NREL/TP-442-7111 • UC Category: 1213 • DE95000293

GPP User's Guide

A General-Purpose Postprocessor for Wind Turbine Data Analysis

Marshall L. Buhl, Jr.

National Renewable Energy Laboratory
1617 Cole Boulevard
Golden, Colorado 80401-3393
A national laboratory of the U.S. Department of Energy
Managed by Midwest Research Institute
for the U.S. Department of Energy
under contract No. DE-AC36-83CH10093

Work performed under task number WE429150

January 1995

NOTICE

Available to DOE and DOE contractors from:
Office of Scientific and Technical Information (OSTI)
P.O. Box 62
Oak Ridge, TN 37831
Prices available by calling (615) 576-8401

Available to the public from:
National Technical Information Service (NTIS)
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650

Notice

I distribute GPP with its source code. You may modify the code if you like, but I ask that you document your changes in the CHANGE.LOG file as you make them. If you distribute a modified version of GPP, please modify the startup screen to warn the user that (s)he is using a modified version of the program. I want neither credit nor blame for your changes.

Disclaimer

I distribute the software described in this user's guide on an "as is" basis, and it is for evaluation purposes only. Please notify me of any problems you have with GPP. I also welcome any ideas and suggestions you may have for improving the product. I can fix bugs and make improvements on a time-available basis only and promise no speedy resolutions.

Acknowledgments

I would like to thank everyone who helped make GPP a reality. Dave Simms made many suggestions for enhancements that changed GPP from being essentially an ADAMS postprocessor to a more generically useful software tool. He alpha-tested parts of the code and suggested improvements in its usability. Dave has also been my greatest supporter—he convinced both management and me that the project was worth doing.

Neil Kelley was another big contributor. He not only provided suggestions and the benefit of his years of experience, but also let me "borrow" his rainflow-counting routines. I would never have understood how NCAR's SPECFT routines worked if it had not been for Neil. I also hope the hole he bit in his tongue while watching me work on GPP instead of his Micon project heals soon.

Speaking of SPECFT, I should also thank Bob Lackman of NCAR for writing this very sophisticated bit of code. It definitely makes my PSD Tool much more versatile than the one I took out of *Numerical Recipes* (Press, et al. 1990).

Alan Wright came up with the requirements for the Azimuth-Average Tool and helped me debug the Fourier coefficient part of the code. Alan also found other bugs while using the code. I appreciate his patience and encouraging words.

Dave Laino, beta-tester extradordinaire, found quite a few bugs and made many suggestions to improve the usability of GPP. George Scott taught me a few UNIX tricks and helped me identify some of my bugs. Chris Courtright and Vallerie Shelton helped me keep the computers behaving properly.

Mechanical Dynamics, Inc., let me use their HP workstation to port GPP to that platform while I was attending an ADAMS/WT training class at their site. The HP has a very different compiler, which forced me to make the code even more portable than it was before.

This user's guide was prepared to document work done at the National Renewable Energy Laboratory in support of the U. S. Department of Energy under contract number DE-AC36-83CH10093.

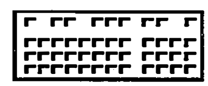
Table of Contents

How to Use This Manual

GENERAL CONVENTIONS

I use different typefaces throughout this manual to help distinguish different concepts. I printed the majority of the body text (including this paragraph) in 11 point Times New Roman. Headings use various sizes of the roman (upright) Arial typeface. Within body text, I also use roman Arial to denote the names of files and other things. When a file name applies to both DOS and UNIX, I will put the name in lower case so that it will work for both operating systems. I use bold **`Courier`** for commands that I expect you to enter into the keyboard exactly as they appear. These lines are indented one quarter inch and are set off from the body text by blank lines. I show screen examples and file excerpts in bold roman `Letter Gothic` type and usually border the screen examples with a rounded box to denote a screen.

KEYBOARD CONVENTIONS



When entering commands at the keyboard, enter text exactly as printed. If you are using a DOS-based personal computer (PC), you may use either upper- or lower-case letters. If you are using a UNIX-based workstation, you must be sure to duplicate the case exactly.

"Naked return," a term used occasionally in this manual, means that you press the Enter key (↵) without typing any previous text. Use naked returns to select default answers or to display the long versions of high-level menus when GPP prompts you with short (one line) menus.

1 Introduction

ABSTRACT

GPP (pronounced "jeep") is a General-Purpose Postprocessor for wind turbine data analysis. The author, a member of the Wind Technology Division (WTD) of the National Renewable Energy Laboratory (NREL), developed GPP to postprocess test data and simulation predictions. GPP reads data into large arrays and allows the user to run many types of analyses on the data stored in memory. It runs on inexpensive computers common in the wind industry. One can even use it on a laptop in the field.

The author wrote the program in such a way as to make it easy to add new types of analyses and to port it to many types of computers. Although GPP is very powerful and feature-rich, it is still very easy to learn and to use. Exhaustive error trapping prevents one from losing valuable work due to input errors. GPP will, hopefully, make a significant impact on engineering productivity in the wind industry.

BACKGROUND

GPP started out as a loose collection of utility programs. As the number of utilities grew, duplication among the programs became excessive. My solution was to write one program that would input the data and then manipulate them in a variety of ways. The first version of GPP included only those original tools and ran only on Silicon Graphics, Inc., (SGI) computers. Since then, others have made requests for many additional features, including the ability to run on a variety of platforms.

To make the code platform independent, I placed all compiler-specific routines into a separate file so that the majority of the code would require no modifications to run on other systems. The compiler-specific routines include all operating system calls such as date and time and all screen output.

As of this writing, GPP runs on UNIX computers from SGI, Sun Microsystems, and Hewlett-Packard (HP), as well as on personal computers (PCs). I have compiled it with Lahey Computer Systems' F77L-EM/32 and Microway's NDP FORTRAN, two PC compilers that are often available in the wind energy industry. These compilers also allow the huge arrays that are necessary for GPP. Because GPP is so platform independent, it should be possible to port it to a new system with less than a day's effort.

GPP offers engineers a user-friendly tool that makes it easy to look at data in many different ways. Experience has shown that researchers need to analyze data with varied techniques. For example, a power spectral density (PSD) plot that compares test data to simulation predictions may show excellent agreement, while rainflow cycle-count spectra tell an entirely different story.

GPP is a menu-driven, question-and-answer type program instead of the old-style batch type program. It is a character-based program instead of one with a graphical user interface (GUI) for several reasons. Although GUI programs are pretty, they are difficult to code, run much slower than character-based programs, and are inherently non-portable. The portability requirement also prevented me from adding graphical output. However, GPP's output files import easily into standard graphics packages.

Because GPP reads entire data sets into arrays, it is *very* fast. Once the data is in memory, a modern PC can perform many analyses in just seconds. Older analysis tools had severe memory restrictions due to the limitations of the technology then available. Because they had to store information on disk, they were very slow.

In GPP, many prompts have intelligent defaults and the less-obvious questions have context-sensitive help. In some cases, there are on-line examples to help users. I put great care into the program's usability and hope I have made it "fool resistant." GPP traps input errors and requests the input again.

The targeted users for GPP are engineers in wind energy research and industry. The program assumes that users are familiar with the concepts and techniques it implements.

Requirements for Running GPP

Most modern PCs can run GPP. Unmodified, GPP needs about 4 MB of RAM and disk storage. Tests on a 33 MHz 486 PC indicated that the program is slow to load, but it is still quite usable. The 60 MHz Pentium PC used to port GPP to the PC runs it quite easily.

To modify GPP on a PC, you will need either the Lahey or Microway compiler or another protected-mode compiler. If you use a different compiler, you will need to modify the routines in either the SYS_PCL.FOR or SYS_PCN.FOR to be compatible with that compiler. You will also need to modify some of the batch files and/or makefiles used to compile the code.

One of my primary uses for GPP is to postprocess ADAMS[1] output. Therefore, I made certain it could run on PCs that run ADAMS and that users can compile it with the NDP FORTRAN compiler needed for the AeroDyn routines I link with ADAMS.

Almost any UNIX workstation can run GPP with few or no modifications to the code. If you want to modify GPP or run it on something other than a MIPS R4400-based SGI system or Sun SPARCstation 10, you will need a FORTRAN compiler. I developed GPP on a 64-bit, R4400-based, SGI Indigo2. It is very fast on this system and on the Sun SPARCstation 10.

[1] ADAMS is a registered trademark of Mechanical Dynamics, Inc.

2 Installing GPP

ACQUIRING GPP

I do not have the resources to distribute GPP on floppy diskettes. You must connect to our computers to download the files you need. This makes it faster and easier for you to get the latest version of GPP when you are ready to use it. You can either ftp to our anonymous FTP (File Transfer Protocol) server or dial into our Kermit server with your PC modem to get the code. Encoded within the file names is the three-digit version number. Check it to see if your version is out of date. GPP prints the version number and date when it starts.

I have archived and compressed DOS files with the ZOO archiver. Please download the latest version with your zoo file. I used tar and compress to archive the UNIX files. Your UNIX system will already have tar and uncompress to unarchive the files.

Anonymous FTP

If you have a live connection to the Internet, you can ftp to our anonymous FTP server. This is the fastest and easiest way to get GPP. Our server's address is ftp.nrel.gov:

```
ftp ftp.nrel.gov↵
```

When the FTP server prompts you for a name, enter:

```
anonymous↵
```

The server will ask you for a password. Enter your full e-mail address at that prompt.

GPP files are in the /pub/gpp directory:

```
cd /pub/gpp↵
```

You will find one file and two subdirectories in that directory. They are README.TXT, dos, and unix. Please read the file:

```
get README.TXT -↵
```

The "-" at the end of the line above tells your local system to display the contents of the file on your screen instead of creating a copy of the file on your system's disk.

You can then move to the appropriate subdirectory you wish to use. There will be another README.TXT file in each subdirectory to explain the files found there. Follow the instructions in these files to download the appropriate archives and unarchive them. When you are through downloading, log out by entering:

```
bye↵
```

After you have unarchived the files, follow the instructions in readme.now to set up GPP on your system.

Kermit

You can use any PC communications program that has a Kermit protocol for binary transfers. The Terminal program that comes with Windows has it. You can even use Kermit on UNIX systems that have dial-out capabilities.

To use Terminal to connect to our Kermit server, you will need to set the proper communications parameters. Select Settings-Communications from the menu. Set the Baud Rate to the speed of your modem (if your modem is 14400, set it to 19200). Set the Data Bits to 8 and the Stop Bits to 1. Set Parity to None and Handshaking to Xon/Xoff. Set Connector to the port your modem is on (for example, COM1:). Make sure there is no ☒ in the Parity Check or Carrier Detect boxes and choose OK.

Next, set the protocol for binary transfers by selecting Settings-Binary Transfers from the menu. Check the Kermit box and choose OK.

Here is a sample session for getting the precompiled version of GPP (what you enter is typeset in `Courier`):

```
ATDT9,1-303-275-4636
CONNECT 14400

Welcome to NREL's dialup server.

To connect to NREL's Gopher/Webb server, enter info at
the login prompt.

nbl login: nrel_access
Password: nrel_94                    (you will not see the password on the screen)

nbl:Top> telnet -8 windcrim
Type ^] (decimal 29) <CR> to return to NetBlazer
Trying 192.88.248.8:23...
Telnet session 1 connected to windcrim


IRIX System V.4 (windcrim)

login: gpp
Password: post-p                     (you will not see the password on the screen)
IRIX Release 5.2 IP17 windcrim
Copyright 1987-1994 Silicon Graphics, Inc. All Rights Reserved.
Last login: Fri Jul 22 09:30:12 PDT 1994 by UNKNOWN@nbl
C-Kermit, 4E(072) 24 Jan 89, AT&T System III/System V
Type ? for help
C-Kermit>? Command, one of the following:
```

```
 !                   bye                 cd                  close
 connect             cwd                 dial                directory
 echo                exit                finish              get
 hangup              help                log                 quit
 receive             remote              script              send
 server              set                 show                space
 statistics          take
C-Kermit>dir
total 5732
total 5624
-rw-r--r--   1 marshall wind      1387 Jul 11 11:18 README.TXT
-rw-r--r--   1 marshall wind   2584902 Jul 21 10:42 gpp_452d.zoo
-rw-r--r--   1 marshall wind    145283 Jul 21 10:37 gpp_4521.zoo
-rw-r--r--   1 marshall wind    146782 Jul 21 10:37 gpp_452n.zoo
-rw-r--r--   1 marshall wind       221 Jul 29 12:03 zoo210.exe
C-Kermit>send README.TXT
Escape back to your local system and give a RECEIVE command...
```

At this point, select Transfers-Receive Binary File from the Terminal menu. Enter the appropriate drive, directory, and file name and choose OK. When the transfer is complete, use an editor such as Notepad to read the README.TXT file you just downloaded.

Tell the remote system to assume that future file transfers will be for binary files. Then, download the zoo archiver and the appropriate GPP file. These transfers will take a very long time—even for a fast modem.

```
C-Kermit>set file type binary
C-Kermit>send zoo210.exe
Escape back to your local system and give a RECEIVE command...
C-Kermit>send gpp_4521.zoo
Escape back to your local system and give a RECEIVE command...
```

After downloading the files, log out of the Kermit server and the terminal server.

```
C-Kermit>quitTelnet session 0 closed: EOF

nb1:Top> logout

NO CARRIER
```

Follow the instructions in the README.TXT file to unarchive the GPP files and set up the program.

COMPILING GPP

You may want to make some changes to GPP to tailor it to your needs. The distributed version of GPP can read files with up to 20 columns and 20,000 rows of data. GPP has two matrix arrays dimensioned for that size and two other vector arrays that hold 20,000 values each. These large arrays make GPP very big. If you need more or less storage, you can change the MAX_COLS, MAX_COL1 (MAX_COLS-1), MAX_ROWS, and MAX_ROW2 (half MAX_ROWS) parameters in data.inc and recompile the program.

Once all our compilers have the Fortran 90 ALLOCATE command (only Lahey's has it now), I will make the arrays dynamically allocatable, so this process will be unnecessary.

DOS-Based Personal Computers

By this point, you should have downloaded the appropriate archive from one of the NREL servers and unarchived it. If you do not have an extended-memory compiler, you cannot compile GPP, and you must use the precompiled version of GPP. If this is the case, skip ahead to the next section, "**Running GPP**."

If you have either Lahey's F77L-EM/32 or Microway's NDP FORTRAN compiler, you can recompile GPP with the supplied files. If you have another extended-memory compiler (for example, Microsoft FORTRAN), you will have to modify one of the makefiles or the CL.BAT file to build GPP. You may also need to change the routines in either SYS_PCL.FOR or SYS_PCN.FOR to port the code to your new compiler. See "**Porting to an Unsupported Computer**" for advice on modifying the SYS_???.FOR file. If GPP does not function properly without changes to the other files, then GPP is not quite as portable as I would like. Please let me know of any required changes to other source files.

Building GPP is easiest if you use makefiles. With makefiles, you can make changes to some program modules and recompile only those that you changed. The supplied CL.BAT file (for NDP users) recompiles all FOR files even if you did not change some of them. Lahey compilers come with a MAKE utility; if you have one of these compilers (even the non-extended version, F77L), you can use Lahey MAKE for the NDP compiler as well. You can also get MAKE utilities from other sources. I recently got it with the MKS Toolkit from Mortice Kern Systems Inc. for approximately $250.

UNIX-Based Workstations

This section applies to owners of supported workstations. If you are not using either a HP, SGI, or Sun, you should skip to "**Porting to an Unsupported Computer**" for advice. If you want to make changes to GPP, you will need a FORTRAN compiler installed on your workstation. If your workstation does not have a FORTRAN compiler, you can install and compile GPP on another workstation with the same operating system (OS) that does have a compiler. Then copy the executable (gpp) and help file (helpfile.txt) to your machine.

For supported workstations, you should not need to make significant changes to the makefile, but you may want to adjust the compiler flags. The makefile contains lines for both debugging and optimization. Just comment out the unwanted FFLAGS line with a pound sign (#).

Porting to an Unsupported Computer:

If you want to compile GPP with an unsupported computer or compiler, you will probably have to modify the routines in one of the sys_* source files. You should copy that file and give the new one a similar name with the letters after the "_" denoting the platform and/or compiler. To use this file, you should modify the makefile, batch, or script files to account for the name change. If you have to make changes to other source files to port GPP, please let me know so that I can make the code even more portable.

Within the sys_* file, you will likely have to make changes to many routines; these routines are described on the following page.

CURDATE	Fix this function to call the system date routine that comes with your compiler.
CURTIME	Similar to CURDATE.
DIR	Unless you are using an operating system other than DOS or UNIX, you should not need to change DIR. Just use the appropriate one.
FLUSHOUT	This routine flushes the screen buffer so that informational messages do not need to wait for the buffer to fill before the computer displays them on the screen. Some compilers use the FLUSH function and others write an end of file to the appropriate unit with ENDFILE. FLUSHOUT may need to be changed, or, in the case of HP-UX, you may need to call a system routine from OPEN_CON to set the buffer size of standard output to zero.
GET_ENV	This routine examines the OS's environment to get the value of one of its variables. Some compilers supply a GETENV (without the "_") routine for this purpose. Otherwise, you will have to issue some OS commands with SYS_CALL to put a copy of the environment variables into a file and then read the file to find the requested variable. See the supplied file SYS_PCL.FOR for an example of this method. If your OS does not use environment variables, you may have to make some major kludges.
GET_HOME	This routine examines the OS's environment to get the user's home directory. Some compilers have a simple routine to do this. Others call GET_ENV for the HOME environment variable. You may not need to change this routine.
GET_SYS	This routine sets some system-specific information. It must supply GPP with the name of the user's command shell (for example, csh or COMMAND.COM). It also sets the maximum length of a file name (without an extension) and the path separator (like "\" for DOS, or "/" for UNIX). It should set a three-letter mnemonic to designate the name of the platform and/or compiler (probably the same as the second part of the sys_* file name).
GETONAME	GPP uses this routine to get the name of the user's options file. It must begin with the path separator. Because so many Windows programs use INI files, I chose \GPP.INI for DOS-based systems. The C shell under UNIX usually uses rc (run command) files for this sort of thing, so I use /.gpprc for UNIX systems. Use whatever file name is appropriate for your system.
GPPALARM	You probably do not need to change this routine, but if your system has advanced sound features, you may want to take advantage of them. GPP currently uses the ASCII "bell" character (^G) to sound the alarm.
GPP_WAIT	GPP can be asked to wait a specific amount of time with this routine. I eliminated all references to this routine from GPP, but I may want to use it in the future. You can remove the routine if you like, or modify it so that it accesses the proper system routines to get the time. Be sure to account for going past midnight.

OPEN_CON	For some compilers, GPP re-opens standard output (the display) so that it uses the proper options for displaying formatted text to the screen. You must use options that allow you to generate prompts that are not followed by new lines. They must also allow new text to be appended to the end of an existing line. This routine is not necessary for some compilers. If you do not need it, just eliminate all executable statements but the RETURN. This is also where you should put the call to the routine that sets the display's buffer size to zero if that is necessary. (See FLUSHOUT above for more information.)
READLIST	This routine should not really be in the sys_* file, but the NDP compiler has a bug (or feature) that makes it think there is one extra entry in the list. You should probably use one of the other sys_* files for your system.
SYS_CALL	GPP executes OS commands with this routine. Most compilers supply a routine called SYSTEM for this function. This routine fails when using an NDP-compiled GPP under Microsoft Windows.
TELLHOME	This routine gives you advice on how to set the HOME directory. It is not needed for UNIX, because the HOME directory is set by default. DOS users (and perhaps other operating systems' users) may need to set it themselves.
WR_BL	This routine writes out the beginning of a multipart line to the screen. If you are using standard FORTRAN carriage control for the screen (where the first character means something special like form feed), be sure to prepend a space to the string. The NDP compiler will not allow you to concatenate dummy-argument strings, so I had to copy the string into a local variable to get this to work.
WR_EL	To terminate a multipart string and generate a new line, GPP calls WR_EL. If you modify this routine, be sure to tell the compiler to append the new string to the end of the previous line. FORTRAN carriage control uses a prepended ampersand ("&") to append text.
WR_ML	This routine is similar to WR_EL, but it does not follow up with a new line.
WR_NR	This routine is used to write a string to the screen without generating a new line (no return) afterward.
WR_SCR	This routine writes one line of text to the screen. Be sure to account for carriage control if necessary.

RUNNING GPP

DOS-Based Personal Computers

To make GPP easy to run, either add the GPP directory to the search path in your AUTOEXEC.BAT file or move the executable and HELPFILE.TXT to a directory that is already in the path. You can, instead, create a batch file (call it GPP.BAT) to start GPP and put the batch file in one of your searched directories.

After doing the above, just type the following to run GPP:

```
GPP↵
```

UNIX-Based Workstations

If you correctly set up the symbolic links in the setup step, you should be able to start GPP from any directory with:

```
gpp↵
```

3 Using GPP

Background

Once you understand the working philosophy of GPP, you should be able to use it fairly effortlessly. I have tried to make GPP as user-friendly as possible; it contains a lot of on-line, context-sensitive help and it's good at trapping input errors. To aid your efforts to learn GPP, I have included a glossary of the terms I use in the manual and in the program. You can find it near the back of this manual.

GPP uses a combination of menus and a question-and-answer coding style. At higher levels, you choose tools from various menus to tell GPP what tasks to perform. Once GPP finishes processing the data with the specified tool, it returns to an appropriate menu. GPP removes tools that are not suitable in a given situation from the menus to avoid confusion. For example, it is meaningless to use the Azimuth-Average Tool option before GPP has read in some data.

GPP reads ASCII data files with white space or comma-separated columns that represent different channels of data. It also recognizes several specific types of data files and can modify its behavior to streamline their use. The code currently recognizes ADAMS[2], YawDyn, and GPP Merge Tool output files. When GPP reads one of these special files, it does not need to inquire about headers and columns. It also parses out column headings so the various tools can display information based on column name rather than column number. GPP classes files that it does not recognize as "generic."

You would rarely change some tool options, so GPP saves them in a permanent file. You can change these options from any of the high-level menus. Examples of these options are the identifying strings for ADAMS and YawDyn files. See the subsection on the Options Tool on page 3-18 for more information on the options.

It is important that you understand the different purposes for the storage arrays. GPP has two storage areas for data—a primary array and a secondary array. When working with the primary array, you use tools that (generally) convert time-series data to another domain. These tools write their results to external files. An example of a primary array tool is the PSD (power spectral density) Tool. This tool converts from the time domain to the frequency domain. Another is the Rainflow-Cycle-Count Tool, which generates cycle counts versus cycle amplitudes. The one tool that does not change domains is the Interpolation Tool. "**The Tool Menu**" section on page 3-3 provides a comprehensive list of tools that process the primary array.

The Merge Tool builds up the secondary array by merging columns of data from one or more primary arrays. All the tools available from the Merge Tool Menu leave the data in the time domain, although some leave a series of blocks of time instead of a steadily increasing time. This is where GPP does its filtering, including decimation, low-pass filtering, and cycle limiting. "**The Merge Tool Menu**" sec-

[2] ADAMS is a registered trademark of Mechanical Dynamics, Inc.

tion on page 3-4 provides a comprehensive list of tools that process the secondary array. When you finish building up and "filtering" the secondary array, you can move it to the primary array and/or write it to an external file. You can read these external files into either array later for additional processing.

When GPP presents you with a list of choices (enclosed in parentheses) for a menu, the one prefixed with an equal sign is the default response. Entering a naked return (an ↵ with no other text) will select that option. Sometimes GPP will ask you to enter a number (or series of numbers). In this case, GPP will enclose the default response in square brackets `[default]`. Except for file names on UNIX systems, all input to GPP is case insensitive.

THE MENUS

The Startup Screen

When you start GPP, it displays a startup screen with some program information and a menu. Figure 3-1 shows an example of the startup screen.

```
Running the General-Purpose Postprocessor, GPP, v4.57pcl, 12/05/94.
     Copyright (C) National Renewable Energy Laboratory 1994.

Startup Menu:

 C - Choose a primary file to activate.
 D - Divide an ADAMS .out file.
 E - Escape to a new command shell.
 Q - Quit GPP.
 O - Set options.
 ? - Help.

[No primary file has been activated]  Enter "?" for help.

 Enter Startup Menu choice (C,D,E,O,Q,?) >
```

Figure 3-1. The Startup Screen.

The first line tells you that you are running GPP. It is version 4.57 for the PC using the Lahey compiler. The effective date of this version is December 5, 1994. The screen lists the obligatory copyright information and then the long version of the Startup Menu. After that, it says you have not activated a primary file and that you can get help on these (and on all high-level tools) by entering a question mark. See the Help Tool subsection on page 3-18 for more information on obtaining help. The last line is the short form of the Startup Menu. To save time and screen space, GPP displays long menus only when first starting up or when you enter a naked return at one of the high-level menu prompts.

Probably the first thing you will want to do is get data into GPP. If you want to analyze an ADAMS out file, you should call up the Divide Tool by entering a `D`. This will break out the different ADAMS requests from the file and create individual files that you can read into GPP. If you have done so, or if you want to look at other data, select the Choose Tool to choose a file and read it. For more information on the Divide Tool and Choose Tool, please see the "**System Tools**" section on page 3-15.

The Tool Menu

After choosing a file, you will see the short form of the Tool Menu on the screen. In the example below, GPP tells you that you have read in a primary data file, that its name is adams.rq, and that it is an ADAMS output file. If you enter a naked return, you will see the long version of the menu, which is shown in Figure 3-2.

```
Tool Menu:

 A - Azimuth average the primary data.
 B - Bin the primary data.
 C - Choose a primary file to activate.
 D - Divide an ADAMS .out file.
 E - Escape to a new command shell.
 H - Histogram of the primary data.
 I - Interpolate the primary data.
 L - Least squares fit the primary data.
 M - Merge files.
 O - Set options.
 P - Generate PSDs of the primary data.
 Q - Quit GPP.
 R - Rainflow count the primary data.
 S - Show statistics of active data.
 ? - Help.

[The ADAMS file "adams.rq" is now the primary file]

 Enter Tool Menu choice (A,B,C,D,E,H,I,L,M,O,P,Q,R,S,?) >
```

Figure 3-2. The Tool Menu.

As you can see, the menu is quite a bit longer than the one in Figure 3-1. The tools seen on the Startup Menu are still available, but GPP added some new tools that process the chosen data in the primary array and produce some sort of output file. Please see the "**Analysis Tools**" section on page 3-4 for more detail on these tools.

A useful tool here is the Statistics Tool. It is slightly different from the other tools that manipulate the primary array. It does not produce an output file (the Choose Tool produces an st file with full statistics when you read in the primary data), and it works on both the primary and secondary arrays. A detailed description of the Statistics Tool is provided on page 3-11.

The Merge Tool Menu

Once you have read a primary data file, you can choose the M option to go to the Merge Tool Menu. The short form of the Merge Tool Menu repeats the information about the primary file and tells how many primary files you have already merged into the secondary array. Again, if you do a naked return, you can see the long form of this menu, which I show in Figure 3-3.

```
Merge Tool Menu:

  C - Choose a primary file to activate.
  D - Divide an ADAMS .out file.
  E - Escape to a new command shell.
  M - Merge columns from primary array.
  O - Set options.
  Q - Quit merging files.
  S - Show statistics of active data.
  ? - Help.

[The ADAMS file "adams.rq" is now the primary file]
[No files have been merged into the secondary array]

Enter Merge Tool Menu choice (C,D,E,M,O,Q,S,?) >
```

Figure 3-3. The Merge Tool Menu when the secondary array is empty.

The tools from the Startup Menu are still available, some new tools appear, and most of the tools that manipulate the primary data are gone. Only the Statistics Tool remains. Once you merge some data into the secondary array, this tool will let you see the statistics of that array. Otherwise, you will get statistics for the primary array.

The Q option will let you leave the Merge Tool Menu, but not GPP. If you have merged data into the secondary array, GPP will ask you if you want to move the merged data to the primary array. GPP will also ask if you want to write these results to a Merge Tool file. You can read Merge Tool files into the primary array just like any other data. Because GPP is familiar with the format of Merge Tool files, it knows how to extract the heading and column titles to make GPP easier to work with.

Figure 3-4 shows what the full Merge Tool Menu looks like after you have merged some data into the secondary array. You now see tools for manipulating the secondary array. These tools enable you to "filter" the data in various ways, trim rows from the array, convert units, and write the merged results to a file. You can look in the "**Filter Tools**" section on page 3-12 for information on the new tools.

ANALYSIS TOOLS

GPP's analysis tools perform the mathematical and engineering analyses for which I wrote GPP. The other tools "filter" the data or are system related.

```
Merge Tool Menu:

 C - Choose a primary file to activate.
 D - Divide an ADAMS .out file.
 E - Escape to a new command shell.
 F - Filter the primary data.
 M - Merge columns from primary array.
 O - Set options.
 Q - Quit merging files.
 S - Show statistics of active data.
 T - Trim rows from the secondary data.
 U - Convert units in the merge data.
 W - Write out the merge file.
 ? - Help.

[The ADAMS file "adams.rq" is now the primary file]
[One file has been merged into the secondary array]

Enter Merge Tool Menu choice (C,D,E,F,M,O,Q,S,T,U,W,?) >
```

Figure 3-4. The Merge Tool Menu when there is data in the secondary array.

The Azimuth-Average Tool

You can bin your data against azimuth with the Azimuth-Average Tool. For ADAMS and YawDyn files, azimuth must be in the second column. For Merge Tool files and generic files, the Azimuth-Average Tool prompts you for the azimuth column. If your azimuthal data are in radians, use the Units-Conversion Tool to convert them to degrees.

This tool also produces the Fourier coefficients (converted to magnitude and phase) of the averaged data. You can compute these coefficients for as many as ten frequencies, which is the default. This feature makes the Azimuth-Average Tool superior to the Bin Tool for azimuth averaging.

For long time series with varying wind conditions, it may not be meaningful to azimuth average the entire data set. Consider using the Limit Tool (a subtool of the Filter Tool) to limit the data to cycles with similar conditions. Figure 3-5 is an example of a plot generated using data from the Azimuth-Average Tool.

The results file has an aa extension.

See also: Bin Tool, Limit Tool, Units-Conversion Tool.

The Bin Tool

The Bin Tool allows you to bin one or more columns of the primary data against an independent column. It is useful for tasks such as plotting power versus wind speed or lift coefficient versus angle of attack. You could even use the Bin Tool to compute azimuth averages, but the Azimuth-Average Tool has additional features.

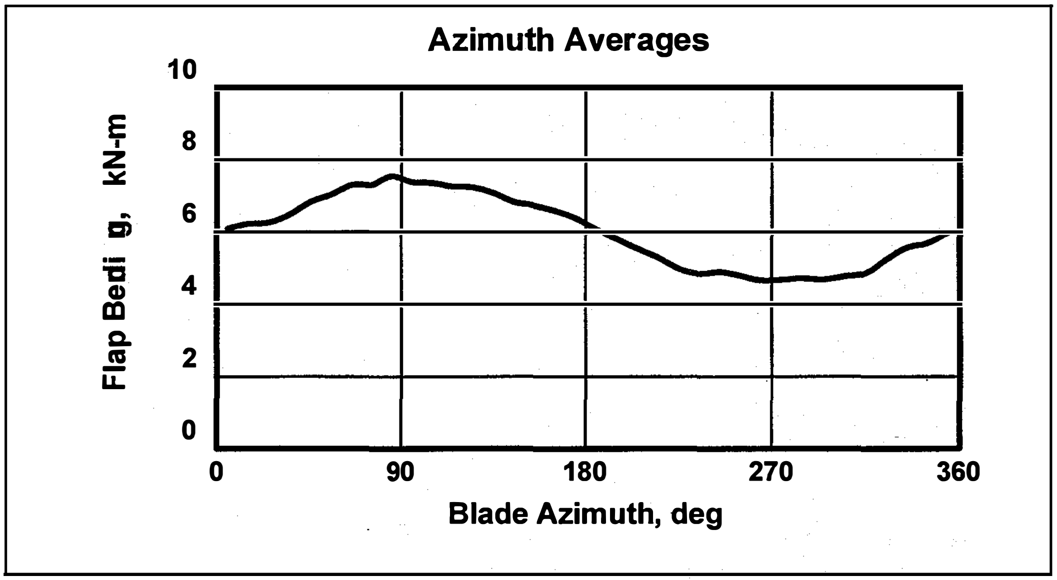


Figure 3-5. Sample plot using data generated by the Azimuth-Average Tool.

Like the Azimuth-Average Tool, it may be more meaningful to limit long time series to blocks or cycles that have similar conditions with the Limit Tool. Figure 3-6 is an example of a typical power curve using data generated by the Bin Tool.

The results file has a bi extension.

See also: Azimuth-Average Tool, Limit Tool.

The Histogram Tool

You can generate histograms (or probability density functions) of your data with the Histogram Tool. The tool writes a two-column table for each requested primary data column to the results file. The table represents the x- and y-values for the analyzed column. For each requested column, the tool takes the domain between the minimum and maximum and divides it into bins of equal size. You tell the tool how many of these bins you want and there will be a row in the output table for each of them. The x-value for each bin is its midpoint. The y-value for each bin is the fraction of the total data points for the column that fall within the bin. The area under the curve defined by the table is unity. Like some of the other analysis tools, it may be a good idea to limit your data to cycles with similar conditions.

For ADAMS or YawDyn files, you can select which columns to process, but GPP will not process time or azimuth. For GPP Merge Tool and generic files, you can select any set of columns. Figure 3-7 is an example of a plot generated using data from the Histogram Tool.

The results file has a hi extension.

See also: Limit Tool, Merge Tool.

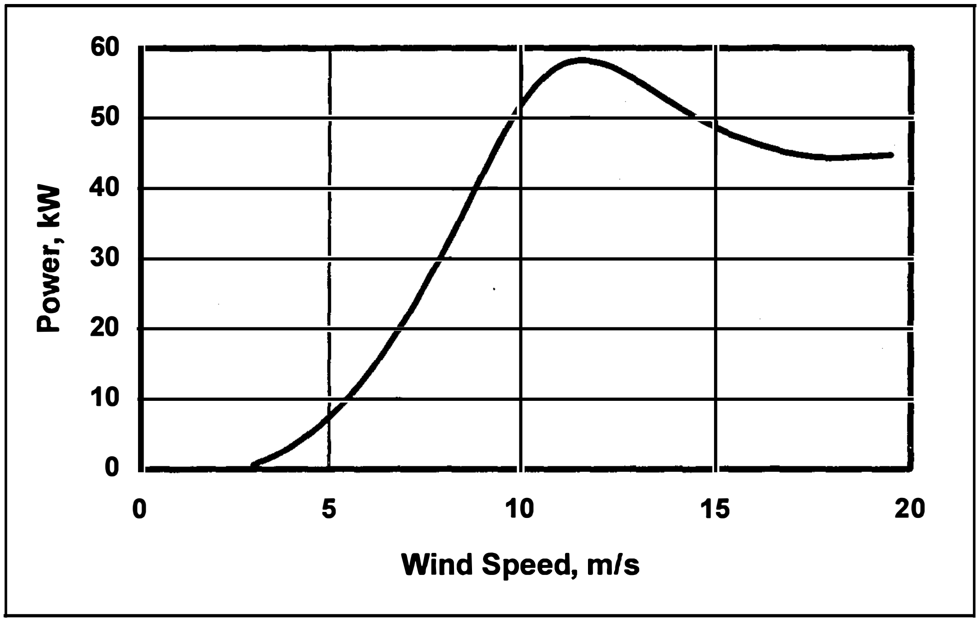


Figure 3-6. Sample power curve using data generated by the Bin Tool.

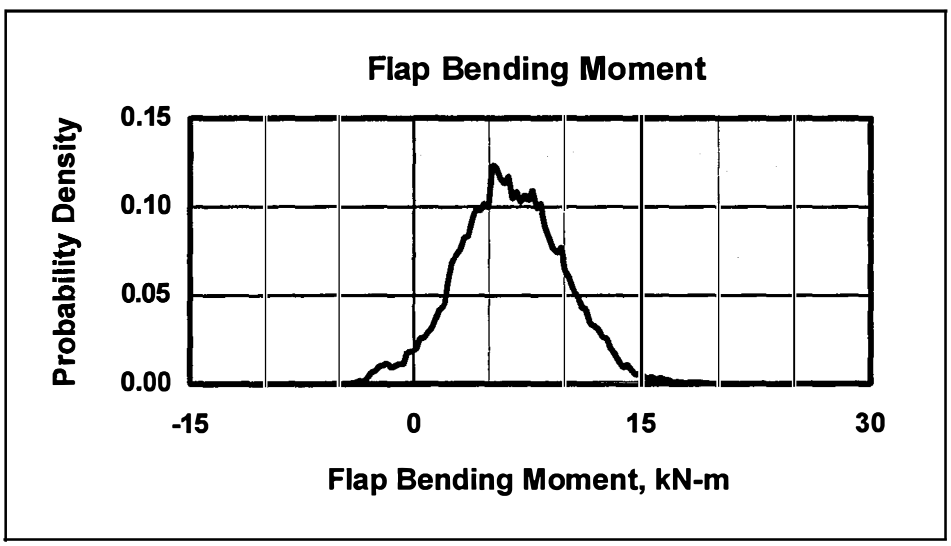


Figure 3-7. Example of a plot of Histogram Tool results.

The Least-Squares-Fit Tool

The Least-Squares-Fit Tool allows you to fit the primary data with polynomials of up to the tenth order. The tool uses modified versions of the Singular Value Decomposition routines from *Numerical Recipes* (Press, et al. 1990).

The tool generates a table of polynomial coefficients for each selected column. It also computes the χ^2 goodness of fit. It writes these tables to the results file.

I tested these routines using data generated by evaluating known polynomials. Results for polynomials greater than the third order did not fit the data well. Results for low orders were quite excellent. I may look for a better algorithm for this tool.

The results file has an ls extension.

The PSD Tool

You can generate power spectral densities (PSDs) of your data with the PSD Tool. It uses routines derived from GenPro, an old, batch-oriented, generalized post-processor developed at the National Center for Atmospheric Research (NCAR). Robert Lackman of NCAR developed much of GenPro, including the SPECFT routines it uses to generate PSDs. The SPECFT routines call another package, REALFT, that NCAR acquired elsewhere. The Fast Fourier Transform (FFT) algorithm in REALFT is desirable in that it does not limit the number of input points to powers of two[3]. I heavily edited the SPECFT routines I got from NCAR to eliminate unused features.

The PSD Tool removes the means from each time series before generating the PSDs. It also tapers the ends of the data with a cosine rolloff and zero-fills the rest of the array that holds the data. You have the option to detrend the data with a straight line, which *Numerical Recipes* (Press, et al. 1990) recommends. You can set (or clear) this option with the Options Tool.

The default PSD technique uses band smoothing. Other PSD types are cosine (Hamming), triangular (Bartlett), and rectangular windows. Rather than window averaging, you can also choose logarithmic spacing. Initial tests indicate that there is very little difference in the results for different PSD types. An advantage of band smoothing is that you can specify the number of output frequencies. The Options Tool lets you set the type of PSD technique. Table 3-1 explains the benefits of each technique. Figure 3-8 shows a sample PSD plot using data generated by the PSD Tool with band smoothing.

The results file has a ps extension.

See also: Choose Tool, Merge Tool, Options Tool.

---

[3] In porting the REALFT routines to GPP, I discovered a bug in the kernel for the radix prime transform. The only work-around we found for this bug is to make sure that REALFT does not call this transform. REALFT will call it only if the number of data points factors down to at least one prime number greater than five. I modified SPECFT so that it will pad the time series with zeroes to fill the array that holds it. This array *must* be dimensioned to a length that factors nicely. The parameter that determines the array length is called MAX_ROWS and it is set in the file data.inc. This does not impact your time series length—just the array dimension.

Table 3-1. Types of PSDs available for GPP.

Band Smoothing	Band smoothing allows you to specify any number of output frequencies (up to #points/6). This allows you to reduce the volume of output and is a good choice if you have a lot of data.
Cosine Windows	Using Hamming-style cosine windows is the best choice for random signals. Use this type of PSD if you are unsure. The technique produces half as many output frequencies as there were data points.
Logarithmic Spacing	Logarithmic spacing of output frequencies will give you less resolution at high frequencies.
Rectangular Windows	Rectangular windows are a good choice if your data have discontinuities or singularities. The technique produces half as many output frequencies as there were data points.
Triangular Windows	Triangular (Bartlett) windows give good results for deterministic signals like sine waves. The technique produces half as many output frequencies as there were data points.

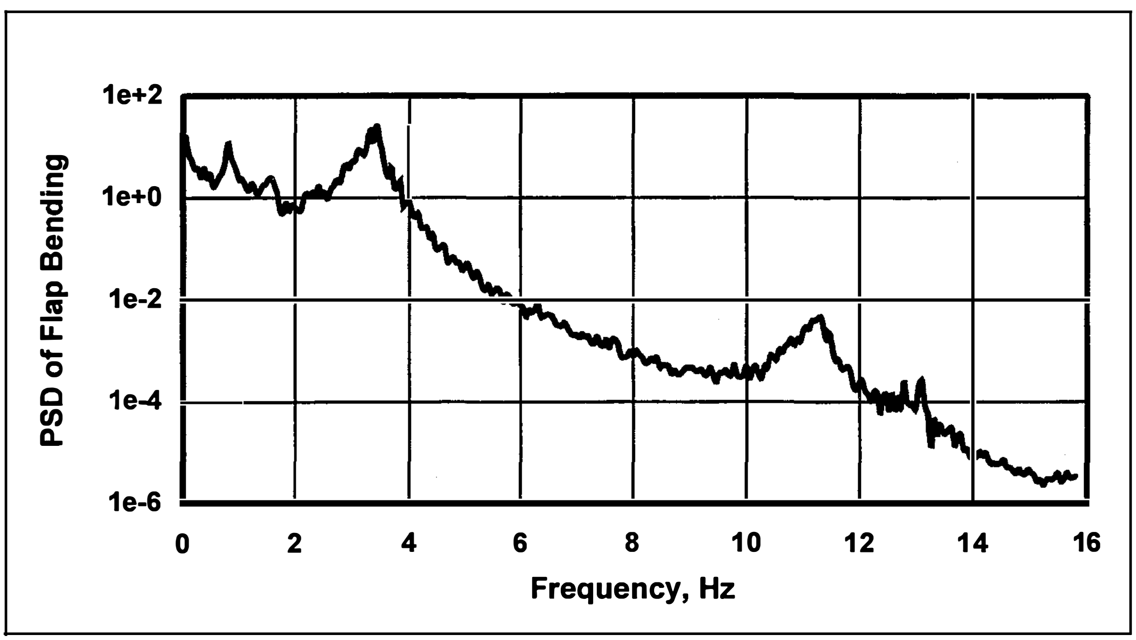


Figure 3-8. Example PSD plot using data generated by the PSD Tool.

The Rainflow-Cycle-Count Tool

You can rainflow cycle-count your data with the Rainflow-Cycle-Count Tool. This tool implements the vector-based rainflow counting algorithm derived from Downing and Socie (1982). It incorporates a modification by Okamura, et al. (1979) to account for half cycles. The tool rainflow counts a time history as it occurs and identifies the same cycles as the two-pass algorithm that requires that the history be rearranged. After the tool identifies the cycles, it bins them to generate curves of cycles per the number of seconds specified for the count period versus the peak-to-peak cycle amplitudes. You can specify the count period with the Options Tool.

Neil Kelley of NREL originally coded this routine, but I eliminated parts of the code unneeded by GPP. I also converted it to my programming style and simplified it to count alternating (or range) cycles only.

Figure 3-9 shows a sample plot of data generated by the Rainflow-Cycle-Count Tool. If you look at the two right-most data points, you will see that they have the same y-value. This is because this algorithm closes all incomplete cycles. I consider these artificially closed cycles to be very misleading, so I normally delete them. These cycles will have a rate that is the inverse of the elapsed time. The Rainflow-Cycle-Count Tool does not output any bins that have no points. This eliminates zeroes from the series and makes log-linear plots possible.

For ADAMS and YawDyn files, GPP does not give you the option of processing time or azimuth. For Merge Tool files, you must ensure that time is in the first column in order to use this tool. For Merge Tool and generic files, only time is eliminated from possible processing.

The results file has an rf extension.

See also: Merge Tool, Options Tool.

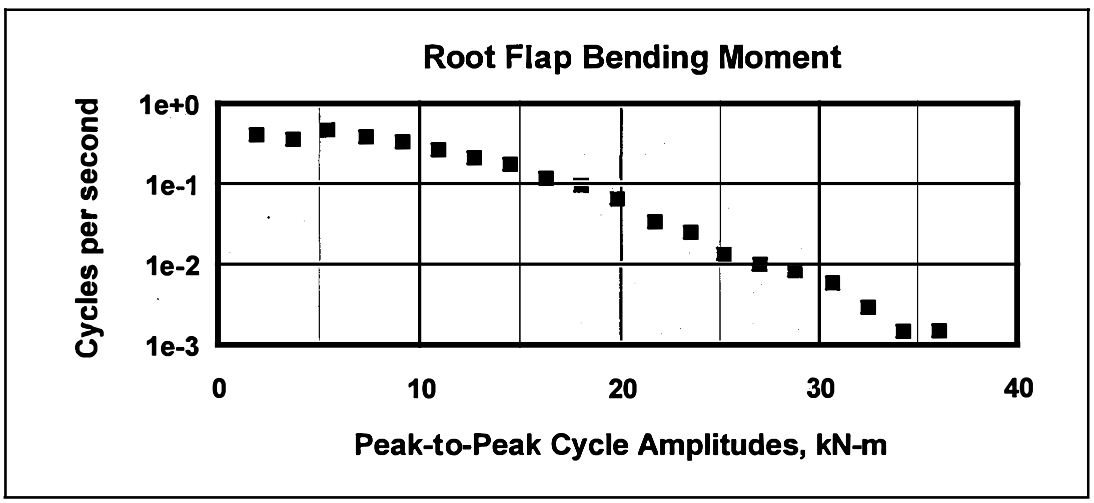


Figure 3-9. Example plot using data generated by the Rainflow-Cycle-Count Tool.

The Statistics Tool

The Statistics Tool displays statistics for the data arrays. These statistics include minimum, mean, maximum, standard deviation, and skewness values. They also include the data record numbers on which the minima and maxima[4] occurred. Figure 3-10 shows a sample statistics screen.

Because the data came from a familiar file, the Choose Tool knew how to determine the column headings. Thus, the tool includes column descriptions in the output. If this had been a generic file, the first column in the table would have been the column number. This is another example of how GPP changes its behavior for familiar file types.

If there are data loaded into the secondary array, the Statistics Tool will ask you whether you want to look at the statistics for the primary or secondary data. The default is the secondary data, as this is most likely what you would be working with at the time. If there are no secondary data, this tool will automatically show the primary array statistics. The Choose Tool calculates the statistics for the primary array after it reads in the data. After each merge action, the Merge Tool calculates the statistics for the secondary array. Accessing the Statistics Tool only displays statistics—it does not compute them.

When the Choose Tool reads a primary file, it also includes a kurtosis calculation and writes all of the statistics to a file with the extension st. Figure 3-11 shows an example of this file. Here, you can see that GPP tells you where it got the data and when it generated the statistics. It includes the original heading and sets it off with dashed lines. It then lists the statistics for each column of data. As you can see, GPP writes the statistics file with one more significant digit than you saw in the screen display of the same data.

See also: Choose Tool, Merge Tool.

Statistics for data from "adams.rq":

Parameter	Minimum (Index)	Average	Maximum (Index)	Std. Dev.	Skewness
Time	5.0 (1)	35.	65. (1921)	17.	-5.63E-07
AZIMUTH	-1.80E+02(1311)	0.50	1.80E+02(782)	1.04E+02	-5.89E-03
BLD_1_RF	2.6 (1)	6.7	11. (146)	2.7	0.13
BLD_1_RE	-4.3 (19)	1.9	8.1 (243)	4.2	2.12E-02
LSS_TOR	5.5 (19)	6.3	6.9 (25)	0.36	-0.18
POWER	-34. (25)	-31.	-27. (19)	1.8	0.18
YAW_MOM	-6.54E+03(17)	-5.89E+03	-5.46E+03(5)	1.39E+02	0.22
FA_THRST	-1.58E+02(796)	-7.10E-02	1.6 (8)	3.6	-43.
SS_THRST	-3.9 (796)	-8.06E-02	2.4 (609)	0.43	-1.6

Figure 3-10. An example display from the Statistics Tool.

[4] It may be useful to note that the index for the time maximum is the number of records in the file.

```
Statistics for data from "adams.rq":
[Generated by GPP on 01-Jun-94 at 08:54:35]

Original heading:
-----------------------------------------------------------------------------
ADAMS/View model name: micon65_v3.21
Request Number        1

Units are in seconds. degrees. kN-m. and kW.
-----------------------------------------------------------------------------

Parameter   Minimum  (Index)   Average    Maximum  (Index)   Std. Dev.  Skewness   Kurtosis
--------   ----------------   ---------  ----------------  ---------  ---------  ---------
Time          5.00    (    1)   35.0        65.0    ( 1921)   17.3      -5.629E-07  -1.20
AZIMUTH      -180.    ( 1311)  0.495       180.    (  782)   104.      -5.887E-03  -1.21
BLD_1_RF      2.63    (    1)   6.72        10.8    (  146)   2.71       0.129      -1.48
BLD_1_RE     -4.33    (   19)   1.92        8.10    (  243)   4.22       2.118E-02  -1.45
LSS_TOR       5.54    (   19)   6.34        6.92    (   25)  0.357      -0.177      -1.41
POWER        -34.2    (   25)  -31.3       -27.3    (   19)   1.77       0.177      -1.41
YAW_MOM    -6.544E+03(   17) -5.893E+03  -5.456E+03(    5)   139.       0.223     -0.385
FA_THRST     -158.    (  796) -7.100E-02   1.60    (    8)   3.60      -43.4       1.894E+03
SS_THRST     -3.93    (  796) -8.060E-02   2.36    (  609)  0.432      -1.60        17.2
```

Figure 3-11. An example statistics file.

Filter Tools

I use the term "filtering" to mean more than noise filtering. I use the term to mean something that does not let everything get past it. A low-pass filter reduces noise in your data and a decimator reduces the volume of your data by eliminating all but every nth record.

The Filter Tool

The Filter Tool is available from the Merge Tool Menu after you have used the Merge Tool to load the secondary array. The tool allows you to select subtools that let you average, decimate, or limit data in the secondary array. A future release may let you low-pass, high-pass, band-pass, or notch filter your data. One of the subtools is the Limit Tool, which lets you set up criteria for blocks or cycles of data that the tool will keep or delete. A similar tool is the Trim Tool, which allows you to throw away blocks of data you do not want to process.

See also: Limit Tool, Merge Tool, Trim Tool.

The Interpolation Tool

The Interpolation Tool allows you to interpolate dependent columns of the primary array using a new set of independent values (a new column 1). You can read the new independent values from a single-column file or specify a minimum, maximum, and step size. You may use either linear or cubic-spline interpolation. I used modified versions of the cubic-spline routines found in *Numerical Recipes* (Press, et al. 1990). I used natural splines for the endpoints so the second derivatives there are zero.

When the interpolation is finished, the tool tells you how many points it had to extrapolate. You probably want this number to be small or even zero.

Example:

> Suppose you have two files with different time steps and want to merge them using the time history of the first file. You can use the Merge Tool to select only the time column and save the result to a file with the Write Tool. Then, read in the second data file with the Choose Tool. Start the Interpolation Tool and tell it to get the new independent series from a file and give it the name of the file you generated with the Write Tool. The file created by the Interpolation Tool will be compatible with the first file so you can merge them together.

The results file has an in extension.

See also: Merge Tool, Trim Tool, Write Tool.

The Limit Tool

The Limit Tool is very powerful. It allows you to specify which blocks of data you want to keep for later processing. Blocks can be of any length (even a single record), and you can specify them by number of records or amount of time.

You can also specify block length as being a single rotor cycle. The Limit Tool does not assume a constant RPM machine. It defines a cycle boundary as being when the azimuth goes from greater than 300 degrees to less than 60 degrees. If your data goes from -180 to 180 degrees, the tool will map the negative values as 180 to 360 degrees. The Limit Tool uses this mapping only to determine cycle boundaries. It will not change the azimuth values in your data. If your azimuth data are in radians, use the Units-Conversion Tool to convert them to degrees.

The Limit Tool asks you for a list of criteria that can be in a stored file or entered through the keyboard. Each criterion consists of a column, a minimum, a maximum, and a type. The type can be the minimum value, the mean, the maximum value, the standard deviation, or every value of a candidate block. Entering the type as a negative value tells the Limit Tool to pass only those blocks for which the statistic lies outside the specified range. Table 3-2 shows the values you enter for different types of limiting factors.

Table 3-2. Types of Limiting Factors.

1	The minimum value in each block
2	The mean of each block
3	The maximum value in each block
4	The standard deviation of each block
5	All values in each block

Examples:

Suppose you want to exclude all rotor cycles where the wind speed (channel 5) exceeds 15 mps even once over the cycle. Using 500 as an arbitrarily large number, you would limit by cycles and specify the limiting criterion as:

```
5,15,500,-3
```

Suppose you want to find all 1-minute time blocks where the wind direction (channel 6) changed little. In this case, you should limit by blocks of 60 seconds and may want to specify that all values in the block lie between -10 and +10 degrees:

```
6,-10,10,5
```

Suppose the azimuth (channel 2) goes from -180 to 180 degrees. If you want to include all records that lie outside the tower shadow, you would limit by individual records and specify the limiting criterion as:

```
2,-170,170,5
```

See also: Filter Tool, Interpolation Tool, Trim Tool, Units-Conversion Tool.

The Merge Tool

This tool allows you to select columns from the primary array and put them into the secondary array. After transferring columns from the primary array, you can read in another primary file and merge columns from it into the existing secondary array. You can thus merge multiple files into one file. If you already have data in the secondary array, the Merge Tool will ask you if you want to erase the old data and start anew.

When you are done using the Merge Tool, you can move the secondary array to the primary array and/or write it to a GPP Merge Tool file. After you move the secondary array to the primary array, you can use the other tools on the merged data. You can also read merged files using the Choose Tool and analyze them like other files. GPP recognizes Merge Tool files and takes advantage of their structure as it does for YawDyn or ADAMS files.

When you merge data into the secondary array, the Merge Tool creates a header that gives you a history of merging operations. This header tells you which files you merged and when you merged them. The column labels indicate which file they came from and what the original columns were. The format for these labels is file#-col#[5]. If the original data files came from ADAMS or YawDyn, the Merge Tool will replace the column number with the column name. When you manipulate the data, the tools you use update the header to record your changes. Figure 3-12 shows an example header.

The section on the Interpolation Tool explains how to merge files with different time steps and/or start and end times.

Some high-level tools manipulate data in the secondary array. See appropriate sections for help on the Filter Tool, Limit Tool, Trim Tool, and Units-Conversion Tool.

[5] We have found that Microsoft Excel for Windows interprets these labels as if they are dates. If we come up with a workaround for this problem, we will add it to the code.

```
GPP Merge Tool file from "adams.rq"; column 2 converted by "DEG2RAD".
Generated by GPP on 31-May-94 at 12:40:07; 31-May-94 at 13:04:10.
1-Time  1-AZIMUTH   1-BLD_1_RF  1-BLD 1 RE  1-LSS_TOR   1-POWER
```

Figure 3-12. Example of a Merge Tool file header.

See also: Filter Tool, Interpolation Tool, Limit Tool, Trim Tool, Units-Conversion Tool.

The Trim Tool

The Trim Tool eliminates rows from the secondary array. You can specify a block of rows by row number. If the first column has monotonically increasing values (like time), you can also eliminate rows by value. This tool is available only from the Merge Tool Menu after you have merged some data into the secondary array.

See also: Filter Tool, Interpolation Tool, Limit Tool, Merge Tool.

The Units-Conversion Tool

The Units-Conversion Tool is available from the Merge Tool Menu. This tool allows you to rescale the data in selected columns in the secondary array with the same scales and offsets. The Units-Conversion Tool does the scaling using the formula:

Xnew = Scale*(Xold - Offset)

I built many automatic conversions into the Units-Conversion Tool. When asked for the scale and offset, enter a ? to see the list of those available. For example, instead of entering `0.55555,32` to convert from Fahrenheit to Celsius, you can enter F2C. This way, you do not need to look up conversion factors. Figure 3-13 shows the help screen for these conversions.

SYSTEM TOOLS

The system tools allow you to interact with the system resources of you computer and to tell GPP how to behave. They perform essentially no analyses.

The Choose Tool

The Choose Tool allows you to choose a file to read into the primary array. The tool prompts you for the name of the file and will read it into memory. In the default version of GPP, you cannot read files with more than 20 header lines, 20 columns, or 20,000 rows of data[6]. The current version of the Choose Tool can read only ASCII files. I may add the ability to read binary files in the future. After reading the

[6] You can change these limits by modifying the MAX_HEAD, MAX_COLS, MAX_COL1 (MAX_COLS-1), MAX_ROWS, and MAX_ROW2 (half MAX_ROWS) parameter statements in the data.inc include file. You must then recompile GPP to put the modifications into effect.

```
Units conversion is done with the following formula:

     Xnew = Scale*( Xold - Offset )

For a shortcut, some conversions have been set up for you.
Enter the mnemonic for the old units, a "2", then the
mnemonic for the new units.  Here's a list of conversions:

 C   - degrees Celsius     KG  - kilograms          MPS - meters/second
 DEG - degrees             KPA - kiloPascals        N   - newtons
 DPS - degrees/second      LBF - pounds force       NM  - newton-meters
 F   - degrees Fahrenheit  LBM - pounds mass        PSI - pounds/inch^2
 FP  - foot-pounds         M   - meters             RAD - radians
 FPS - feet/second         MB  - millibars          RPM - revolutions/minute
 FT  - feet                MPH - miles/hour         RPS - radians/second

For example, if you enter "DEG2RAD", GPP will convert all
selected columns from degrees to radians.

Enter the Scale and Offset (? for help; 0,0 aborts) [1,0] >
```

Figure 3-13. A help screen for the Units-Conversion Tool.

data, the Choose Tool will compute statistics and create a file of them and give the file an st extension. You can use the Statistics Tool to see the statistics within GPP.

The Choose Tool can automatically detect files created by ADAMS, YawDyn, or the Merge Tool and tailor its behavior accordingly. GPP does this by detecting a specific string at the beginning of the first line. The Options Tool section describes how to change these strings for ADAMS or YawDyn.

When ADAMS generates an out file, it always creates nine columns—even if you do not request any data for all of them. If you label a column as "UNUSED", the Choose Tool will eliminate that column from the primary array. If columns with real data follow an unused column, the tool will shift them to the left.

For generic files, the Choose Tool will ask you for the number of header lines and number of columns of data. If you enter a ? in response to either of these two questions, GPP will display the file with the More Tool to allow you to scan the top of the file.

Many of GPP's tools assume that the azimuth is in column 2 for ADAMS and YawDyn files. If yours is not, you can change the appropriate identifier string with the Options Tool. GPP will then treat the file as generic, and the various tools will ask you for the number of the azimuth column.

GPP will write results from most of its tools to files that have the same root name as the chosen primary file. Those tools will add unique extensions to the root name to identify the tool.

See also: Merge Tool, More Tool, Options Tool, Statistics Tool.

The Directory Tool

When GPP prompts you to enter an input file name, you can enter a ? and the Directory Tool will show you a list of files in the current directory[7].

The Divide Tool

You can divide an ADAMS out file with the Divide Tool. It will create a new file for each ADAMS request in the file. The new file names will have sequential numbers added to their root names and rq extensions (for example, adams.out may generate adams1.rq and adams2.rq). The sequential numbers denote the ADAMS request numbers. You can read these new files using the Choose Tool and manipulate them with other GPP tools.

If the resulting file name is too long for the operating system, the Divide Tool will prompt you for a new name. DOS limits root names to eight characters, and GPP limits UNIX root names to twenty characters. The Divide Tool limits ADAMS out files to nine different requests.

See also: Choose Tool.

The Escape Tool

You can temporarily escape to the operating system with the Escape Tool[7]. This allows you to look around and perform other commands without having to leave GPP. Enter the `exit` command when you want to return to GPP.

If you are a UNIX C shell user, your aliases will not be available to you unless you set them in your .cshrc file. If you have a lot of aliases, setting them all in this file can degrade performance for your other processes. You may want to consider putting your aliases into a separate file (say, .aliasrc) and placing an alias in your .cshrc file to allow you to set them interactively (example: `alias .a 'source ~/.aliasrc'`). You can then just enter `.a` after you start the new command shell to set all your aliases.

The Get-Columns Tool

GPP calls this low-level tool from many other tools. The tool asks you for a list of columns to process. If you enter a naked return, the tool will select most columns automatically. Some high-level tools tell this tool to not include certain columns. For instance, the Azimuth-Average Tool will not let you select the azimuth column for processing, as this would not make sense.

You can also enter a list of column numbers and separate them with spaces, commas, or tabs. The Get-Columns Tool will display a table of selected columns and notify you of their validity. You can enter a naked return to accept the list or enter `N` to reject it. The tool will display a list of numbered column headings to help you choose columns for familiar files. It displays the file header for generic files.

[7] This tool does not work with the Microway NDP FORTRAN compiler when running GPP from a window on a PC. This is not necessarily a big drawback, as you can always do the same thing from a different window. The NDP bug does not manifest itself when running from the normal DOS prompt (without loading Microsoft Windows), so it will not bite you when you really need this tool. If you really need this feature, use a Lahey-compiled version of GPP.

The Help Tool

You can get on-line help for using GPP by entering a ? at a high-level menu. A new menu of help options will appear on the screen, from which you can repeatedly request help on different subjects. If you enter an option letter after the question mark at a main menu, you will get help on that option only and will be returned to that menu.

Examples:

?	sends you to the Help Menu
?A	gives help on the Azimuth-Average Tool
??	gives help on the Help Tool

The More Tool

When you read in a generic file with the Choose Tool, you can enter a ? when asked for the number of rows and/or columns in the file. If you do, the More Tool will display the first seven lines of the file and ask if you want to see more. Just press ↵ to see more of the file or enter a Q to quit the More Tool. I chose seven lines so that files with long lines could wrap the screen without scrolling important information off the top. The More Tool precedes each line with the text Line #>, where # is the number of the line. This helps you determine the length of your header.

The Options Tool

The Options Tool allows you to review and/or change program options. If you do not like the defaults, you can change them for just the current run or for all future GPP sessions. GPP saves permanent changes to the options in a user file. For UNIX users, GPP stores options in .gpprc in your home directory. For DOS users, GPP stores options in GPP.INI in the current directory. DOS users can change the location where GPP.INI is stored by setting an environment variable called HOME to point to another directory. You may want to put this SET command in your AUTOEXEC.BAT file. Here's an example of a DOS SET command:

```
SET HOME=C:\USER\BUHL
```

Options currently available for GPP include strings that identify ADAMS or YawDyn files. See help on choosing files for more information on these strings. Another option is the "dot display rate," which determines how many operations elapse before GPP displays a dot on the screen to let you know that it is processing data. You will want to make this number smaller for slower systems. Other options let you tell GPP to prompt you when you quit to tell GPP how to generate PSDs, or to set the count period for the Rainflow-Cycle-Count Tool. Figure 3-14 is an example of the GPP Options Menu.

See also: Choose Tool, PSD Tool, Rainflow-Cycle-Count Tool.

See also: Merge Tool.

The Write Tool

The Write Tool is available from the Merge Tool Menu. With it, you can write out the results of a merge to a file. The tool prompts you for a file name. You can name your file anything within the limitations of your system (as long it is no longer than 100 characters; paths are allowed). If you specify an

```
Current option settings are:

 A - ADAMS identifier            = "1ADAMS"
 D - Dot display rate            = 2000
 L - Linear detrending for PSDs  = Yes
 P - Prompt before quitting      = No
 R - Rainflow count period (sec) = 1
 T - Type of PSD                 = "Band Smoothing"
 Y   YawDyn identifier           = "     Time   Azimuth"

 Q - Quit changing options

Enter option you wish to change (A,D,L,P,T,Y,=Q) >
```

Figure 3-14. Example Options Menu.

existing file, GPP will ask if you want to overwrite it. If you enter a naked return, GPP will cancel the Write Tool and return you to the Merge Tool Menu.

See also: Merge Tool.

QUITTING GPP

Enter a Q at the menu prompt to quit GPP. If you enter a Q when in the Merge Tool Menu, you will quit the Merge Tool, not GPP. If you want GPP to prompt you before quitting, you can set that option with the Options Tool.

See also: Merge Tool, Options Tool.

References

Downing, S. D.; Socie, D. F. 1982. "Simple Rainflow Counting Algorithms," *International Journal of Fatigue*, Vol. 4, pp. 31–40.

Okamura, H.; Sakai, S.; Susuki, I. 1979. "Cumulative Fatigue Damage Under Random Loads," *Fatigue Engineering, Materials, and Structures, Vol. 1*, pp. 409–419.

Press, W.J.; Flannery, B. P.; Teukolsky, S.A.; Vetterling, W. T. 1990. *Numerical Recipes, The Art of Scientific Computing (FORTRAN Version),* Cambridge University Press, Cambridge, pp. 52–64, 86–89, 515–520.

Glossary

Familiar File	I programmed GPP so that it would recognize often-used files. GPP modifies its behavior to take advantage of its knowledge of the file's structure. The current list of familiar files includes those produced by ADAMS, YawDyn, and the Merge Tool. This is the opposite of a generic file.
File Header	GPP considers the file header to be the text at the beginning of a file. For generic files, you must tell the Choose Tool how many header lines there are in the file. The current configuration of GPP limits headers to twenty lines. You can change the MAX_HEAD parameter in data.inc to modify this limit. *Do not use the header to eliminate unwanted data at the beginning of the file—use the Trim Tool for that.* When GPP asks you for information like a list of desired columns, the Get-Columns Tool will display the entire header to help you figure out column numbers.
Generic File	Generic files are those that are unfamiliar to GPP. It does not understand their structure, so you must tell it how they are organized. They are the opposite of familiar files.
Naked Return	You enter a naked return by pressing ↵ (the Enter key) without pressing any other keys first. You use it to choose a default answer or to ask for the long version of a high-level menu.
Parse	To "parse" is to scan text for meaningful "words." The words can be English-like commands or numbers.
Primary Array	The primary array stores data that you process with most of the high-level tools. GPP generally considers it to be a time series (time in the first column), but it does not need to be for some tools. You use the Choose Tool to fill the primary array with data. You can also copy the contents of the secondary array into the primary array when you quit the Merge Tool Menu. The default dimensions of the primary array are 20,000 rows by 20 columns. You can change these dimensions by modifying the MAX_ROWS, MAX_ROW2 (half MAX_ROWS), MAX_COLS, and MAX_COL1 (MAX_COLS-1) parameters in the data.inc include file.
Root name	The root name of a file is all the text before the last period in the name. DOS limits root names to eight characters and there can be only one period. These limits do not apply to UNIX file names.
Secondary Array	You create the secondary array by copying columns from the primary array with the Merge Tool. The filtering-type tools manipulate the data in the secondary array and do not change its time-series nature. The Limit Tool and Trim Tool can eliminate blocks of data from the array and make time discontinuous. Be sure you do not use the PSD Tool on such data after you have loaded it into the primary array.

Index