

EMERGE: The Merge File Editor

John Strawn

Center for Computer Research in Music and Acoustics

Stanford University

Stanford, California 94305

March, 1983

Revised 21 January 1985

21 Jan 1985 20:05 EMERGE.AWN[UP,DOC] PAGE 1-1

COMMENT • VALID 00078 PAGES

C REC	PAGE	DESCRIPTION
C00001	00001	
C00008	00002	
C00009	00003	Introduction: what does EMERGE do?
C00011	00004	Abbreviations in this document
C00013	00005	How to read this document
C00015	00006	Merge Files
C00019	00007	sample rate, compression ratio, channel numbers, N
C00021	00008	Menus types of menus
C00023	00009	Features common to all menus: α ?, cursor, αE , prompt area
C00026	00010	Running EMERGE the Main Menu
C00032	00011	Using a Selection Menu
C00034	00012	Dealing with Merge Files
C00035	00013	Using a List menu to read in or select a merge file
C00042	00014	Looking at the information in the merge directory
C00044	00015	Using a Parameter Menu
C00052	00016	The information in the directory
C00059	00017	Editing defaults for current merge file
C00063	00018	Create (averaged) spectral slice
C00069	00019	Looking at data in current input .MF file:
C00070	00020	Show One Channel
C00075	00021	Choice of amplitude or frequency or both
C00076	00022	Specifying screen coordinates
C00081	00023	Amplitude plots: linear or db? Specifying maximum amplitude
C00084	00024	Frequency plots: Channel Overlap
C00086	00025	Squelching frequency functions
C00088	00026	Including a label
C00090	00027	Making a .PLT file
C00092	00028	Specifying file names for plot files
C00094	00029	Three-Dimensional Plot
C00102	00030	Designing a 3-D Plot: Angle, Z-axis Spacing
C00109	00031	Freq. Display Range
C00111	00032	Spectrographic plot
C00114	00033	What does a spectrographic plot show
C00115	00034	Parameters for spectrographic plots
C00117	00035	Dealing with .FUN files
C00120	00036	CAVEATS and incompatibilities with CCRMA .FUN Files
C00122	00037	Write out .FUN file
C00130	00038	Read in .FUN file
C00133	00039	Mixing functions from the file with those inside EMERGE
C00136	00040	Ignored functions
C00138	00041	Function slots inside EMERGE
C00140	00042	Using function slots
C00143	00043	Looking at SEG functions
C00145	00044	Operating on data in current merge file:
C00146	00045	Pavlidis' Split/Merge Approximations
C00148	00046	The Menu
C00155	00047	Approximate amplitude function?
C00156	00048	Error Norm
C00157	00049	Threshold
C00159	00050	Debug File Name
C00160	00051	Storing the approximations in the A slot
C00162	00052	Output File Name
C00165	00053	The EMERGE function editor
C00169	00054	Edit Amp? (T = amp, F = freq)
C00170	00055	Edit previously Edited, Approximated, or Other functions?
C00172	00056	Using the EMERGE function editor
C00173	00057	Cursor moving commands: 7 8 9 U O J K L , M
C00175	00058	Moving along the functions: $\leftarrow \rightarrow$
C00176	00059	Inserting/deleting a new point: I, D
C00177	00060	Attaching a point: A
C00179	00061	Miscellaneous commands: α ., αE , $\alpha \beta E$, αP , αV , αW , $\alpha ?$
C00181	00062	Extended commands: αX
C00184	00063	EMERGE function editor LORE
C00186	00064	Storing the edited functions in the E slot
C00188	00065	Output File Name (in the menu)
C00190	00066	News
C00191	00067	Acknowledgments

C00192 00068 References
C00193 00069 Appendix 1: A Plaine and Easie Introduction to Merge Files
C00196 00070 Merge File Directories (for WIZARDs)
C00199 00071 Representation of merge file directories by the merge file software
C00203 00072 Appendix 2: Overview of files
C00207 00073 Appendix 3: Compiling EMERGE
C00212 00074 Compiling the high-segment version (starting with version 11.2) *****
* not yet done
C00214 00075 Appendix 4: Wizardry
C00215 00076 adding yourself to getWizard
C00216 00077 command line switches and the preloaded file
C00219 00078 wizardry in menus
C00220 ENDMK
C@;

Introduction: what does EMERGE do?

EMERGE is a graphics-oriented, window- and menu-based interactive editor for "merge files." Typically, merge files contain a large number of functions, often the result of time-varying Fourier analysis of sounds from traditional instruments.

As one would expect from an editor, EMERGE allows the user to

- find help when lost (self-documenting as much as possible)

- access merge files

- examine their contents in a variety of ways

- modify the merge file

- output (selected) portions of the merge files in a number of formats

In its final form, EMERGE will accept only .MF files. There are slots for expanding it to other kinds of merge files as well. I recommend that a new version of EMERGE be written for each kind of merge file, rather than trying to put all merge files into one program. On this system, that would be too large a program.

EMERGE was designed and implemented September 1982 - February 1983 by John Strawn (AWN) at CCRMA. Final revisions, January 1984.⁵

Abbreviations in this document

aka also known as

α aka control key --- hold down the key labelled "control" when you type the next character. For example, " α ?" means to hold down the control key, and type question-mark.

β ditto, but for the "meta" key

$\alpha\beta$ ditto, but for both control and meta together

• ditto, but you have the choice: control alone, or control and meta together

<alt> type the alt key

<break> type the break key, THEN the following key. For example, <break>N means to type N after the <break> key.

<bs> backspace

<call> the call key

<cr> or
<crlf> crlf

<ff> formfeed

<lf> linefeed

q. v. quod vide --- refer to the section just mentioned

<vt> the key labelled <vt>

sRate sample rate

How to read this document

The beginner should start with the section on "Merge Files" and read straight through to at least "Looking at Individual Channels" inside "Running EMERGE," skipping the sections marked "WIZARDS:" and "LORE:". Thereafter, the beginner should turn to individual sections of this document for information on particular features. For the ambitious beginner, this document is designed to be read cover-to-cover.

Useful tips, hints on how to exploit kludges, and similar items for the experienced user are marked as LORE.

A variety of information is sprinkled throughout this text, marked "WIZARDS," and is (obviously) for advanced users only. These comments can even include information on compiling extensions to EMERGE.

Merge Files

A merge file is a binary file containing one or more real-valued time-varying functions. For example, the phase vocoder (Gordon and Strawn 1985) produces a time-varying amplitude and frequency function for each "channel" of the analysis. The term "merge" originated in the following manner, according to a verbal account by James A. Moorer (20 November 1982): Each such amplitude or frequency function was stored in a separate file in an early implementation of the phase vocoder. Eventually it made sense to store all of these functions "merged" together into one file. (The entries for "name" and "extension" in the merge file directory make sense, seen in this historical light). The functions are not "interleaved" inside the file, as the name "merge" might imply. Rather, one function occupies one contiguous block inside the file. At the front of the merge file is a directory, which contains information on the function as well as its location inside the file. The details of the merge file directory will be invisible to the user of EMERGE. For more documentation, see Appendix 1, MERGE.JAM[UP,DOC] at CCRMA, MRGPAK.SAI[SUB,SYS] at CCRMA, and UDP2:MERGE.SAI[LIB,AWN] at CCRMA.

The phase vocoder in its current implementation at CCRMA produces a merge file known generically as a .PV file as its output, containing real and imaginary outputs. A number of programs can convert this format into another kind of merge file known as a .MF file, containing a magnitude/frequency representation. (For more information, see PROGS.JAM[UP,DOC] at CCRMA). Currently, EMERGE can only be used to look at ".MF" files.

WIZARDS: hooks are available in the code for incorporating other kinds of files, such as .PV files as well as the output of the PITCH program and Zwicker analysis. See the "includePV" compile-time switch in MERGE.HDR[LIB,AWN]; setting this to TRUE includes code throughout MERGE and EMERGE (still only a torso) for working with .PV files.

sample rate, compression ratio, channel numbers, N

There are three items of interest to the user of EMERGE contained in the .MF file directory. Other information about the directory is found in Appendix 1.

1. The sampling rate sRate at which the original signal was recorded.
2. The "compression ratio." To paraphrase MERGE.JAM[UP,DOC]: the compression ratio represents any decimation in time that may have taken place in the computation of this function. In other words, the original signal was represented by sRate samples per second. In the .PV (and .MF) file, the signal is represented by points every sRate/(compression ratio) seconds.
3. The functions in the .MF file are divided into magnitude (aka amplitude) and frequency functions, and numbered according to channel number. The phase vocoder numbers its channels starting with 1.

N is the number of channels as calculated by the PVCOMP program. Unfortunately, N is not passed explicitly to the .MF file (at least not yet). When you run PVCOMP, keep a careful record of N. Although EMERGE can "guess" what N is, you will need to know exactly what N was in order to make the plots of the frequency functions work right. See also "Editing defaults for current merge file".

Menus; types of menus

The user communicates with EMERGE in general by means of menus. When the user selects an option, a menu is displayed on the screen.

Types of menus

There are 3 kinds of menus:

selection menu --- used to select from one of a number of options.

The use of this kind of menu is explained in the section "Using a selection menu". You will see this menu when you first run EMERGE.

parameter menu --- used to type in control parameters, such as channel number, begin time, and the like. This kind of menu is explained in "Looking at a merge file directory."

list menu --- used to add, modify, or delete elements in a list. See "Reading in a merge file."

Features common to all menus: $\alpha?$, cursor, αE , prompt area

In general, the menus in EMERGE have been designed to use the same control character as the editor E in common use at CCRMA. Experienced E users should try their favorite E commands without fear of making EMERGE blow up. The few extensions to E-style commands are noted in the documentation internal to program and of course here.

All of the menu types have several features in common. The "prompt area" is moved to the bottom three lines on the screen. As in E, if you've done something right, EMERGE will respond with "OK" in the prompt space. Otherwise, EMERGE will complain there. If some error message is flashing by too fast, type <break>N, which clears the screen, and the message will appear (and be retained) on a clean screen. To restore the menu, type αV .

All of the menu features commands to move the cursor, drawn with the character "->".

All of the menus feature a two-level help scheme. If you type $\alpha?$, the screen will display a window containing help for that specific menu. If you type αE , the original menu will be restored. But if you type $\alpha?$ again, you will get help on the current KIND of menu. This second help level will tell you about cursor-moving and line-editing commands, for example. Again, type αE to leave the help. If the help occupies more than one screen, use <vt> and <ff> to move up and down.

In all menus, αE (for "exit") returns you to the next-higher level of the program. In EMERGE, this means that you will return to the "Main Menu" if you're not there already. From the "Main Menu", αE returns you to the monitor.

Some menu features will be explained in this document. The beginning user of EMERGE should study the documentation included in EMERGE itself, and also MENU.AWN[UP,DOC] at CCRMA (included as an appendix in "hard copy" of this document).

Running EMERGE; the Main Menu;

When you type

.R EMERGE

to the CCRMA monitor, the screen will be cleared, and you will see a menu, known as the "Main Menu", which looks as shown in Figure 1.

Welcome to EMERGE, the Merge File Editor (Version 13.3)

Dealing with Merge files:

- Get current input merge file
- Print directory of current input merge file
- Edit defaults for current merge file

Looking at data in current input .MF file:

- Show 1 channel
- 3-d plot
- Spectrographic plot

Operating on data in current merge file:

- Pavlidis' Split/Merge Approximations
- EMERGE Function editor

Dealing with .FUN files:

- Read in .FUN file
- Write out .FUN file

EMERGE miscellany:

News

- α means to hold down the "ctrl" key while typing the next character
- $\alpha?$ for help
- αE to exit to next higher level
- αG to execute option pointed at by cursor"

Figure 1. The Main EMERGE menu.

EMERGE Version numbers are changed whenever EMERGE is modified. A new number to the left of the decimal point means that a feature has undergone a major rewrite, and/or a new feature has been added (probably to the Main Menu). Any changes are recorded in the "News" option.

When you have successfully input a merge file (see "Reading in a merge file", below), the line

Get current input merge file

in the Main Menu will be changed to include the name of the "current" file (again, see "Reading in a merge file"). This is the file with which you are currently working. Opening a file is probably the first thing you'll do.

WIZARDS: The main menu is assembled according to the settings of compile-time switches in EMERGE.SAI. There are 5 standard master compile-time switches there:

systemEMERGEVersion	set this to TRUE to get all of above features
EMERGEVersion	set this to TRUE for debugging one or more features
mfEDVersion	just the function editor and a few other things
mf2EDVersion	for editing two (!) .MF files and two .FUN files together
mfApprVersion	just for line-segment approximation and a few other things

Each of these switches turns on or off the following compile-time switches:

doShow	Show 1 channel
--------	----------------

doThreeD	3-d plot
doThreeD2	3-d plot for 2 funcs from two .MF files
doSpect	Spectrographic plot
doSplMer	Pavlidis' Split/Merge Approximations
doEdSeg	EMERGE Function editor
doedSeg2	EMERGE Function editor for 2 funcs
doSlice	slice of life
doEfl	Get current input merge file
doDir	Print directory of current input merge file
doFDef	Edit defaults for current merge file
doInFun	Read in .FUN file
doOutFun	Write out .FUN file
doMfDif	difference between .MF and .FUN file
doNews	News

These switches, in turn, determine which features are loaded into the main menu. Some of the features controlled by these switches were never released publicly.

There is also the switch

DEFINE breadboard = TRUE;

in EMERGE.SAI which turns off the main menu completely. This is used only for debugging.

Using a Selection Menu

Each line in the Main Menu contains a short description of some EMERGE feature. Move the cursor to the line describing what you want to do, and type αG . A menu for that feature will then appear on that screen. When you are done with that menu, type αE and you will be returned to the Main Menu. Type αE to exit from the Main Menu to the monitor.

Here is a summary of the commands available for Selection Menus:

Cursor-moving commands:

$\langle cr \rangle, \langle lf \rangle, \alpha \downarrow$	move to next line
$\langle bs \rangle, \alpha \uparrow$	move to previous line
$\alpha <, \alpha >$	up, down 4 lines
$\alpha \leq, \alpha \geq$	up, down 16 lines
$\alpha L, \alpha \wedge, \langle vt \rangle$	move to top of menu
$\alpha v, \langle ff \rangle$	move to bottom of menu
αN	move to the line you were just at (toggles)

αN does not update what it thinks was the "old" line unless you move more than one line. This is slightly different from αN in E.

Some other commands:

αE	exit to next higher level (possibly to monitor)
$\alpha ?$	help
αV	redraw screen (useful after $\langle break \rangle N$)
αG	invoke the feature currently pointed at by the cursor

All of these commands are from E, with the exception of αG .

21 Jan 1985 20:05

EMERGE.AWN[UP,DOC]

PAGE 12-1

Dealing with Merge Files

Using a List menu to read in or select a merge file

Probably the first thing which the user will do is to read in a merge file. The menu for this is a List Menu.

Using a List Menu

The list menu is divided into two parts: the file name prompt at the top of the menu, and a list (in this case, a list of file names). List menus have not been publicly released as part of the menu package, so there is no documentation in MENU.AWN[UP,DOC] on list menus.

No matter which line the cursor points at, you can enter the prompt area simply by starting to type, which will invoke the system line editor. Use the prompt area to type in a new file name. When you are done, activate the line editor with <crlf>, and the new name will be added at the top of the list.

The cursor ">" may be moved with the cursor commands given under "Using a Selection Menu." You can modify a file name already in the list with the following three commands:

$\alpha\beta D$	delete the file name currently pointed to by the cursor
$\alpha\beta A$	(attach) delete the file name currently pointed to by the cursor, and move that file name to the prompt area for editing
$\alpha\beta C$	(copy) enter the prompt area with a copy of the file name currently pointed to by the cursor.

When you type αE to the list menu, EMERGE will attempt to find the currently marked file. Whenever you type in a new file name to the list menu, the new name is automatically marked. Also, αM "marks" the file currently pointed at by the cursor. If you delete the currently marked file, the list menu "does the right thing," which means to mark the next-higher (or top) file in the list. If for some reason you want to return to the Main Menu without EMERGE looking for an input file, type $\alpha\beta E$ (known as a "hard exit") to the list menu. Plain old αE will still return you to the Main Menu even if the input file list is completely blank.

If the marked file has already been looked up and opened, then no further work is necessary and you are immediately returned to the Main Menu. In other words, you can have several files open at once, and come to this menu to switch back and forth among them.

If the file as you specify it cannot be found (in particular, if you have not specified an extension), EMERGE will try the following extensions for you (in this order): .MF, .MRG, and <none>. If EMERGE cannot find the marked file, EMERGE will return to the list menu, with the offending file name deleted from the list and moved to the prompt area for your editing convenience.

If EMERGE can find the file, then it will attempt to "determine" what kind of merge file it is. Currently, .MF files are the only kind which can be examined with EMERGE. If the merge file is of an unknown type, EMERGE will return to the list menu, as explained in the previous paragraph.

As with parameter menus, you may use αR or αW to read in/write out an entire list of merge files. The file written out contains documentation on the required format.

LORE: There is a limit to the number of files which can be open for input or output at any one time. If you read in many files using the list menu and then try to write out or look at other files elsewhere within EMERGE you may get an error message. The solution in this case is to come back to the file list and delete a file. In some cases, a file which you had opened may be closed as an EMERGENCY solution, in which case you'll be so informed.

WIZARDS: MERGE.HDR[LIB,AWN] contains two compile-time switches "includePV" and "includeZw." These control whether

code for .PV files and .ZW files is included in MERGE.SAI AS WELL AS in EMERGE itself. At current writing, quite a bit of code necessary for working with .PV files (reading directories, looking at one function from the file ...) has been written.

WIZARDS: You can rename a file with impunity. The "function name" is ignored while the directory is being parsed.

WIZARDS: I no longer use the system MRGPAK [SUB,SYS]; MERGE [LIB,AWN] includes only the parts that I need.

Looking at the information in the merge directory

Each merge file contains a directory, which will have some useful information in it for the user of EMERGE. The parameter menu shown in Figure 2 controls what part of the directory is displayed.

Parameters for printing directory of .MF file UDP1:CA3TA.JMF [CLR,AWN]

Sample rate for this file = 25600
Compression Ratio = 29
Each function is 1852 points long (end time = 2.0968359 sec)
Maximum amplitude = 7.5116417
N currently set to 115

First Channel	1
Last Channel (58)	58
Write out onto TTY? (T or F)	TRUE
Output File Name	(CA3TA.DIR/q)

Figure 2. The parameter menu for looking at a merge file directory.

LORE: The file CA3TA.JMF will be used throughout this document. It contained phase vocoder analysis of the clarinet tongued transition (rising major third) that I used in my thesis.

Using a Parameter Menu

A parameter menu is divided horizontally into two parts: The lines on the left side of the menu give a very short description of a parameter, and the current value of the parameter is on the right.

The user can move the cursor with the same commands as given under "Using a selection menu." If anything besides one of the cursor-moving commands is typed, then the line editor is invoked for the VALUE pointed to by the cursor. <crlf> terminates editing of that value.

LORE: It is possible to invoke any cursor-moving or extended command from within the line editor. Many such commands are illegal within E's line editor.

When no more changes need to be made to the parameters, type αG and the function controlled by the parameters (in this first case, printing out the merge file directory) will be invoked. If some values are invalid, you may be returned to the parameter menu before the function is invoked. This can happen, for example, if you type an invalid output file name. When the function is completed, you will likewise be returned to the parameter menu, so that you can make further changes and execute the function again.

Any parameter menu in EMERGE is initialized to reasonable default values for the merge file in use. Whenever you enter a parameter menu for the first time, you can type αG and get some reasonable results.

As with all menus, αE returns to the next highest level, in this case EMERGE's Main Menu.

Most of the parameter values in EMERGE's parameter menus are TYPed. This typing is explained in the MENU documentation. In general, if you type, for example, a file name where an integer is expected, the menu package will complain. You can turn off this type checking with αX T.

LORE: EMERGE supplies you with default file names for plot output files and the like. In many cases, you can keep the output from happening by starting the file name in the menu with "(", or else by blanking out the file name completely.

WIZARDs: for such entries, the type in STRING ARRAY menu is not set to fileNameType (See the MENU documentation).

WIZARDs: default file names are initialized with padded spaces to allow for EMERGE adding the full ppn once the file name has been checked.

There are many other features of parameters menus, which are explained in the menu documentation. In particular and quite briefly, αX Z (for "zero") will zero out all of the value entries in the right-hand side of the menu. As in E, αX C ("cancel") will cancel all of the changes since the last α. ("save edits"). You can write out a text file with the entire menu using αW, and read in such a file with αR.

LORE: EMERGE "keeps track" of the information in a parameter menu when you go to another parameter menu. When you come back to the first parameter menu (without switching input files), the most recent values will have been saved. This operates on a file-by-file basis (!), so that each parameter menu for each file is saved, until you explicitly get rid of the input merge file using the menu for that purpose.

LORE: EMERGE supplies "additional" information for some of the entries in the left-hand side of the menu. For example, an entry "Channel Number" may be changed to "Channel Number (1:42)" for the current merge file, showing the range of channels available. If you use the menu's αW feature to

write out the current state of the menu to a file, this additional information is also written out as part of the text file. If you try to read in the same text file (using αR) for another merge file, which may only contain channels number 5:33. The parser reading in the values from the text file will not update the value for the line "Channel Number (5:33)" because in the text file it finds "Channel Number (1:42)". The solution here is to remove the "(1:42)" from the text file. The command parser will first try for an exact match of the entry "Channel number" in the text file. Since that will fail, it will then continue and try to find a match between the shorter "Channel Number" and the longer "Channel Number (5:35)", which will succeed, because the first n characters in both match (ignoring case), where n is the length of the shorter.

LORE: EMERGE supplies a different default extension for each of the parameter menus as an extension when writing out the menu to a text file. Each such default extension ends in "C" (for "command"). Here is a list:

.DRC	looking at directory
.FDC	defaults for current merge file
.SHC	show one channel
.3DC	3-d plot of file
.SPC	spectrographic plot
.SMC	Pavlidis Split/Merge approximation
.EDC	Edit functions
.IFC	input .FUN file
.OFC	output .FUN file
.SLC	spectral slice

The information in the directory

Certain important information about the file directory is displayed as part of the parameter menu: sRate, duration, and N. Such information is also repeated when the directory is printed.

For amplitude functions in .MF files, WD1 shows the maximum amplitude of the current channel. Another column in the printed directory shows this maximum amplitude in dB relative to the maximum amplitude in the entire file. For frequency functions, WD1 gives the average frequency across the entire function, which may or may not be close to the "center frequency" of the harmonic.

Figure 3 shows a sample directory.

Directory of .MF file UDP1:CA3TA.JMF [CLR,AWN]:

Function Name	CA3TA
Channels	1:58
Clock	25600
Length	1852 points (end time =2.0968359 sec)
N (from file)	115
Compression Ratio R	29
Intermediate	
Compression Ratio Q	0
Maximum Amplitude	7.5116417
Frequency Wd2	242.7401100

Chan	Amp Wd1	(dB)	Freq Wd1
1	7.51164	.00000	242.74011
2	.14878	-34.06367	454.88728
3	2.47610	-9.63932	622.92038
4	4.51187	-4.42757	862.52072
5	6.71560	-.97300	1102.55120
6	5.17198	-3.24157	1353.85780
7	.69295	-20.70069	1590.41330
8	1.36953	-14.78329	1726.98180
9	.77132	-19.77002	1971.43990
10	.62116	-21.65067	2213.46450
11	.65044	-21.25058	2448.77020
12	.18846	-32.01031	2693.18680
13	.44999	-24.45060	2844.38030
14	.19215	-31.84176	3082.10290
15	.35678	-26.46678	3321.71880
16	.18023	-32.39807	3560.41950
17	.05077	-43.40177	3784.16020
18	.12956	-35.26510	3949.89990
19	.12149	-35.82414	4183.39390
20	.05877	-42.13103	4425.10130
21	.08009	-39.44263	4669.57460
22	.06055	-41.87196	4872.63090
23	.05986	-41.97164	5049.66600
24	.03539	-46.53762	5298.96800
25	.05771	-42.28911	5534.39420
26	.03737	-46.06395	5762.32060
27	.01966	-51.64301	5965.61680
28	.01341	-54.96847	6165.76370
29	.02142	-50.89766	6399.30180
30	.01432	-54.39847	6640.71230
31	.01365	-54.81354	6864.57780
32	.00583	-62.19799	7073.80540
33	.02336	-50.14522	7266.05460
34	.02183	-50.73457	7502.47970
35	.00742	-60.10554	7747.13720
36	.00726	-60.30082	7962.80600
37	.00500	-63.53144	8169.82580
38	.00647	-61.30045	8381.86050

39	.00473	-64.00848	8614.15710
40	.00462	-64.21405	8849.95080
41	.00330	-67.13135	9058.85880
42	.00360	-66.37890	9272.42190
43	.00502	-63.50099	9481.45560
44	.00526	-63.09442	9712.20060
45	.00537	-62.91944	9952.41080
46	.00455	-64.35875	10173.65500
47	.00395	-65.59126	10374.21500
48	.00442	-64.60616	10582.37300
49	.00405	-65.37231	10807.81700
50	.00271	-68.85444	11045.26500
51	.00179	-72.47867	11264.39500
52	.00178	-72.52418	11471.87500
53	.00138	-74.68634	11684.09700
54	.00141	-74.51775	11907.67200
55	.00157	-73.61157	12140.77700
56	.00107	-76.95265	12353.64100
57	.00108	-76.83566	12577.71600
58	.00115	-76.29001	12739.23200

Figure 3. A merge file directory as printed by EMERGE.

LORE: For certain display features of EMERGE, you will need certain information from the directory. For example, in looking at individual channels from a .MF file, it will often be useful to know the maximum amplitude of the channel you're currently looking at (WD1). If you plan to make a large number of such plots, then print out a hard copy of the directory first.

WIZARDS: See Appendix 1 for the meaning of the data in the merge file directory. If you want to see all of this information, use PROCEDURE longFormDirectory instead of the current PROCEDURE shortFormDirectory (in MERGE.SAI).

Editing defaults for current merge file

One of the LORE entries under "Using a Parameter Menu" explains that for each merge file input using the list menu, information is retained inside EMERGE for each of the parameter menus. If you have not read that LORE yet, the rest of this section will not make sense.

Defaults for the MF file UDP1:CA3TA.JMF [CLR,AWN]

N	115
"Function name" default when writing out .FUN files	CA3TA
(we'll supply a "." after this name for you)	

Figure 4. The parameter menu for specify defaults for the current input merge file.

In this parameter menu, you can specify some defaults that apply to the current input merge file, no matter what you're doing to that file.

1. N is explained in the introductory section on merge files. This setting of N is used throughout EMERGE wherever N is needed.

WIZARDS: I actually store N inside
mrgClass:N[EMERGEClass:sonRp[EMERGERp]], aka
mrgClass:N[mrgRp]. Whenever any parameter menu is invoked
or initialized, it looks at THAT value.

2. Function name: You can specify a DEFAULT name used in outputting .FUN files. .FUN files output by EMERGE contain functions of the form "name.A1", "name.F1", or just "A1", "F1" if no "name" is supplied by you. EMERGE supplies the "." for you.

LORE: This default is used 1) when you write out functions from inside the EMERGE's function editor (using αW); 2) you exit from EMERGE's function editor and you have specified a file name, so that the entire set of edited functions is immediately written out to a file; 3) you have finished a function using the Pavlidis approximation, and you have specified a file name so that the entire set of approximated functions is immediately written out to a file.

LORE: When you want to write out a .FUN file using the parameter menu for that purpose, the DEFAULT name you supply here will be supplied initially; but you can supply another name in that parameter menu. That name supplied in that menu WILL NOT overwrite the DEFAULT name supplied in this menu.

LORE: When you input a new current file and enter this menu for the first time, the default file name will be set to the file name (without extension).

Create (averaged) spectral slice

The menu for this option is shown in Figure 5.

Parameters for printing "averaged spectrum" of
MF file UDP1:CA3TA.JMF [CLR,AWN]

Sample rate for this file = 25600
Compression Ratio = 29
N currently set to 115
Each function is 1852 samples long (end time = 2.0968359 sec)
Maximum amplitude = 7.5116417

First Channel	1
Last Channel (58)	58
Beg time	0.1
End time	1.9979688
Use merge file maximum (given above)	
to calculate dB? (F = use maximum found)	FALSE
Write out onto TTY? (T or F)	TRUE
Output File Name	(CA3TA.AVG/Q)

This option allows you to average the amplitude and frequency functions. The notion of "slice" can be explained as taking a slice IN time ACROSS the specified merge file channels. If you wish to see such a spectral slice just one sample wide, then specify identical begin and end times. This EMERGE option is intended as a kludge for getting a rough idea of the spectral content without having to take an FFT with another program, e.g. with EDSND.

The results of this operation look as follows (this is modelled roughly on the output of EDSND's "α3D" option---this shows only 10 channels):

Averaged "spectrum" of .MF file UDP1:CA3TA.JMF [CLR,AWN]:

Channels 1:58
Average taken over .500: .600 secs
(= 89 points from merge file)
(= 2581 points at original sampling rate)
(= approximately 22 periods, assuming fundamental of 222.609 Hz.
calculated from sRate = 25600, N = 115, compr = 29)
Maximum Amplitude 7.426

Chan	Amp	(dB)	Freq (Hz.)	freqN/(freq1*N)
1	7.42632470	.000	221.7419	1.00000
2	.13224537	-34.988	443.4697	.99997
3	2.42749880	-9.712	665.2231	1.00000
4	.36798792	-26.099	887.0356	1.00008
5	6.45843060	-1.213	1108.6799	.99997
6	1.11601680	-16.462	1330.3849	.99995
7	.54495770	-22.688	1552.0290	.99989
8	.64140432	-21.273	1773.9091	.99999
9	.18399553	-32.119	1995.2808	.99980
10	.41242434	-25.109	2217.3609	.99997

dB calculated relative to maximum amplitude shown above.

freq1 is the frequency of the first channel.

N is the channel number; freqN is the frequency for that channel.

The header gives information intended to be useful in interpreting the data: the number of points from the original merge file, the corresponding numbers of samples from the original recording analyzed with the phase vocoder; and the number of periods of the "fundamental" of the waveform included in the average. The fundamental given here is calculated by dividing sRate/N. "Maximum amplitude" is the amplitude used

21 Jan 1985 20:05

EMERGE.AWN[UP,DOC]

PAGE 19-1

Looking at data in current input .MF file:

Show One Channel

Here begins the real usefulness of EMERGE. Figure 6 shows the parameter menu for examining the channels of a .MF file one at a time.

Parameters for showing one channel from MF file UDP1:CA3TA.JMF [CLR,AWN]

Draw amplitude? (T or F)	TRUE
Draw frequency? (T or F)	TRUE
Amplitude plot:	
Amp x0 (in pixels, 0-511)	0
Amp y0	48
Reference amplitude = maximum of channel?	TRUE
If not, use this reference amplitude (=7.5116417)	7.512
Use dB for amp plot? (F = linear, T = dB)	FALSE
dB Range	50
Frequency plot:	
Freq x0	256
Freq y0	48
Squelch frequencies? (T or F)	FALSE
Squelch frequencies to this frequency:	242.7401100
When the amplitude is this many dB down:	50
Freq. Channel Overlap (0.0 to 1.0)	1.0
Parameters for both kinds of functions:	
Channel No (1:58)	1
Beg time	0.0
End time (2.09)	2.0968359
Also show Edited, Approximated, or "Other" functions? (Slot E, A, or O; N or F = none)	N
Show merge function?	TRUE
If so, draw it with lines? (F = dots)	TRUE
Window Width (in pixels)	256
Window Height	428
Label	CA3TA.JMF [CLR,AWN]
Include label (T or F)	TRUE
Plot File Name	(UDP1:CA3TA.PLT/q)

Figure 6. The parameter menu for examining individual channels of a .MF file.

When you type αG to this menu without changing any of the default settings, EMERGE will show you both the amplitude and the frequency function for the first harmonic.

The explanation for this parameter menu will be fairly complete, as the same parameters occur for a number of other parameter menus. Those items which are self-explanatory will not be discussed further here. The prompt for "Also showing ... functions" and the prompt for choosing between lines and dots will be explained in the section "Looking at functions" under "Dealing with .FUN files".

21 Jan 1985 20:05

EMERGE.AWN[UP,DOC]

PAGE 21-1

Choice of amplitude or frequency or both

You can set the boolean variables in the first two lines to see just the amplitude trace, or just the frequency trace, or both. Yes, if you turn them both off, EMERGE will still not blow up. No extra charge.

Specifying screen coordinates

This section deals with the following lines scattered around the parameter menu:

Amp x0 (in pixels, 0-511)	0
Amp y0	48
Freq x0	256
Freq y0	48
Window Width (in pixels)	232
Window Height	428

The terminal screen is divided into physical coordinates (pixels) numbered 0--511. You can put the amplitude and frequency plots anywhere you like on the screen. Each plot is placed inside a "window" (for more information, see GRNLIB.BIL[MUS,DOC]).

The (x0,y0) values give the physical coordinates for the lower-right-hand corner of the amplitude (or frequency) window. In the example above, the amplitude window will start at (0,48), meaning that it's on the left side of the screen, placed toward the bottom. The frequency window starts at (256,48), meaning that its left-most edge is half-way across the screen.

The "width" and "height" terms affect BOTH windows at once. A width of 232 is a little less than half the screen (half the screen would be $512/2=256$ pixels). This allows for a margin between the two windows.

LORE: The bottom edge of both windows is placed at 48 to allow room for the prompt area and label. The height is initially set to 428 to leave room for the WHO line.

You can set the (x0,y0) and (window,height) values to place the windows anywhere desired on the screen. If you specify some "wrong" values, you may get a funny-looking plot, but EMERGE should not blow up at you. These values, for example, will spread both the amplitude and frequency functions out to full screen width, and place them on top of each other.

Amp x0 (in pixels, 0-511)	0
Amp y0	48
Freq x0	0
Freq y0	275
Window Width (in pixels)	511
Window Height	200

LORE: You can overlap the functions (!). These settings, for example, will draw both functions at full screen width, and overlap them so that all of the times match up:

Amp x0 (in pixels, 0-511)	0
Amp y0	48
Freq x0	0
Freq y0	48
Window Width (in pixels)	511
Window Height	428

When you do this, the axis labels will get confused ...

WIZARDS: One character printing on the screen takes up 6 horizontal pixels. One line on the screen takes up 12 vertical pixels. The height is derived as follows: $512 - (12 \text{ pixels per line} * 3 \text{ lines for the WHO line}) - (12 * 3 \text{ lines for the prompt area}) = 428$.

Amplitude plots: linear or db? Specifying maximum amplitude

This section deals with the following lines from the parameter menu:

Reference amplitude = maximum of channel?	TRUE
If not, use this reference amplitude (=7.5116417)	7.512
Use dB for amp plot? (F = linear, T = dB)	FALSE
dB Range	50

If you set the boolean in the third of these lines to "TRUE", the amplitude function will be shown in decibels. The settings show here would give a 50 dB range, with 0 at the top of the y-axis (for the maximum amplitude) and -50 at the bottom of the y-axis.

For either linear or decibel plots, you need to specify the maximum amplitude (aka reference amplitude) for calculating the plot. Linear plots are scaled from 0 - Max. Decibel plots are calculated using Max as a reference amplitude. The first time you enter this menu (for the current input merge file), this value is initialized to the maximum amplitude in the entire merge file (as given by the WD1 entries for the amplitude functions in the merge file directory); this same maximum value is shown in parentheses in the menu. For higher-numbered harmonics, it will probably be necessary to reset this maximum value, just to get a reasonable-looking amplitude plot.

LORE: When you type αG to this menu, you will be reminded (in the prompt area) of the maximum amplitude for the current channel. An easier way is to get this information is to use the appropriate parameter menu and print out a copy of the merge file directory, which will contain the maximum amplitude value (WD1) for each channel.

Frequency plots: Channel Overlap

This section deals with the following line from the parameter menu:

Freq. Channel Overlap (0.0 to 1.0) 1.0

"Channel Overlap" is the best term I could devise to describe the following notion. The frequencies shown in the frequency plot will be limited to

center frequency +/- center frequency * Channel Overlap.

Thus, for a center frequency of 400 Hz., a Channel Overlap of 1.0 will produce a frequency plot limited to 0:800 Hz. In effect, reducing the Channel Overlap "zooms in" on the frequency plot. If you want to see more frequency detail, make this number smaller. If you make it too small, you may get clipping of the frequency plot, sometimes even in the "steady-state."

LORE: In general, the higher the harmonic number, the smaller the Channel Overlap necessary.

LORE: Channel Overlap is limited to the range 0--1. If you type something outside this range, EMERGE will supply the value of 1.0 (and tell you that it did so).

LORE: This was inspired by PRCFRQ in John Grey's SYNTH.

Squelching frequency functions

This section deals with the following lines from the parameter menu:

Squelch frequencies? (T or F)	FALSE
Squelch frequencies to this frequency:	121.727
When the amplitude is this many dB down:	50

When the amplitude of a harmonic is quite low, the phase vocoder can produce a widely varying frequency trace which is essentially noise. If you set the boolean in the first of the three lines above to TRUE, the frequency trace will be "squelched," i.e. set to the frequency in the second of the three lines above. This squelching happens when the AMPLITUDE of the harmonic falls below a certain threshold, specified in the third line above. Depending on the dB setting, the effect can be to remove the noisy parts of the frequency trace, which nicely clears up the resulting plot.

LORE: In squelching frequencies, only the original .MF frequency information is squelched. Any line-segment function you also show will not be squelched in frequency.

Including a label

This section deals with the following two lines from the parameter menu:

Label	CA3TA.JMF [CLR,AWN]
Include label (T or F)	TRUE

If "Include label" is set to TRUE, a label of the following form will be displayed at the bottom of the screen along with the amplitude and frequency plots:

Channel 1 Ref. Amp. = 9.227 (linear) Center Freq. = 337.04 Feb. 20 1983 16:17

If "include label" is TRUE AND you have something in the parameter menu's "label" entry, then that string will appear on the line above the EMERGE-supplied label.

If you are showing the amplitude function, you will be reminded that this is a "(linear)" or "(db)" plot. The Reference Amplitude is the one specified in the parameter menu in the line

"Max (reference) amplitude".

If you are showing the frequency function, "Center Freq." is the frequency taken from the WD1 entry for that channel in the merge file directory. See "The information in the directory" and Appendix 1.

You may not see the label at the bottom of the screen, but it will be there. It will definitely show up when you use PLOT to make hard copy.

Making a .PLT file

At the bottom of the parameter menu for showing one channel is the line

Plot File Name (DSK:CLMD3.PLT).

When you type α G to this parameter menu, the terminal will show the desired display. If you have typed a file name in this entry, then a "plot" file (also generically known as a ".PLT file") will be created before you are returned to the parameter menu. This plot file name is initialized to have the same file name as the current input merge file, plus the extension "PLT", plus whatever device the current input merge file lives on (like "DSK:" or "UDP1:"). The entire display, including any label drawn for the plot, will be output to one the one file specified.

You can look at such a file again with the program SGRN, which is undocumented (only argument: ".PLT file name").

You can make a hard copy of such plots using the PLOTL program, which is nominally self-documenting (type "?" to the first prompt).

WIZARDS: these are Grinnel .PLT files, not JAMLIB .PLT files. READ GRNLIB for details, or HELP PLOT.

Specifying file names for plot files

In many of the parameter menus in EMERGE, a default output file name is provided, surrounded by parentheses (see Figure 6). For these output files, an initial "(" means that the output file will not be written. To cause an output file to be written, simply remove the initial "(".

LORE: If you are trying to finalize a plot, set up the output .PLT file name from the start. When you've finally gotten the plot the way you want it, you'll have another chance to plot the file, because you can type αP to the prompt at the bottom of the plot. You'll be prompted for the plot file name, using as a default whatever file name is in the menu.

LORE: The final ")" is cosmetic and can be omitted.

WIZARDS: For such entries, menu[menuType] is set to untyped rather than fileNameTyped. This turns off the file-name checking, which would blow up on the parenthesis. See MENU.HDR[SUB,SYS] for details.

Three-Dimensional Plot

Parameters for three-dimensional plot from MF file UDP1:CA3TA.JMF [CLR,AWN]

Draw amplitudes? (T or F)	TRUE
Draw frequencies? (T or F)	TRUE
Amplitude plot:	
Amp x0 (in pixels, 0-511)	0
Amp y0	48
Reference amplitude = maximum of each channel?	FALSE
Use dB for amp plots? (F = linear, T = dB)	FALSE
dB Range	50
Plot forced to lie between successive x axes?	FALSE
Frequency plot:	
Freq x0	256
Freq y0	48
Squelch frequencies? (T or F)	FALSE
When the amplitude is this many dB down:	50
Freq. Channel Overlap (0.0 to 1.0)	0.01
Freq. Display Range (0.0 to 1.0)	1.0
Parameters for both kinds of functions:	
First Channel	1
Last Channel (58)	5
Beg time	0.0
End time (2.09)	2.0968359
Show merge function?	TRUE
If so, draw it with lines? (F = dots)	TRUE
Also show Edited, Approximated, or "Other" functions? (Slot E, A, or O; N or F = none)	N
Include label (T or F)	TRUE
Label	
Window Width (in pixels)	232
Window Height	428
Pixels along Z-axis between channels	100
Clockwise angle (degrees) between X- and Z-axes (0-90)	60
Plot File Name	(UDP1:CA3TA.3D/Q)

Figure 7. Parameter menu for creating a 3D plot.

If you type αG to this menu, a reasonable three-dimensional plot will be produced.

Many of the parameters which control three-dimensional display have already been discussed in the section "Show One Channel." Note the following differences:

1. Channel No. has changed into a range of channels (First:Last). Note that the "Last Channel" is initialized to the highest channel in the merge file. It can take a long time to calculate all of those channels...

2. There are two choices for the reference amplitude for each channel. Either the maximum amplitude of each channel is used, OR the maximum amplitude of all of the channels within the specified range.

3. When squelching frequencies, EMERGE calculates some reasonable frequency to squelch to, according to the formula

$$(\text{current channel number}) * \text{sRate} / N.$$

4. If you set

Reference amplitude = maximum of each channel?

to FALSE, the maximum amplitude is set to the maximum of all of the channels in the entire file, not just to the maximum within the specified range.

LORE: This makes it easier for you to make plots of subsets of the channels, for example, Channels 1-5, then 6-10, and

so on, all relative to the same amplitude.

5. The prompt for "Also show ... functions" and the prompt for choosing between lines and dots will be explained in the section "Looking at functions" under "Dealing with .FUN files".

6. If you realize you've made a mistake while the plot's being prepared, then type <esc>I, and you'll be returned to the menu on a bed of feathers.

LORE: There are two other programs for making 3-d plots from .MF files: MOV3D and NEW3D. MOV3D (which you get by typing R R;MOV3D) allows you to move the 3-D plot around. NEW3D (R R;NEW3D) draws lines by connecting all the channels at a given time--- the plot given in EMERGE is made by connecting all the points for a given channel. This is the program I used to make the plot on the front cover of the signal processing book published by Kaufmann. There is also MAK23D, for viewing two overlapping .MF files in three-D; but that program was never released publicly. You can find it on P252 as MAK23D.DMP [ROM,AWN].

Designing a 3-D Plot: Angle, Z-axis Spacing

This section deals with the following two lines from the parameter menu:

Angle (degrees) between X- and Z-axes (0-90)	60
Pixels along Z-axis between channels	100

The three-dimensional plot is designed as shown in Figure 8.

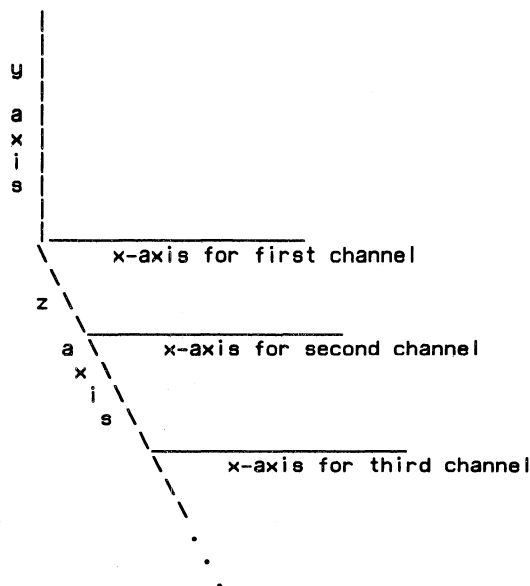


Figure 8. The x-, y-, and z-axes in a three-dimensional plot.

The lowest-numbered channel to be shown is displayed on an x-y axis pair at the back of the plot. Higher-numbered channels are displayed in front along other x-axes, for which y-axes are not drawn. The two lines from the parameter menu listed above control where the successive x-axes are placed.

The "Angle" is the angle in degrees which the z-axis makes with the x-axis for the first channel. This angle is measured on the plane formed by the y-axis and the x-axis for the first channel. This angle is limited to the range 0--90. Angle = 0 means that the z-axis overlaps the x-axis. Angle = 90 means that the y-axis "continues down into" the z-axis, so that the left-most points of all of the x-axes are vertically aligned.

LORE: The angle is measured counterclockwise from the x-axis for the first channel.

The quantity "Pixels along Z-axis between channels" controls the spacing of the successive x-axes along the z-axis. Setting this quantity to 0 means that all of the x-axes will be on one horizontal line. Setting this to some huge number will make the plot fill the entire window.

LORE: When you change the angle and/or the spacing along the z-axis, the x-axis will be correspondingly widened or narrowed so that as much as possible will fit into the window.

LORE: The three degenerate cases "angle" = 0 and/or angle = 90 and/or "pixels" = 0 do not blow up at you. They even produce interesting results. Setting angle to 90 and spacing to 0, for example, means that all of the functions

will be drawn on the same x-y scale, effectively overlapping all of the functions. However, angles OUTSIDE the range 0--90 (inclusive), and/or negative settings for the separation in pixels along the z-axis, will produce comic results (and will not blow up either).

LORE: 1. These are VIRTUAL pixels, not the PHYSICAL pixels mentioned in the section "Specifying Screen Coordinates." Every window has virtual dimensions (0,0) -- (511,511). EMERGE checks to make sure that this number is not too big --- which would result in some of the channels dropping off the edge of the window. In such cases, EMERGE (internally, not in the parameter menu) changes this number so that the number of channels specified will fill as much of the window as possible. You will be notified of this in the prompt area, and told what the temporary setting is. 2. Like the angle, the pixels are measured as though the z-axis were projected onto a plane formed by the x- and y-axes. The result is a measure of virtual pixels in this plane. Dust off your trig, it's good for you.

WIZARDS: The 3-D plot is placed inside a "window frame," which is required to be at least one PHYSICAL character width (6 PHYSICAL pixels) or one PHYSICAL line height (12 PHYSICAL pixels) from the window border. For the given window, these quantities are returned by the procedures hiXFrame, loXFrame, hiYFrame, loYFrame.

Freq. Display Range

This quantity is closely related to "Frequency Overlap," discussed in the section "Show One Channel." You will not understand this section unless you have grokked the discussion of Frequency Overlap. To review: Frequency Overlap determines the RANGE OF FREQUENCIES to which the frequency display for the current harmonic will be limited.

Display Range, on the other hand, determines the RELATIVE AMOUNT OF ROOM available between successive frequency x-axes for showing that range of frequencies.

If, for example, the frequency traces are bumping into each other in the three-dimensional display, then make Display Range smaller. This will "shrink" each frequency plot.

LORE: Another solution is to squelch the frequency traces some, using the squelch facility in the parameter menu for three-dimensional plots. This will take out those huge ugly excursions in .MF files where the harmonic amplitude is quite low, and make for a much clearer plot.

Spectrographic plot

Parameters for spectrographic plot from MF file UDP1:CA3TA.JMF [CLR,AWN]

For both merge and SEG functions:

First Channel	1
Last Channel (58)	6
Begin time	0.0
End time (2.0968359)	2.0968359
Use dB? (F = linear, T = dB)	TRUE
dB Range	50
Reference amplitude = maximum of each channel?	FALSE
Channel Overlap (0.0 to 1.0)	1.0
Squelch when bar height = 0?	FALSE
Show merge functions?	TRUE
If so: Skip this many points	8
Show vertical bars?	TRUE
Outline each harmonic?	FALSE
Show center line in each harmonic?	TRUE
Show Edited, Approximated, or "Other"	N
functions? (Slot E, A, or O; N or F = none)	FALSE
For SEG functions: Show vertical bars?	TRUE
Outline each harmonic?	FALSE
Show center line in each harmonic?	FALSE
Suppress beginning and ending 0-valued points?	FALSE
Include label (T or F)?	TRUE
Label	UDP1:CA3TA.JMF [CLR,AWN]
Plot File Name	(CA3TA.SPE/Q)

Figure 9. The parameter menu for making spectrographic plots.

Many of these parameters have been discussed in the preceding two sections.

What does a spectrographic plot show

Each vertical bar represents the instantaneous amplitude of a harmonic; the taller the bar, the greater the harmonic's amplitude. The height of each bar is calculated in decibels relative to the maximum amplitude of all of the channels being displayed. The vertical bars tracing out the lowest-numbered harmonic are strung out horizontally at the bottom of the plot, with higher-numbered harmonics above that.

The vertical center position of each bar shows the current frequency of the harmonic, in Hertz. The frequency scale on the y-axis refers to these center frequencies.

Parameters for spectrographic plots

1. Skip this many points. If you set this to 1, then every point from the .MF file will be shown. Usually, you won't want this. It will take a lot of time to compute, and will just show up as a white blur on the screen anyway. Make this number nice and big, like 20 or 50, at least until you know what the other parameters settings will be. For a final plot to go into hard copy, maybe make this number smaller again.
2. Outline each harmonic? The horizontal bar representing the harmonic may be outlined.
3. Show center line in each harmonic? If TRUE, we draw for you a line representing the center frequency. This was suggested by JC.
4. The prompts governing .FUN files will be covered in a later section.

Dealing with .FUN files

A function file which EMERGE can recognize, generically known as a ".FUN file," is a text file containing one or more SEG or SYNTH functions. Here are samples of an amplitude and frequency function, respectively:

```
SEG (A1 );
0.0087 0.0000
0.2212 0.1925
3.0719 0.2183
4.2386 0.2458
4.3338 0.5603
3.6786 0.6153
1.0906 0.6548
0.1431 0.7494
0.0000 0.9969
```

```
SEG (F1 );
121.7279 0.0000
146.4153 0.2252
154.0527 0.7477
121.7279 0.9952
121.7279 0.9969
```

"SEG" gives the class of function (the only class recognized by EMERGE). "A1" and "F1" are the function names. The numbers in the right-hand columns are times, in seconds. The numbers in the left-hand columns give amplitude on an arbitrary linear scale (not guaranteed to be normalized to 1.0) for amplitude functions, and frequencies in Hertz for frequency functions.

LORE: Note the differences between merge and .FUN files: merge files are binary, they have a directory through which the functions are accessed, and time is implicit (calculated from the sampling rate and compression ratio stored in the directory). .FUN files are text files, each function value is explicitly assigned a time, and each function is directly preceded by its own name.

WIZARDS: this is not enough to define a .FUN file for all uses. See SEGSYN.SAI [SYS,MUS], the SEGSYN incorporated into EMERGE, and/or MunchSegFile in SAMBOX.SAI [SYS,MUS].

CAVEATs and incompatibilities with CCRMA .FUN Files

1. EMERGE ignores any SYNTH functions (and any class of function except SEG). Your .FUN files read in by EMERGE may contain SYNTH functions with impunity.
2. Normal CCRMA function files have time values normalized to a scale of 0--100. Many function files also have frequency/amplitude values normalized to a scale of 0--1. EMERGE will read such files without blowing up, but will not deal with them in any desired manner, possibly excepting the comic. Other CCRMA software (such as FUN or SAMBOX) is not guaranteed to work with EMERGE-produced .FUN files.
3. EMERGE "knows" only about functions with names of the form y.Ax and y.Fx, where "x" is a channel number and "A" or "F" stands for amplitude or frequency, respectively. "y." is an OPTIONAL function name ignored by EMERGE when reading in .FUN files, and optionally supplied by EMERGE when writing them. The "y" may be of ANY length (as far as EMERGE is concerned) and may contain any alphanumeric character except ".". "." is the only recognized delimiter, and is always supplied by EMERGE when needed for writing out .FUN files.

Write out .FUN file

EMERGE may be used to convert from a merge file to a .FUN file, using the menu shown in Figure 10.

```

Parameters for creating .FUN file from
MF file UDP1:CA3TA.JMF [CLR,AWN]

Output Edited, Approximated, "Other", or Merge
file functions (Slot E, A, O, or M)      E
Output File Name                          UDP1:CA3TA.FUN/Q
First Channel                            1
Last Channel (58)                         3
Begin time                               0.0
End time (2.0968359)                     ∞
(∞ = stop at final point in function)
Shift functions by this time              0.0
Use this string as function "name":       CA3TA
Output amplitude functions?               TRUE
Output frequency functions?               TRUE
Output any functions ignored when reading
into this slot?                           TRUE

Outputting merge functions:
Skip this many points:                    0

```

Figure 10. The parameter menu for writing out .FUN files.

The first line is explained in the section "Function slots inside EMERGE," q. v. For the purpose of this section, it is sufficient to state that setting this parameter to "M" causes EMERGE to output a .FUN file directly from the current input file.

The amplitude function and the frequency function (whichever you specify, or both) for the indicated slot within the indicated channel range are output to the specified file when you type αG. If the amplitude or frequency function happens to be missing, EMERGE will continue outputting the rest of the specified functions without telling you. If the specified slot is completely empty of functions, you will be returned to this menu without any file being written.

For the "ignored functions", see the section of the same name (below).

The specified begin times and end times are inclusive ranges. If the function does not include a point at the specified time, then an extra point will be added. Its corresponding value will be the value of the point directly before (after) it. If you specify "∞" as the end time, then this is turned off; and whatever is the end point of the function will be written out to the file. For writing out the merge file directly, "∞" specified as the end time will get you the entire length of the merge file.

You can specify some arbitrary time shift. For example, if you specify a begin time of .283, then you might specify a shift of -.283, so that the function will start at time 0.0 sec. When you read in a .FUN file to EMERGE, you can specify the same time offset to restore the times to their original values.

If you specify a "function name" (see the section "CAVEATS and incompatibilities"), EMERGE will tack that name on to the front of the standard "A1", "A2", ..., "F1", "F2" function names. EMERGE will supply the "." delimiter for you.

When you are outputting merge functions (slot set to "M"), you can cause every Nth point to be skipped by setting N in the last line of the menu. N=0 means that the .FUN file contains all of the points in the original merge function; N=1 produces every other point, and so on.

WARNING: text files created directly from merge files can be quite large (100 K or so for 20 harmonics of a 1-sec. note!).

LORE: The entry for Output File Name is initialized to the same device and file name as the current input merge file, with the extension .FUN. If you type some other file name, sorry, you don't get the ".FUN" extension as a default, so you'd better type it in if you want it (or leave it there as you're editing the line). The section "Specifying file names for plot files" discussed the possibility of beginning an output file name with "(". That is not possible in this menu.

LORE: The optional "name" is initialized to the name specified using the menu for setting up defaults for the current input merge file. If you supply some other name here, it will not overwrite the default name supplied in that other menu.

WIZARDS: Actually, the function names are "normalized" to A1, A2, ..., F1, F2, ... when a .FUN file is read in, stripping off any "name." constructs in those function names right away. The "name" specified in this menu is added by the code which writes out to the file (wrtFuncFile), leaving the internal representation of the function (including the function name inside EMERGE) alone.

Read in .FUN file

The parameter menu for this is quite simple:

Parameters for reading in .FUN file associated
with MF file UDP1:CA3TA.JMF[CLR,AWN]

Input File Name	UDP1:CA3TA.FUN
Read these into slot E, A, or O	E
Add this time to all function times	0.0
If there's already a function in the slot:	
Replace it with the function from the file?	TRUE
In that case, clear out the whole slot first?	FALSE
Meld both functions?	FALSE
Function from slot overwrites function from file?	FALSE
Function from file overwrites function from slot?	FALSE

Kludge: to clear a slot completely, leave "Input File Name"
blank. The TRUE/FALSE settings in the bottom four
lines will be ignored.

Figure 11. Reading in a .FUN file.

The entry for Input File Name is initialized to the same device and file name as the current input merge file, with the extension .FUN. If you type some other file name, the extension ".FUN" will be supplied as a default.

The third entry in this parameter menu can be used to offset times in functions written with EMERGE. See the section "Write out .FUN file" for the intended use.

Mixing functions from the file with those inside EMERGE

This section concerns the lines:

If there's already a function in the slot:
Replace it with the function from the file?
In that case, clear out the whole slot first?
Meld both functions?
Function from slot overwrites function from file?
Function from file overwrites function from slot?

from Figure 11.

Any function already inside EMERGE will be left alone unless the .FUN file contains something to write over it with. In this case, you can

1. overwrite the entire function inside EMERGE ("replace" = TRUE),
 2. or the two functions will be interleaved ("meld" = TRUE),
 3. or you can splice one into the other (one overwrite option = TRUE).
- Only one of these 3 may be "TRUE" at any one time. Yes, EMERGE will check this.

In case you explicitly want to replace the functions inside EMERGE with those from the file, you have the option of clearing the entire "slot" first (see below).

For melding and overwriting:

Consider the functions

A = {(0,0), (1,.25), (1,.75), (0,1.0)} and

B = {(0,.1), (.8,.15), (.8,.85), (0,0.9)}.

"Meld" means to rigorously retain all points in both functions, but to interleave them in time, comme ca:

(0,0), (0,.1), (.8,.15), (1,.75), (.8,.85), (0,0.9), (0,1.0)

Overwrite means that the overwriting function is "dropped into" the overwritten function; any points inside the overwritten function disappear, such as when B overwrites A:

(0,0), (0,.1), (.8,.15), (.8,.85), (0,0.9), (0,1.0).

This information, of course, is available under α ? for this menu.

Ignored functions

See "Caveats and incompatibilities" for restrictions on the names and types of functions that EMERGE will recognize. If EMERGE "gets confused" completely about what's in the file, it stops trying to read the file in and returns to this menu (with an appropriate complaint in the prompt area).

If EMERGE finds a function not of type "SEG", or finds a SEG function with an unrecognizable name, then it "ignores" that function --- but keeps it lying around. When you write out functions to a .FUN file using the appropriate menu, you can also write out such functions.

When EMERGE is done reading in the .FUN file, it prints out a list of the functions in the current "slot" (see below). Type αE when you're done looking at the list. EMERGE will also complain about any ignored functions as they're being read, but they remain invisible forever thereafter as far as EMERGE is concerned.

Function slots inside EMERGE

For each merge file input into EMERGE, there are three "slots" named "E", "A", and "O" inside EMERGE for storing .FUN-style functions related to that merge file. Each slot has the following format:

Frequency	Amplitude
F1	A1
F2	A2
F3	A3

⋮

Each slot has enough room to contain as many amplitude and frequency functions as the merge file itself contains. Not all of these positions in the slot need be filled.

Using function slots

Using the menu for reading in .FUN files, you can read in functions into any of the three slots A, E, or O. EMERGE will overwrite anything already in those slots according to the settings in the menu.

The O ("other") slot is a miscellaneous slot for your personal convenience; no extra charge. Whenever you create an approximation using the Pavlidis split/merge algorithm (see the thusly named section in this documentation), the resulting function is stored in the slot called "A." Whenever you edit a function using EMERGE's editor, the resulting function is stored in the "E" slot. Again, EMERGE will overwrite anything already in those slots.

Using the menu for writing out .FUN files, you can write out functions from any of the three slots A, E, or O.

LORE: In the menu just mentioned, "M" is not really a slot inside EMERGE, but gratuitously serves as another source designation.

LORE: There are thus various ways to work. For example, you can create some functions using the Pavlidis split/merge algorithm, and write out that set into a .FUN file. Later, you can read back that set into the "A" slot, and continue to create functions (or make some modifications using the split/merge algorithm). Another possibility is to read such a set of functions into the E slot, but to fine-tune them using EMERGE's editor. When you write out the entire E slot, any changes you made will be included, and you also get for free all of the functions that you didn't need to change.

Looking at SEG functions

This section concerns the following lines from the menu for showing one channel (and similar lines from the menu for creating 3-D plots):

Also show Edited, Approximated, or "Other"
functions? (Slot E, A, or O; N or F = none) N
Show merge function in Lines? (F = dots) TRUE

If there is a function in the specified slot for the specified channel, then EMERGE will show it along with the original merge function. (In the menu for creating 3-D plots, you can turn off the display of the functions from the merge file altogether).

In order to make the distinction between the two functions clearer, you can have the original merge function outlined with dots rather than dots connected with lines. This can be especially useful for frequency functions, or amplitude functions with a lot of squiggle.

LORE: The O slot is especially useful for comparing functions. You can have one set stored in the O slot, with perhaps an edited version of the same set already in the E slot. Using this menu, you can easily compare the edited functions with the original functions in the O slot.

21 Jan 1985 20:05

EMERGE.AWN[UP,DOC]

PAGE 44-1

Operating on data in current merge file:

Pavlidis' Split/Merge Approximations

The purpose of this EMERGE option is to allow you to more or less quickly and painlessly derive a line-segment approximation for the amplitude and frequency functions in the .MF file.

The following is taken from Strawn 1981. A copy of that article is included as an appendix to hard copy versions of this document. If you want to work extensively with this algorithm, you should plan on reading the following parts of that article:

p. 3; Second full paragraph under section 2 (p. 5 --- starting with "Furthermore, one important ..."); all of section 3 (p. 6 --- the "=" in the denominator of Eq. 6 should be "-"); Section 4.1; 4.2; 4.2.2 (first paragraph); Section 4.3 (first paragraph, p. 12); Figure 8 (p. 13).

The Menu

Figure 12 shows the menu for controlling this EMERGE option:

Parameters for appying split/merge to one channel from
MF file UDP1:CA3TA.JMF[CLR,AWN]

```

Channel No. (1:58)                1
Approximate amplitude function?    TRUE
  (FALSE = use frequency function)
Error Norm                        I
  (I = integral, X = maximum, M = mean)
Threshold                        10000
Beg time                         0.0
End time (2.09)                   2.0968359
Output File Name                   (UDP1:CA3TA.SMF/Q)
Debug File Name                    (DSK:CA3TA.TMP/Q)
Label                             CA3TA.JMF[CLR,AWN]
Include label (T or F)            TRUE
Plot File Name                     (UDP1:CA3TA.C/Q1)
Show merge function in Lines? (F = dots)  TRUE

For amplitude plots:
  Use dB for amp plot? (F = linear, T = dB)  FALSE
  dB Range                                  50

For frequency functions:
  Freq. Channel Overlap (0.0 to 1.0)        1.0
  Squelch frequencies before approximating?  TRUE
  Squelch frequencies to this frequency:    242.7401100
  When the amplitude is this many dB down:  50

```

Figure 12. The parameter menu for invoking Pavlidis' split/merge approximation algorithm.

Much of this menu will be familiar if you have already learned to use the menu for showing one channel at a time; documentation for that EMERGE option explains the following items:

```

Channel No.
Beg time
End time
Label
Include label (T or F)
  [The label here will remind you of the threshold, and
   gives a count of the number of approximating segments.]
Plot File Name
For amplitude plots:
  Use dB for amp plot? (F = linear, T = dB)
  dB Range
    [Note: these two setting affect only how the amplitude plot
     is DISPLAYED. The approximation algorithm still operates on the
     original linear form of the function.]
For frequency functions:
  Freq. Channel Overlap (0.0 to 1.0)
    [Again, this affects only the plot of the frequency function.]
  Squelch frequencies before approximating?
  Squelch frequencies to this frequency:
  When the amplitude is this many dB down:
    [Note: the squelching is performed BEFORE the function is
     shipped off to be approximated. I recommend that you do this,
     for reasons explained in LORE, below.]

```

The entry

Show merge function in Lines? (F = dots)

is explained in the section "Looking at functions" under "Dealing with .FUN files".

The Pavlidis Split/Merge approximation algorithm can take quite a while to run, perhaps a few CPU minutes for a one-second function. When you type αG to this menu, the system will create the approximation and then display it on the terminal screen.

You can type <escI> to interrupt the approximation; you will be returned to the menu.

LORE: As explained in "Squelching frequency functions", squelching has the effect of removing the noisy part of the frequency trace. I recommend that you invoke the squelching facility when approximating frequency functions. If you don't, the approximation may pick off one of the peaks in the "noisy" part of the frequency trace as a main point in the approximation. This can actually result in inharmonicity in the resynthesized signal: if the approximated slope of the frequency function for one harmonic goes up, and that for another function goes down, this can lead to undesired results.

21 Jan 1985 20:05

EMERGE.AWN[UP,DOC]

PAGE 47-1

Approximate amplitude function?
(FALSE = use frequency function)

Setting the above line in the menu means that the amplitude function of the channel specified elsewhere in the menu will be approximated. Set this to FALSE to work on the frequency function.

Error Norm

(I = integral, X = maximum, M = mean)

These three option are explained under "Pavlidis' Split/Merge Approximations."
See also "LORE" under "Threshold".

In general, the Maximum norm is most sensitive to local large variations. The mean norm is less sensitive than the other two to a lot of small local variations (like dither).

Threshold

This menu entry is explained under "Pavlidis' Split/Merge Approximations."

LORE: You will have to play around with various settings for Error Norm and Threshold. Figure 8 in the CMJ article (listed under "Pavlidis' Split/Merge Approximations") gives three examples. For amplitude functions using the Integral norm, try setting the threshold to 25-30 dB above the channel's maximum amplitude. For frequency functions, try 5 or so times the harmonic's center frequency.

LORE: Having a copy of the merge file directory will facilitate your work with the Pavlidis algorithm. See "Looking at the information in the merge directory."

Debug File Name

If you want a record of what how the Pavlidis algorithm converges to its solution, then provide a file name here. See "Specifying file names for plot files" for details on how to specify a file name here.

Storing the approximations in the A slot

The idea of "slots" of functions inside EMERGE is explained in "Function slots inside EMERGE" and "Using function slots," q. v.

Whenever you have approximated a function, the new approximation is stored in the A slot automatically, overwriting anything that was already there.

LORE: We force the stored form to have a breakpoint at the very beginning and end of the original merge file function, EVEN IF you specify some other begin and end time for creating the approximation. If you actually want to output the approximation using the shorter duration, use the menu known as "Write out .FUN file," explained elsewhere in this document.

Output File Name

The idea of "slots" of functions inside EMERGE is explained in "Function slots inside EMERGE" and "Using function slots," q. v.

See "Specifying file names for plot files" for details on how to specify a file name here. If you supply a file name, then the entire set of functions in the "A" slot will be output to the file when the new approximation is done.

LORE: The complete range (in time) for all of these functions will be written out, no matter what begin or end time is supplied elsewhere in the menu. If you actually want to output the approximation using the shorter duration, use the menu known as "Write out .FUN file," explained elsewhere in this document.

LORE: I recommend that you always supply a file name here. By doing so, you will always have a record of the most recent state of your work.

LORE: I also find it convenient to use the menu package's αW option. This writes out a copy of a menu to a text file on the screen. I write this out before typing αG to the menu. If for some reason something bombs, you can read in exactly the same text file and pick up right where you were.

Indeed, I wrote a SAIL program called MAKMEN (archived on [ROM,AWN]) which automatically created

1. a menu with main settings, like output file name and choice of error threshold.
2. a set of menus, one for each harmonic, with an appropriate threshold value.
3. a .DO file that runs EMERGE and steps through all of the harmonics. This is the kind of thing that runs well late at night.

The EMERGE function editor

The menu for editing functions is shown in Figure 13.

Parameters for editing one channel from MF file UDP1:CA3TA.JMF [CLR,AWN]

Edit Amp? (T = amp, F = freq)	TRUE
Amplitude plot:	
Use dB for amp plot? (F = linear, T = dB)	FALSE
dB Range	60
Reference amplitude = maximum of channel?	TRUE
If not, use this reference amplitude (=7.5116417)	7.512
Frequency plot:	
Squelch frequencies? (T or F)	FALSE
Squelch frequencies to this frequency:	242.7401100
When the amplitude is this many dB down:	50
Freq. Channel Overlap (0.0 to 1.0)	1.0
Parameters for both kinds of functions:	
Channel No (1:58)	1
Beg time	0.0
End time (2.09)	2.0979688
Edit previously Edited, Approximated, or "Other" functions? (Slot E, A, or Q; N or F = none)	N
Edited function inserted into function in E slot? (FALSE = old function completely overwritten)	TRUE
Show merge function?	TRUE
Include label (T or F)	TRUE
Label	CA3TA.JMF [CLR,AWN]
.FUN Output File Name	(UDP1:CA3TA.FUN/Q)

Figure 13. The menu for the EMERGE function editor.

This menu is quite similar to the one shown in Figure 6 in the section "Show One Channel". See documentation for that menu for further information. The rest of this section assumes that you are familiar with that menu.

Note that dB (for plotting amplitude, or for squelching frequency) are always calculated relative to the maximum amplitude of the current channel.

When you type αG to this menu, the display will show a plot almost identical to the plot for showing one function (see "Show One Channel"). In this case, however, the EMERGE function editor (explained below) will be invoked. You may edit only the amplitude or the frequency function, not both together.

21 Jan 1985 20:05

EMERGE.AWN[UP,DOC]

PAGE 54-1

Edit Amp? (T = amp, F = freq)

Set this to true or false, depending on which function for the current channel (set by "Channel No." in the menu) you want to edit.

Edit previously Edited, Approximated, or Other functions?

The idea of the EMERGE function editor is to prepare some line-segment approximation to the function in the merge file. If you specify "N" or "F" to this entry in the menu, then the EMERGE function editor will be initialized to a blank state.

You can choose to edit a function in the other "slots" inside EMERGE. (The idea of "slots" of functions inside EMERGE is explained in "Function slots inside EMERGE" and "Using function slots," q. v.) IF EMERGE can find a function in the specified slot for the specified channel and side (amplitude/frequency), then that function will be loaded into the EMERGE function editor. If you specify an invalid slot, or no function can be found in the offending slot, then your request will be ignored, and the editor will be initialized to be empty, as explained above.

See also "Storing the edited functions in the E slot", below.

Using the EMERGE function editor

Using commands from the terminal keyboard, you move a "cursor" (the character "*") around on the terminal screen using the commands listed below. You can also deposit a point anywhere inside the space formed by the x- and y-axes. To return to the menu, type αE to the EMERGE function editor.

Cursor moving commands: 7 8 9 U O J K L , M

Surrounding the letter I on the QWERTY keyboard is a ring of keys:

7 8 9
U I O
J K L

These keys move the cursor 1 pixel from its current position in the following directions:

7	8	9	diagonally up to left	↑	diagonally up to right
U		O	←		→
J	K	L	diagonally down to left	↓	diagonally down to right

To move the cursor more than one pixel, use these "control" bits:

α	multiply distance times 2
β	times 4
m	times 16
,	times 256

Combine the α, β, ", " "m" keys to get intermediate multiples like 32 and 64.

LORE: This ring-of-keys idea was adopted from the old NS news service program that runs on SAIL. Attempts to use CCRMA's extra microswitch keys failed for reasons that will be clear to any seasoned CCRMA observer.

21 Jan 1985 20:05

EMERGE.AWN[UP,DOC]

PAGE 58-1

Moving along the function: $\leftarrow \rightarrow$

These two keys move the cursor along the APPROXIMATION to the next point to the left (to the right for pad6) of the current cursor location.

$\alpha\leftarrow$ and $\alpha\rightarrow$ move along the original function.

Inserting/deleting a new point: I, D

I inserts a point into the edited function at the current cursor location.

If the cursor lies exactly on top of a point in the edited function, D deletes it.

LORE: to get on top of the point, use → or ←, q. v.
If you are not on top of a point, EMERGE will complain.

Attaching a point: A

A "attaches" a new point at the current cursor location. An "attached" point is one that you can move around and look at, but isn't part of the edited function --- yet. The cursor for an attached point is shown as the letter "A".

When the editor is in the attached state, lines are drawn to the attached point to show what the approximation will look like when the attached point is released. You can move the attached point with the cursor-moving keys (7 8 9 U O J K L , M)

If a point is already attached, A releases it before grabbing a new one at the same place.

LORE: If the cursor is right on top of a point in the edited function when you type αA , then that point is deleted (from the edited function) at the same point time that the attached point is created; in other words, αA attaches that old point. If a point is already attached, αA simply make that attached point (and the attached state) go away, without inserting the attached point into the edited function.

Miscellaneous commands: α ., αE , $\alpha\beta E$, αP , αV , αW , $\alpha?$

Unless explicitly specified, α (or \bullet) are optional with the following E-style (and menu-style) commands:

- ? Help
- . If attached, insert a point at current cursor location. Then save the current state of the approximation. Useful with $\alpha X-C$ (see "Extended Commands").
- E Insert any attached point into the edited function, then return to the menu.
- $\alpha\beta E$ Abort and exit. Like $(\alpha)E$, except that nothing which you ever edited will be saved.
- P Print out, in the prompt area, the current location of the cursor.
- $\alpha\beta P$ Print out the values of the entire edited function on the terminal screen. Type "V" when you're done staring at the points.
- V Refresh screen (useful after <break>N)
- W Insert any attached point, then write a .FUN file of the current edited function (ONLY --- not the entire "E" slot). You will be prompted for the file name. See also "Storing the edited functions in the E slot".

Extended commands: α X

Actually, the " α " is optional with α X. After typing α X, you will be prompted for one of these commands:

A Toggle whether axes are to be drawn along with original/edited function. This affects only the side (amplitude or frequency) that you are editing.

C Cancel edits (or restore most recent "." state, if any) and redraw the original edited function (also turns off attached state).

D Delete all points between (but not including) marked points. See α X M.

F Toggle whether the original function is to be drawn along with edited function.

M Mark the current point in the approximation --- used in conjunction with α X D.

P Output a .PLT file of entire screen. You will be prompted for the

T Force a point to be inserted at user-specified time. You will be prompted for the time. NOTE: If you are currently attached to a point, that point will be dropped at the current value, taking on the time you specify.

V Force a point at a value. You will be prompted for the value. prompted for the time. NOTE: If you are currently attached to a point, that point will be dropped at the current time, taking on the value you specify. file name.

Z Zero out edited function to just beginning and ending point of original

. Toggle whether original function is drawn with lines or DOTS.

WIZARDS:

(\odot)X-W may toggle wizard status. This will print out copious amounts of debugging information on the fly. Also, the printout for (\odot)X-P is expanded.

α β X-D (bucky bits required) may get you debugging info probably useful only to JAWN.

EMERGE function editor LORE

When attached, you can move only as far to the left or right as the end points of the lines that connect the attached point to the approximation. Got it? When you leave the attached state, the line in the approximation connecting those two end points will disappear. Try it!

In general, work from left to right. If the points get connected in an order you don't like, you'll have to delete one of the points and re-insert it.

Let's say you read in 40 or 50 channels from a .FUN file, then you want to edit each of those functions. I've found that it's quicker to write out the edits every 5 or 10 channels, rather than writing out the new .FUN file every channel. Writing out all of those functions can take longer than it takes to edit 2 or 3 functions.

Storing the edited functions in the E slot

The idea of "slots" of functions inside EMERGE is explained in "Function slots inside EMERGE" and "Using function slots," q. v.

Whenever you have edited a function, the new edited function is stored in the E slot automatically. From the menu in Figure 13:

Edited function inserted into function in E slot?
(FALSE = old function completely overwritten)

With this option set to TRUE, you get the same effect as discussed under "Overwrite" in the section "Mixing functions from the file with those inside EMERGE".

LORE: We force the stored form to have a breakpoint at the very beginning and end of the original merge file function, EVEN IF you specify some other begin and end time for editing the function. If you actually want to output the edited function using the shorter duration, use the menu known as "Write out .FUN file," explained elsewhere in this document.

LORE: Even if you are editing with an amplitude function displayed in dB, the function will be written out in the usual linear form.

Output File Name (in the menu)

The idea of "slots" of functions inside EMERGE is explained in "Function slots inside EMERGE" and "Using function slots," q. v.

See "Specifying file names for plot files" for details on how to specify a file name here. If you supply a file name, then the entire set of functions in the "E" slot will be output to the file when the new edited function is done.

LORE: The complete range (in time) for all of these functions will be written out, no matter what begin or end time is supplied elsewhere in the menu. If you actually want to output the edited function using the shorter duration, use the menu known as "Write out .FUN file," explained elsewhere in this document.

21 Jan 1985 20:05

EMERGE.AWN[UP,DOC]

PAGE 66-1

News

When a new version of EMERGE is finished, the changes are logged in the News area by version number and date.

Acknowledgments

I would like to thank the following people for the following help and/or code:

James A. Moorer, for designing the original MRGPAK, from which I lifted some things (with modification)

John Gordon and Bill Schottstaedt for code granting access to the keys on the microswitch keyboard; even though it never worked right, and it wasn't their fault

Bill Schottstaedt and David Jaffe for help with SAMBOX

Bill Schottstaedt for his window system and GRNLIB, and much much much help

Rick Taube and Xavier Serra for suggestions, patience and help during final debugging

References

Gordon, J. W., and J. Strawn. "A Musician's Guide to the Phase Vocoder." In J. Strawn, ed. Digital Audio Signal Processing: An Anthology. Los Altos, Ca.: Kaufmann, 1985.

Grey, John M. "An Exploration of Musical Timbre." Ph. D. Dissertation, Dept. of Psychology, Stanford University. Department of Music Report STAN-M-2, 1975.

Strawn, John M. 1981. "Approximation and Syntactic Analysis of Amplitude and Frequency Functions for Digital Sound Synthesis." Computer Music Journal 4(3):3-24.

Appendix 1: A Plaine and Easie Introduction to Merge Files

(see also the section "Merge Files")

The functions contained inside a merge file are named according to the following convention:

<instrument name>.<extension>

where both the name and the extension may contain up to six characters. For .MF files, the extension will consist of the letter A (for amplitude function) or F (for frequency function) followed by a number representing the harmonic number (aka channel number).

<instrument name> is up to the user of the file. Often, the instrument name is identical to the file name (minus extension). Typical instrument names from John Grey's thesis would be OB59, CL03, etc., and would be contained in files called OB59.MRG, CL03.MRG, etc. "Instrument name" would thus be derived from the instrument originally recorded.

At the beginning of a merge file there is a directory, which contains for each function in the file the following information:

```

      instrument NAME
      EXTENSION of function

      WD#                      REC#
      #WDS                     LAST_REC

      Sampling RATE (SRATE)

      COMPReSSion ratio

      VAL

      QVAL
  
```

NAME and EXTENSION were discussed above. VAL and QVAL are two words which may be set by the program(mer) to be any value needed. Continuing the example of functions for .MF files:

amplitude functions	VAL peak amplitude for current function	QVAL peak amplitude for all amplitude functions in file or: peak amplitude of harmonic #1
frequency functions	mean frequency for current function	fundamental frequency (i.e. mean frequency of harmonic #1)

VAL and QVAL are sometimes referred to as WD1 and WD2, respectively.

WD#, REC#, #WDS, and LAST_REC are explained on the next page, which should be skipped by beginners. The other quantities are explained in the section "Merge files".

Merge File Directories (for WIZARDS)

For each function in the merge file, 8 successive 36-bit words are filled with directory information. The directories live at the beginning of the file; the first word of the first directory is in word 1 (use USETI) of the file. The directory ends with one or more words containing 0.

Using FRAID, it is possible to examine and modify the contents of the file, including the directory.

The rest of this page is a guide to the individual words (and storage formats) in the directory.

Word

- 0 contains the "instrument name" in 6-bit ASCII format (use 6xT in FRAID)
- 1 contains the "extension" in 6-bit ASCII. Notice that the "." printed out by some programs (including EMERGE) is not stored here.
- 2 right (!) half: REC# (octal) record number in file where function starts. Records in a file are numbered starting with 1.
left half: WD# (octal) displacement from begin of record where function starts. 0 means the first word in the record.
- 3 left half: #WDS (octal) number of words (points) in function.
right half: LAST_REC (octal) record number in file where function ends.

There are '200 words in each record. The address (octal) of the first word (point) in a function is given by

$$\text{start} \leftarrow (\text{REC\#} - 1) * '200 + \text{WD\#}$$

where the right single quote denotes an octal number. The last datum in the function occurs at the address given by

$$\text{end} \leftarrow \text{start} + \text{\#WDS} - 1.$$

LAST_REC is redundant, because LAST_REC of the current function is the same as REC# of the next function.

- 4 SRATE (decimal integer)
 - 5 compression ratio (decimal integer)
 - 6 WD1
 - 7 WD2
- note that some programs expect WD1 and WD2 to be integer, while others write it out as a real (use αF in FRAID).

Representation of merge file directories by the merge file software

This entire page is for WIZARDS only.

Each variety of merge file is represented in SAIL by a record of the appropriate class. Each class contains a field for an INTEGER ARRAY mrgDir. For each type of merge file, this page lists the entries in that directory. Nominally, each merge file function is assigned to mrgDir[chanNo...]. Certain miscellaneous functions such as .RMS for .K files are assigned certain offsets from the highest-numbered channel in the file. .P files have their functions assigned to dummy channel numbers. All of these channel numbers are available as macros or typed subroutines.

For any type of merge file, the directory for the comment function COMMEN.TXT is contained in endChan+1.

In the following, "dirEntry" refers to the macros dirNameEntry ... dirParEntry, defined in MERGE.HDR.

.MF files

The directory is mfClass:mrgDir[rp][chanNo,dirType,dirEntry], where dirType can take on the values dirAmpType and dirFreqType, defined in MERGE.HDR.

COMMEN.TXT is in [endChan+1,dirAmpType].

The SNDMF family of programs uses the dummy function N.TXT. Only the function entry in the directory contains any information. The word in the file pointed at by the function directory is unused. The fields of this dummy function contain the following information:

wd1	N	
wd2	Q	intermediate compression ratio

RESERVED: that 3 words pointed to by the directory contain beta, nQs, and nGroups from SNDMF

.K files

The directory is KClass:mrgDir[rp][chanNo,dirEntry]. "chanNo" refers in this case to the coefficients for the filter of that order.

Additional functions:

<function name>.RMS	endChan+2
<function name>.ERR	endChan+2
<function name>.GF	endChan+3

.P files

The functions are assigned the following arbitrary channel numbers:

<function name> .P	1
<function name> .C	2
<function name> .E	3
<function name> .C1	4
<function name> .ZCR	5
<function name> .PH	6
<function name> .NH	7
COMMENT.TXT	8

Appendix 2: Overview of files

All of this source code lives on UDP2:[LIB,AWN]. Older versions were on UDP2:[ROM,AWN].

EMERGE.SAI is the main program

EMERGE.HDR contains all of the EXTERNAL declarations for the entire system, including record!class descriptors.

UMERGE.SAI contains a large number of miscellaneous utility routines, including INTERNAL declarations for many RECORD!CLASSES.

The rest are listed alphabetically by file name:

APPR.SAI	is the interface to Pavlidis line segment approximation software. See appr.sai for where the Pavlidis code lives.
CURED.SAI	the EMERGE cursor editor
EDSEG.SAI	is the EMERGE interface to cured.sai
EDPT.SAI	maintains the linked list of points in the EMERGE cursor editor
EFL.SAI	is the file list editor
FILEDE.SAI	edits EMERGE default settings for current input merge file
GETFUN.SAI	read in a .FUN file
MAKDIR.SAI	shows the directory of the current input merge file
MAKE3D.SAI	creates 3D plots
MERGE.SAI	my extensions to JAM's mrgpak. These are based on my library FILEIO.REL[F,AWN] (see "Compiling EMERGE")
NEWS.SAI	shows EMERGE news
OUTFUN.SAI	is the EMERGE interface to seg.sai
SEG.SAI	contains a large number of utilities for dealing with .FUN files
SEGSYN.SAI	reads in .FUN files; see also getfun.sai
SHOWON.SAI	shows one channel of the current input merge file
SLICE.SAI	create spectral slice
SPECT.SAI	makes a spectrographic plot
WIN.SAI	is the code which displays merge and .FUN functions
WORKED.SAI	is the EMERGE function editor

The following are also needed:

SAIL.AWN	SAIL macros
AWNLIB.REL	a library containing miscellaneous junk, file i/o, and the line segment approximation software
MENU.REL	the menu package, obviously

The following files are not used in the standard system version of EMERGE:

CURED2	for MFED2, which edits two .MF files at once. This is a version of the EMERGE cursor editor.
--------	--

EDSEG2 for MFED2, which edits two .MF files at once. This is the
 interface to CURED2.

GETIF prompt for an input .MF files

MAK23D make a 3-D plot from two overlapping .MF files

MFDIF take difference between .FUN file and .MF file; used
 for program MFDIF

Appendix 3: Compiling EMERGE

EMERGE.DO[LIB,awn] contains all the code necessary for compiling EMERGE.

In addition, you will need the following source files and/or packages (if you leave them on an area other than specified here, you will have to change EMERGE.SAI).

SAIL.AWN[LIB,AWN] --- SAIL macros

UDP2:AWNLIB[LIB,AWN] --- SAIL and FAIL library
sources on UDP2:[LIB,AWN]
compiled with AWNLIB.DO[LIB,AWN]
header in AWNLIB.HDR[SUB,SYS]

UDP2:MENUL[LIB,AWN] --- menu package for selection and parameter menus
compiled from UDP2:MENU.SAI[LIB,AWN]
header in MENU.HDR[SUB,SYS]

WIZARDS: The rest of this page is for you.

1. EMERGE.DO actually uses two files for conditional compilation to speed up the compilation process.

For conditional compilation of EMERGE.SAI, there is a file DECL.SAI, which contains the REQUIRE declarations necessary for invoking all of the header files. When compiled, this yields a version of SAIL known as JAIL (Jawn's SAIL).

There is also the file called EDECL.SAI, which contains the REQUIRE declarations necessary for compiling all of the .SAI files EXCEPT EMERGE.SAI. It also contains an ENTRY statement, hence the "E" in EDECL. EMERGE.DO calls the resulting version of SAIL EJAIL.

2. There is a large number of compile-time switches in EMERGE.SAI, all beginning with "do...". You can set these to FALSE to make a version of EMERGE which contains only a subset of its options. The compilation is set up with a few tricky macros so that the main menu in such a version is shrunk accordingly (!). This is useful for debugging one of the features, as a considerable amount of load time is thus saved.

3. ACHTUNG: MERGE.HDR contains the following:

```
DEFINE myDev = <"PIT:">;
```

This corresponds to a line in EMERGE.DO that should look like:

```
a udpl pit*
```

Here's the trick: The two versions of SAIL (JAIL and EJAIL) expect to be working with a device named PIT:. The REQUIRE statements throughout the code expect a device DEFINED in the SAIL code as "myDev" with the value "PIT:". EMERGE.DO simply dumps all the .REL files onto PIT (hence the name: bottomless pit), and then loads EMERGE from that device as well. If you use JAIL and EJAIL for your own compilation, you will have to 1) assign some device (UDP or DSK) as PIT, and make it the same device as in EMERGE.DO; 2) make sure that you write any hand-compiled versions of .REL files onto PIT, as in

```
.RU PIT:EJAIL
*pit:showon+disk:showon
```

LORE: If when running JAIL or EJAIL the following happens:

```
.ru jail
*EMERGE+EMERGE
EMERGE.SAI 2
OPEN ERROR: TMQ
EMERGE, Page 2
```

21 Jan 1985 20:05

EMERGE.AWN[UP,DOC]

PAGE 73-2

00001

?↑C
↑C

then SAIL is getting confused about PIT. Have you assigned a device as PIT? Have you changed that assignment? If the latter, you may have to create a new jail/ejail to run it on the new PIT.

Compiling the high-segment version (starting with version 11.2) ~~xxxxxxxxxxxxxxxx~~ not yet done

WIZARDS: This whole page is for you; this assumes you've read the previous page.

For this, you need the high-segment versions of the libraries listed on the previous page.

High-segment file names: file names of length LEQ 5 have an "h" prepended from the versions discussed earlier in this appendix. For file names of length six, the first letter is replaced with "h". In addition to applying to the names in the above table, note the following:

decl [LIB,awn]	hdecl
edec1	hedec1
jail	hjail
ejail	hejail

Here's a note from uncle bill:

21-Mar-83 0711 BIL highseg

To make a high segment, compile everything with the (H) switch, then load it (still with (H)), SETUWP <cr> and SSAVE <prog>. See E.DO [LIB,BIL] for an example. Wherever you require grnlib, require HGRNLB for the high segment (see PLA.SAI [1,BIL]/3P for an example). HGRNLB.MAK [LIB,BIL] creates the high segment grnlib (as an example of that process). Good luck!

Actually, I find that SETUWP is an incorrect thing to do for EMERGE; no one knows why.

21 Jan 1985 20:05

EMERGE.AWN[UP,DOC]

PAGE 75-1

Appendix 4: Wizardry

This entire appendix is for wizards only.

21 Jan 1985 20:05

EMERGE.AWN[UP,DOC]

PAGE 76-1

adding yourself to getWizard

UMERGE.SAI contains

INTERNAL SIMPLE BOOLEAN PROCEDURE getWizard;

which contains a list of PPNs of those classed as wizards.
Documentation there shows you how to add yourself to that list.

command line switches and the preloaded file

The major benefit of classing yourself as a wizard is that you get a merge file for free when you start EMERGE. This works as follows:

The boolean inDefault is set to TRUE for wizards when EMERGE starts. For indefault=TRUE, EMERGE attempts to read in a default .MF file, the name of which is specified in EMERGE.SAI. If the preloaded file cannot be found, EMERGE continues gracefully.

command line switches

If you are not a wizard but want to have the default input file (for example, just to see how a feature works), then invoke EMERGE with

```
.R EMERGE;I
```

You can turn off the attempt to read the default input file by invoking EMERGE with

```
.R EMERGE;-I
```

The command line switches are parsed by the routine

```
INTERNAL SIMPLE PROCEDURE procInputSwitches;
```

in UMERGE.

LORE: Yes, you can get away with nonsense like

```
.R EMERGE;I/-I/I
```

LORE: If you call out of EMERGE with <CALL> and type ".S" to the monitor to restart, we will check your wizard status before attempting to read in the default input file. You can use the switches as well, as in

```
.S;I
```

(to restart, invoking the default input file), or

```
.S;-I
```

to restart, this time with no input file.

wizardry in menus

α X D (for debug) typed to a parameter menu may provide you with debugging information about the setup of the menu. This will tell you, for example, why the cursor "→" is in a funny column.

See also the entry for α X-W under "Extended commands" in the section "The EMERGE function editor."

File under: A

Date: 21 Jan 1985

Name: John Strawn

Name: John Strawn

Project: J Programmer: AWN

File Name: EMERGE.AWN[UP,DOC]

File Last Written: 20:04 21 Jan 1985

Time: 20:05 Date: 21 Jan 1985

Center for Computer Research
in Music and Acoustics
Department of Music
Stanford, California