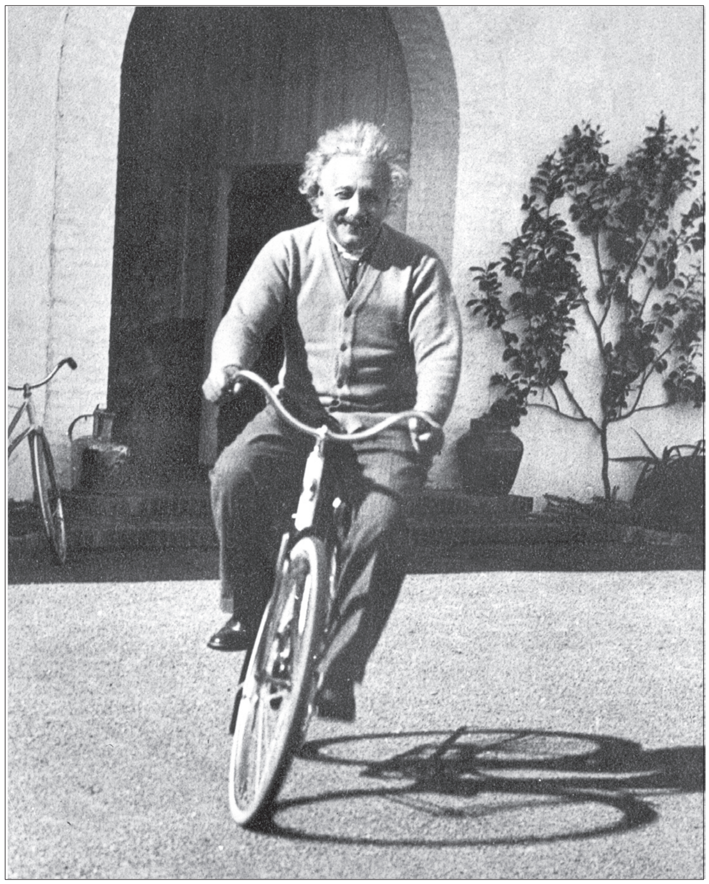
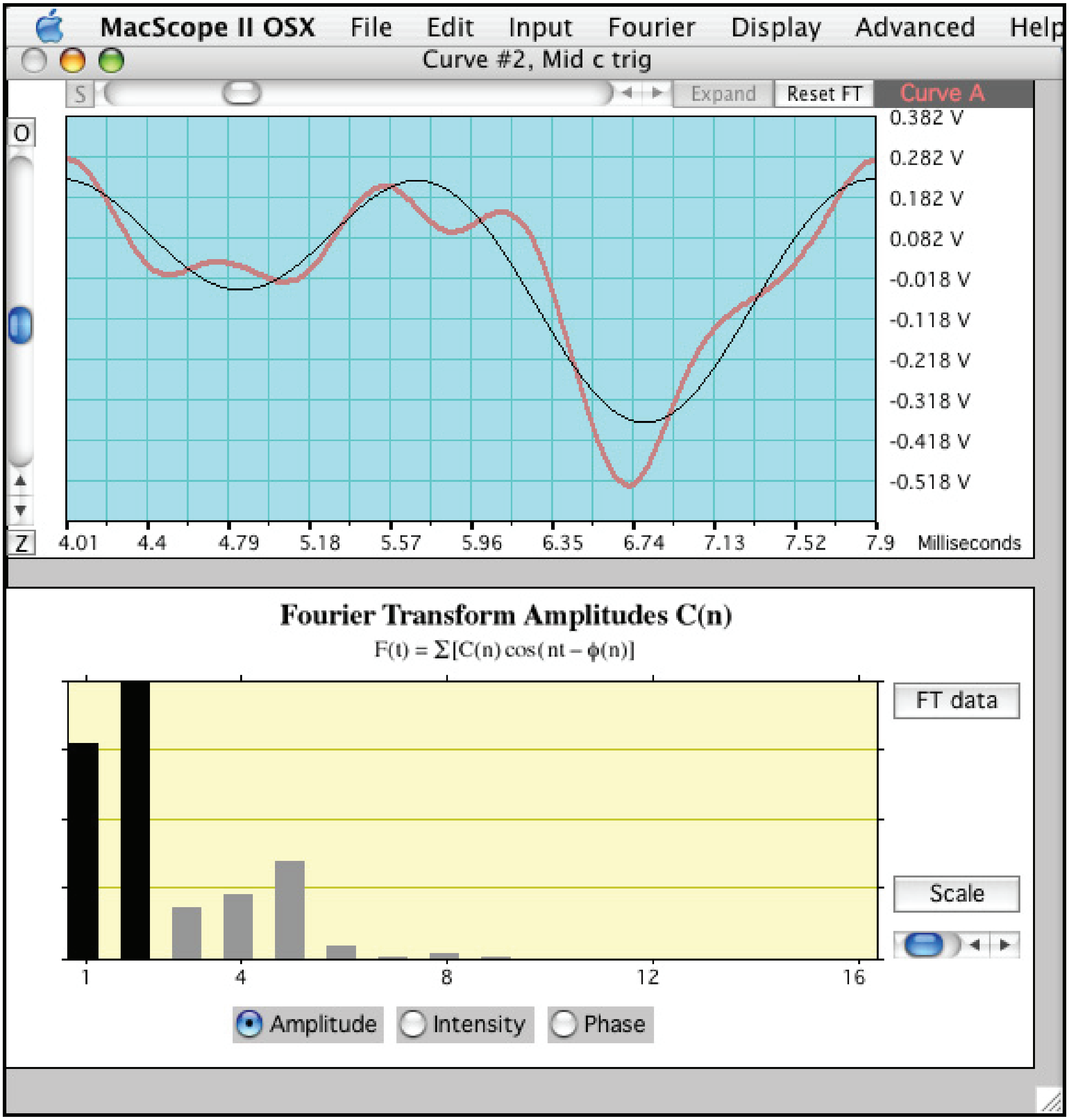


Photo courtesy of The Archives, California Institute of Technology

Based on the *Physics2000* text

Free download at
Physics2000.com



First two harmonics in a piano c note.

Moose Mountain
Digital Press

ISBN 0-9770828-0-6
90000
9 780977 082803