Music 302: EXCEL Project Status

Musical Acoustics Demonstrations

by

Rob Maher -- EE498, etc.

Submitted: May 14, 1986

INTRODUCTION --

This report concerns the current state of the EXCEL project software for Music 302: Musical Acoustics, and other software developed during the spring semester of 1986.

First, the version 1.0 demonstrations are described, and planned extensions are considered. Next, several demonstrations that are nearing completion are discussed. Finally, some software projects related to graphics, D/A/D, the FFT processor, and other areas are described.

An appendix section is included in this report. It contains brief users' manuals for the three operational demonstrations, version 1.0, as of May 1, 1986.

Note:

Demonstration #4, Membrane Vibration, was presented to the Music 302 class on May 12, 1986. It is not fully operational, and planned enhancements have made the preparation of a users' manual unwise. Demo #4 is located in \rcm\cprg\demo4\demo4.exe, and will run from the menu with static graphical displays.

Demo #1: Wave Phenomena

The first demonstration is intended to visually depict examples of transverse and longitudinal wave motion, collision, and reflection. The examples used are:

Standing waves (transverse and longitudinal)
Wave pulse collision
Wave pulse reflection
Plucked string motion

The demonstrations use animation to portray the motion of the medium. In the case of complex graphical representations, such as a dithered gray scale density mapping in the longitudinal standing wave example, the animation is implemented by manipulating entries in the graphics color lookup table. This technique uses images that are pre-drawn at various positions using different pixel intensity index values, then animated by activating sequentially the desired color index values. This method (when used with complicated graphics images) requires a "canned" demonstration, in that the time required to regenerate a complete image from scratch may be several minutes. Thus, the result is effective if a demo with fixed characteristics is acceptable.

A more straight forward use of the color lookup table for animation is to treat the image as 8 "planes", where each bit of the 8 bit color index controls one of the planes, i.e. either intensity-on or -off. In this way, each of the planes may be assigned a color from the palette of 4096 possible shades. An "invisible" image can be drawn in one of the planes by setting the color of that plane equal to the background color. The image can be made visible by simply changing the color assigned to that plane to a visible color. Overlaps can be handled by perfoming a logical OR, AND, XOR, etc. on the bits of the current pixel with the bits of the color index being drawn at that pixel. Thus, a separate image can be predrawn in each of the eight planes, then sequentially made visible to simulate motion. Also, during the time that one image is visible, one or more of the other planes can be updated invisibly to extend the animation sequence. However, this method only works effectively if the image can be drawn and updated rapidly enough for a useful rate of animation.

For rapid animation of simple images, two image planes is quite sufficient. This method is used for the pulse collision and reflection sequences, and for the plucked string. The calculation and drawing of the next position is carried out while the previous position is displayed, then the next position becomes the previous position and the cycle is repeated.

Planned improvements to the version 1.0 Wave Phenomena demonstration include more user input capability, and extension to two and perhaps three dimensional propagation.

For user input, it would be desirable for the user to specify a wave or pulse in any arbitrary shape using a mouse to "draw" it on the screen. Changes of wavelength, propagation velocity, etc. might also be useful.

Two dimensional images of wave interference and behavior near obstacles would be a logical inclusion ("ripple tank" types of experiments). More challenging programs could be conceived for generation of wavefronts in three dimensions, although the visualization might not be as effective as for the one- and two- dimension cases.

See the appendix of this paper for the users' manual about demo #1.

Demo #2: Fourier Series

The second version 1.0 demonstration is of Fourier series periodic waveform synthesis. The demo allows the user to specify the weight and initial phase angle for the fundamental and each of the first 19 partials. The resulting time-domain waveshape is displayed, along with a representation of the magnitude spectrum. Pre-calculated series for square, triangle, and sawtooth waves are available for direct display or editing of the various weights and phases. Input series are specified using polar { k*cos(wt+phase) } or rectangular { a*cos(wt) + b*sin(wt) } form.

This demonstration uses the FFT processor board to perform the waveform calculation: two buffer files are created containing the harmonic weights and phases converted to real and imaginary parts. The FFT processor is then instructed to perform an inverse transform on the real and imaginary frequency domain data to obtain the desired real time domain waveform. The result is displayed on the top half of the screen, with a bar graph representation of the spectral content on the lower half of the screen. A version of the demonstration using a software transform instead of the hardware is available in case the demonstration was used on a machine without the FFT processor.

Planned improvements for this demonstration include several ways to speed up the graphics output so that the time between specifying the harmonics and actually seeing the corresponding waveform is negligible. The use of the mouse to input harmonic weights might also be useful.

Combination of the graphical representation with actual audio waveform output is an important improvement. Completion of the D to A hardware and software is thus near the top of the list for work during the summer of 1986.

Further extensions could include a common link between the audio analysis demonstration (described as Demo #3) and this synthesis demonstration to allow for editing and resynthesis of sampled data. To facilitate these options, a common file format and command structure will be documented.

See the appendix for the Demo #2 users' manual.

Demo #3: Audio Spectrum Analysis

The final version 1.0 demonstration to date is a spectrum analyzer on sampled data using A to D input to obtain the samples and the FFT processor to obtain an approximation to the steady state spectrum of the data after windowing with a Hamming window function. The demo begins by repeatedly scanning the input channel and displaying the current waveform and peak amplitude. Once the user is satisfied with the samples, he may exit the data acquistion phase and view the calculated magnitude spectrum. This cycle of sample/view spectrum can be continued until the user decides to quit. Once the demonstration is ended, the last sequence of samples and their transform data is saved on disk for further display on the screen or hardcopy generation using the printer or plotter (see the manual for the details of this process).

At present, the analog to digital conversion process consists of obtaining 1024 samples at a fixed sample rate using the DMA facilities of the data acquisition board. The samples are windowed using a Hamming window of length 256. Thus, only the first 256 samples are actually used in the spectrum calculation. However, the FFT length used is 1024, so the effective FFT input sequence consists of the first 256 samples windowed to length 256, resulting in zero-padding out to length 1024. This manipulation is used to accomplish two things: First, to smear the spectral components (due to the window) and accentuate the narrow peaks relative to the background ripples in the frequency domain; and second, increase the density at which the spectrum is sampled by the FFT (due to the zero pad).

Multiplying by a window in the time domain is equivalent to convolving the Fourier transforms of the data and the window in the frequency domain. The transform of the data is the desired spectrum, and the transform of a Hamming window is sort of bell shaped with sidelobes. If no window function is employed explicitly, the input sequence can be thought of as a time sequence truncated by a rectangular window of height 1 and length equal to the sequence length. Since the discrete Fourier transform (obtained using the FFT algorithm) is actually a sequence of samples of the Fourier transform (equally spaced in frequency) of the continuous time input waveform, it is inherent that Gibbs ripples will be present in the spectrum due to the truncation of the input time sequence. The Hamming window has a transform that has twice the first-null bandwidth of the rectangular window, but sidelobes significantly smaller. The resulting convolution in the frequency domain smears any spikes in the data spectrum, but reduces the ripples due to the sidelobes of the window spectrum. The tradeoff between smearing the spectral features and reduction of background ripple determines the appropriate length of the window.

Zero padding is employed to increase the density of the samples of the Fourier transform present in the discrete Fourier transform. Adding zeros to the end of a sequence does not change the spacing of the input sample times relative to the input data, so the input frequency components do not shift relative to the sample frequency. Since the input spectrum is not changed by appending zeros but the number of samples of the spectrum has been increased by the same amount as the increased length of the transform, the density of the frequency samples per bandwidth must also increase. Note that the same effect could be achieved by repeating the original time sequence out to the new length rather than appending zeros, i.e. zero padding is simply a convenience. This also implies that no additional information has been created, only that a more accurate representation of the available information has been obtained at the expense of a longer FFT.

Extensions to the current demonstration are many and varied. The means by which a user samples an input signal will be improved to allow for specification of the sample rate (now set at an arbitrary fixed value), sample interval, interval start via threshold detection or user trigger (now continuous cycling), and a faster update of the current waveform display to make the system more useful as a crude digital oscilloscope. For the spectrum calculation and display, a means of measuring features using the mouse as a pointer will be examined. This would allow spacing of components or comparisons of magnitude to be determined by simply pointing to the appropriate positions on the spectrum. The ability to specify linear or log scales would also improve the versatility of the spectrum display. One simple improvement would be a single summary display containing a graph of the input waveform on the top half of the screen, and its spectrum estimate on the lower half of the screen (similar to the display used for the Fourier series display in demo #2).

As mentioned under demo #2, a means of editing and resynthesizing data using the D to A system and a common file format needs to be examined.

See the appendix for the users' manual describing demo #3.

Additional Demonstrations Not Yet Complete:

At present, two additional demonstrations are under development. The first is a 3-D representation of vibrational modes in a stretched membrane and other objects, and the second is a audio output system for comparison of various scale tunings and other phenomena. These demonstrations were planned for completion during the spring of 1986, but remain unfinished due to time spent on the other demos and software projects relating to graphics, modifications, and mathematical calculations.

The vibration demo presently allows the user to specify the desired mode of vibration for a rectangular membrane. The display consists of a 3-D perspective projection of the membrane, represented by a mesh, at an exaggerated maximum displacement position for the requested mode. Extensions contemplated for early completion are vibration of a circular membrane, animation of the motion, representation of multiple simultaneous modes, and image rotation for arbitrary view angle and view plane orientation.

The audio output demonstration will most likely consist of a series of C functions that can be organized into utility subroutines. Examples of functions and subroutines might include fixed-wavetable synthesis, playback of a recorded sample sequence, generation of test signals and noise, and so on. The first task will be to complete the output hardware and driver software to allow for sustained, continuous transfer of data to the D to A converters at a precise and reliable rate.

See later sections of this paper for descriptions of some of the implications and considerations needed for these and other demos.

Graphics Discussion:

At present, much of the graphics software used for the EXCEL demonstrations is written using the GKS package and device drivers for the graphics display, printer, and plotter. This has been done to try to keep the various programs and projects somewhat consistent and compatible. However, GKS is often too slow for animation and other purposes, and must be supplanted by specialized "home-brew" alternatives. Two alternatives are now available: direct use of the instruction channel of the Professional Graphics Controller, and a version of George Chaltas' G_RAPH package modified for the IBM PGC.

Direct use of the PGC functions is available in a primitive way using C and assembler functions found in the library \rcm\cprg\excel\pgc.lib . The functions have been written and extended as necessary to keep up with the needs of the demonstrations. The basic routines in the PGC library were described in the fall 1985 EXCEL Project report (Rob Maher, EE498). The use of direct calls to the PGC hardware results in the highest speed performance of which the controller is capable. A goal for future work would be to finish the PGC library by completing C and assembler code functions corresponding to each of the functions provided by the PGC hardware. In this way, a C program could use any of the features simply by linking to the pgc.lib library. This task will be worked on during the summer of 1986.

A version of the G_RAPH graphics package available on the LMC and VAX machines has been written for the IBM PC-AT and Professional Graphics Display. The modifications were detailed, but not a major overhaul: for the most part, the IBM G_RAPH works identically to the LMC version. Unfortunately, some changes were necessary to accomodate the restrictions of the Lattice C compiler used on the PC-AT, and the differences in the capabilities of the display devices. In particular, Lattice places an arbitrary limit of 8 on the number of significant characters in a function name. Hence, the function names like "g_initialize_graph()" must be reduced to "g_ini_gr()" or some other name. A tentative list of the name conversions has been set up, but changes in the interest of clarity will probably be made.

It would be possible to extend the IBM G_RAPH package to include some of the special capabilities of the PGC, such as colored lines, fill patterns, etc. Other possibilities are development of input capability using the mouse, and output capability to the color plotter and graphics printer. For the time being, these projects are probably of a low priority.

Analog-Digital-Analog Discussion:

The Metrabyte DASH-16 board has 8 differential input channels and 2 output channels. At this time, two of the input channels are wired to an external connection box, with a switch-selectable gain of 0 dB or 40 dB. Connection to the input channels is from 1/4" phone jacks. Provision for the 2 output channels has been made, also via phone jacks. The output level from the DASH-16 board is 0 to +5 volts, so a simple level shifter will be necessary to make a bipolar audio signal and add some drive capability. This task is of high priority and will be completed as soon as time permits.

In addition to the hardware, data management software is needed to provide for fast data sampling and D to A conversion. The software provided with the DASH-16 board was written with the proper protocol for execution as calls from a BASIC language program. Unfortunately, BASIC and C use different methods of passing parameters to subroutines, so modification of the software to accomodate this was necessary. In particular, BASIC uses only 64K of code and data space, while the Lattice C compiler has various memory models. All of the original software written for this EXCEL project had used the 1 Megabyte code and 1 MB data model, but this was incompatible with the DASH-16 software. The solution has been to compile a driver program using the C 64K/64K model to call the DASH-16 software, then use a 1M/1M model program to execute the driver using a "fork" command. Experiments will be necessary to find out the maximum transfer rates for D to A conversion -- from the disk or from a generation program -- and the limits on data manipulation during DMA transfer of samples from A to D conversion. Some simple tests indicate that sample collection may be possible (mono) direct to disk at rates approaching 30 kHz without too much trickery. The D to A rate will be lower, since DMA transfers are not available.

FFT Processor Discussion:

Like the Metrabyte DASH-16 software, the software provided with the Ariel PC-FFT board was written to be called from a BASIC language program. The way in which the assembler code was written made it difficult to use the software in the way the DASH-16 software was used, as described above. It was necessary to make extensive modification of the source code and re-assemble it for linking to a 64K/64K C program. Once the modification and re-assembly was done, the software worked as advertised when called from the C program.

One area of work will be to accelerate the graphics drawing speed so that the results obtained very rapidly by the PC-FFT processor can be displayed quickly. This would make the process of sampling, transforming, and displaying the data as near to real-time as possible.

Music 302: Musical Acoustics

EXCEL Demonstration #1

version 1.0

Wave Phenomena

USER MANUAL
May 1,1986

MUSIC 302 EXCEL PROJECT -- Demonstration Set #1

Wave Phenomena ver. 1.0

This demonstration contains five animated displays of standing wave behavior, pulse collision, pulse reflection, and a plucked string. This document is a simple users' manual for this demonstration, ver. 1.0, as of April 1, 1986.

PATH:

The demonstration is located on the PC-AT in directory:

```
\rcm\cprg\demo1
```

To get to this directory, type:

```
cd \rcm\cprg\demo1
```

(followed by the <ENTER> key)

RUN DEMO:

To begin the demonstration, type:

```
demo1 <ENTER>
```

MENU:

The options from the demonstration menu are selected by pressing the key at the top of the keyboard corresponding to the number from the menu (no <ENTER> is needed). After the key is pressed, the screen will go blank, and the setup operations for the selected option will be executed. To exit from a running demonstration, press the <ESC> key once or twice.

Note that the 'beep' tone will sound when exiting from the graphics display mode. This is part of the program, not an error indication.

OPTIONS:

<1>.......Transverse Standing Wave

The transverse standing wave display consists of three parts: at the top, a representation of the incident disturbance; in the middle, a representation of the reflected disturbance; and at the bottom, the actual motion observed when the right hand boundary is perfectly reflecting. An example would be an ideal string driven by a sinusoidal generator at the left, and rigidly clamped at the boundary on the right.

The motion can be stopped and started by pressing any key. Exit to the menu by pressing the <ESC> key once or twice.

<2>.......Longitudinal Standing Wave

The longitudinal standing wave display is of the same format as the transverse display, option <1> above. In this case, the disturbances are sinusoidal changes in density of the medium. The dark areas are where the particles are pushed together (compression), and the light areas are where the particles are spread out (rarefaction). This example could represent a disturbance in a liquid such as air or water, where the disturbance encounters a solid, non-absorbing boundary at the right end.

The motion can be stopped and started by pressing any key. Exit to the menu by pressing the <ESC> key once or twice.

<3>......Collision of Pulses

Two triangular pulses traveling in opposite directions on one string are depicted. The collision of the pulses occurs at the center of the screen, and then the pulses continue on in their original form (the demo stops shortly after the collision). The displacement of the string during the collision shows that a stop-action picture of the string at a particular instant in time is not enough to determine the possible combination of moving disturbances present on the string.

The motion can be stopped and started by pressing any key. Exit to the menu by pressing the <ESC> key once or twice.

<4>......Reflection of a Pulse

This demonstration shows the reflection of a triangular pulse at the ends of a clamped string. As the pulse reaches the end of the string, all the pulse's energy must be reflected, since the clamp is fixed and will not absorb any energy (ideally). Also, the fixed end position represents a boundary condition on string displacement: it must be zero at the clamp. The result is an inverted reflection at the ends. Conceptually, this property can be interpreted as an inverted "phantom" pulse moving in the opposite direction as the incident pulse, which collides with the incident pulse exactly at the clamp. (Compare the collision demo <3> with the reflection demo).

The motion can be stopped and started by pressing any key. Exit to the menu by pressing the <ESC> key once or twice.

<5>......Plucked String

The plucked string display represents an ideal string which can be displaced by a fixed distance anywhere along its length, then released. When this option is selected from the menu, the program will first request the position along the string where the pluck should be made (0 for the left end, up to 50 for the right). The displacement is made automatically by a small "hook". Once the string is released, a representation of two "phantom" disturbances moving in opposite directions is shown at the bottom of the display, while their resulting sum is the actual disturbance observed on the string.

Once the string has been released, the motion can be stopped and started by pressing any key. Exit to the menu by pressing the <ESC> key once or twice.

<6>......Exit from demo

Last but not least, the demonstration can be ended by selecting option <6> from the menu. This action returns the DOS prompt, and the PC-AT may be used as desired.

Music 302: Musical Acoustics

EXCEL Demonstration #2

version 1.0

Fourier Series

USER MANUAL

May 1, 1986

MUSIC 302 EXCEL PROJECT -- Demonstration Set #2

Fourier Series   ver. 1.0

This demonstration contains a data input and graphical output mechanism for Fourier synthesis of complex waveforms. This document is a simple users' manual for this demonstration, ver. 1.0, as of April 1, 1986.

PATH:

The demonstration is located on the PC-AT in directory:

```
\rcm\cprg\demo2
```

To get to this directory, type:

```
cd \rcm\cprg\demo2
```

(followed by the <ENTER> key)

RUN DEMO:

To begin the demonstration, type:

```
demo2  <ENTER>
```

MENU:

Options from the menu are selected by pressing the number key at the top of the keyboard that corresponds to the number of the menu item (no <ENTER> is needed).

OPTIONS:

<1>......Enter or Edit Series Data

The data for the Fourier series synthesis can be entered either as the amplitude and phase of cosine components, or as the amplitude of sine and components. Twenty components may be specified.

FORM 1: A*cos(nwt+theta)

FORM 2: a*sin(nwt) + b*cos(nwt)

where A,theta,a,b are functions of the component value, 'n'.

The program will request whether form <1> or <2> is the desired method of data presentation.

Next, the screen will show three columns of numbers. At the left is the component number, and the center and right columns are the amplitude(s) and/or angle according to which form of data entry was selected. If no previous data entry has been made, the data will all be zero. Otherwise, the data will be the values stored at the last data entry.

To enter or edit a component, use the arrow keys on the keypad to move the cursor up or down and from column to column. Once the cursor is at the desired component, start typing the new value using the number keys at the top of the keyboard, followed by <ENTER>. Then continue to enter or edit values as necessary. When the data are all correct, press the <ESC> key to return to the main menu.

<2>.......Display Waveform and Spectrum

Option <2> generates the waveform corresponding to the data input to the Fourier Series. After the calculations and graphics setup actions are complete, the display will show two cycles of the time waveform at the top of the screen, and the magnitude spectrum on the lower half of the screen. Press any key to return to the main menu.

<3>.......Clear Out Series Data

If an entirely new set of data is to be entered, the current data values can be erased by selecting this option.

<4>.......Load Data for Square Wave
<5>.......Load Data for Triangle Wave
<6>.......Load Data for Sawtooth Wave

These options load a pre-calculated data set for easy display and editing. Any current data present when one of these options is selected will be lost. After loading the data, choose option <2> to see the waveforms, or option <1> to edit the data.

<7>.......Exit from demo

Last but not least, the demonstration can be ended by selecting option <7> from the menu. This action returns the DOS prompt, and the PC-AT may be used as desired.

Music 302: Musical Acoustics

EXCEL Demonstration #3

version 1.0

Spectrum Analysis

USER MANUAL

May 1, 1986

MUSIC 302 EXCEL PROJECT -- Demonstration Set #3

Spectrum Analysis   ver. 1.0

This demonstration performs a spectral analysis on data sampled from a microphone or other input source. This document is a simple users' manual for this demonstration, ver. 1.0, as of April 1, 1986.

PATH:

The demonstration is located on the PC-AT in directory:

```
\rcm\cprg\demo3
```

To get to this directory, type:

```
cd \rcm\cprg\demo3
```

(followed by the <ENTER> key)

RUN DEMO:

If no graphics display has been made since powering up the computer, it is necessary to "prime" the system by typing:

followed by the <ENTER> key. After several seconds, a white waveform will be drawn. When the drawing is complete, press the <ENTER> key. The system is set for the demo. (The need for this process seems to have something to do with the fact that this demonstration uses a virtual RAM disk).

To begin the demonstration, type:

```
demo3  <ENTER>
```

OPERATION:

After some setup time, the display will appear as a dotted line being drawn repetitively across the screen. At the lower left corner, a numerical readout of the current peak amplitude of the input waveform is given. The program is designed to sample the input voltage from the microphone or other souce, display the waveform, erase the waveform, then sample again, and so on. The display will not show the input waveform if the peak amplitude is less the 50. The maximum peak amplitude is 2048, so adjust the input signal level to avoid clipping.

To freeze the waveform on the screen, press the <ENTER> key BEFORE the waveform begins to be erased. In other words, press the <ENTER> key when the drawing scan is about half way across the screen. Press <ENTER> again to resume scanning, or press <ESC> to perform a spectral analysis on the displayed waveform. Alternatively, the <ESC> key can be pressed without first pressing the <ENTER> key. This will go directly to the spectral analysis without freezing the waveform display first.

Once the analysis calculation is complete, the display will show the result of the analysis as normalized magnitude versus frequency. Press any key to exit the spectrum display.

At this point, the program will request whether to begin the input data scan again or exit the demonstration:
Press <ENTER> to begin the sampling again, or <ESC> to quit.

HARDCOPIES:

If a hardcopy of the waveform and/or spectrum is desired, EXIT THE DEMONSTRATION AT THE PROMPT AFTER THE SPECTRUM DISPLAY. When the DOS prompt returns, the following actions can be taken:

Type demo3
... to re-run the demonstration

Type pdat n
... to display the time waveform, where 'n' is the magnification. Try an 'n' value of 7 or 8 to start with.

Type plotdat n or prntdat n
... to send the waveform display to the plotter or printer, respectively. 'n' has the same meaning as for pdat.

Type pfft

... to display the spectrum on the screen.

Type plotfft or prntfft

... to send the spectrum to the plotter or the printer, respectively.

Remember that the plotter and printer must be on, online, selected via the port switch box, ...and have paper.

Once the demonstration is over and the DOS prompt is given, the PC-AT may be used as desired.