

MetCal and the SPA (version 1.0.0)

An extendible meteorological library and graphical driver

Ron McTaggart-Cowan
and the Mesoscale Research Group at McGill University

This manual describes how to use MetCal a graphically-driven meteorological diagnostics package and the expandable Shared Procedure Archive on which it depends (version 1.0.0).

Copyright © 2001, 2003 Free Software Foundation, Inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with the Front-Cover texts being “A GNU Manual”, and with the Back-Cover Texts as in (a) below. A copy of the license is included in the section entitled “GNU Free Documentation License” in the Emacs manual.

(a) The FSF’s Back-Cover Text is: “You have freedom to copy and modify this GNU Manual, like GNU software. Copies published by the Free Software Foundation raise funds for GNU development.”

This document is part of a collection distributed under the GNU Free Documentation License. If you want to distribute this document separately from the collection, you can do so by adding a copy of the license to the document, as described in section 6 of the license.

Table of Contents

1	Overview	1
1.1	What is the SPA?	1
1.2	What is MetCal?	1
1.3	What data formats are acceptable?	1
2	The SPA	3
2.1	Purpose of the SPA	3
2.2	Structure of the SPA	3
2.3	Extending the SPA	4
3	Metcal	6
3.1	Purpose of MetCal	6
3.2	File IO Page	6
3.3	Major Modes	6
3.3.1	Standard Outputs	6
3.3.2	Difference Calculation	7
3.3.3	Simple Math Calculation	7
3.3.4	Advection Calculation	7
3.3.5	Averaging	7
3.3.6	Filtering	7
3.3.7	Interpolation	8
3.3.8	Sounding	8
3.3.9	Cyclone Tracking	8
3.4	Standard Outputs Page	9
3.5	Modifying the Interface	9
4	Other Tools	10
4.1	Developer's Interface	10
4.2	PlotFST Plotting Utility	11
4.3	splitFST file management utility	12
	Appendix A Copying Conditions	13
A.1	GNU Free Documentation License	13
A.1.1	ADDENDUM: How to use this License for your documents	19
	Index	20

1 Overview

The Shared Procedure Archive (SPA) was originally developed at McGill University, starting in about 1999. The intention of the archive was to reduce create an easy-to-use diagnostics package for meteorological applications. It began as a set of portable FORTRAN77 functions and subroutines, and has since then grown into an evolving FORTRAN90 library with a Graphical User Interface (GUI, MetCal) and perl scripts which enable the developer's interface.

As noted in the licensing and copying agreement (see [Appendix A \[Copying Conditions\]](#), [page 13](#)) we encourage all users of this software to become developers and to contribute to the evolution of the SPA. Please share the library entries that you develop so that others may benefit from your efforts.

1.1 What is the SPA?

The SPA is a collection of FORTRAN90 modules, subroutines, and functions that create a working environment for standard meteorological calculations. The include data access routines (generally subroutines), setup and formatting routines (again, mostly subroutines), and calculation subprograms (generally generic functions). It is this latter set of routines that will be of primary interest to the user or developer who wishes to access or extend the SPA without introducing a new data format capability (which is, of course, much appreciated).

These subprograms are intended to abstract the user (and even the end-product developer) from the particulars of the data format supplied, and to make interfacing with existing routines as painless as possible. Inevitably, there will be small bugs in some of the calculations which will be caught and corrected over time — the end result will be a package which is as error-free as possible and meets the needs of users in the meteorological community.

1.2 What is MetCal?

MetCal (the “Meteorological Calculator”) is the GUI for the SPA as described in [Chapter 2 \[The SPA\]](#), [page 3](#). It is the interface that most new users of this package should become familiar with before they begin to work with the developer's interface (see [Section 4.1 \[Developers interface\]](#), [page 10](#)). The interface contains a notebook with three tabs, each of which is described in detail, See [Chapter 3 \[Metcal\]](#), [page 6](#). The end-user simply has to define input and output files, along with the data type, and then request outputs from MetCal. The GUI driver, written in perl/Tk (a modular Tcl/Tk addon to basic perl), accesses a pre-compiled FORTRAN90 executable which handles the program flow at runtime. The result is a very simple interface for the user, who needs to know only basic information to access the computational routines of the SPA.

1.3 What data formats are acceptable?

Version 1.0.0 of the archive accepts input data files in the following formats:

RPN Standard File (1989)

RPN Standard File (1998)

GEMPAK Data Format

Recherche en Prévision Numérique (RPN) is a division of the Meteorological Service of Canada on whose in-house data storage format the original inception of the SPA was based. The data access routines stored in the RPN library (`librmn.a`) are backwards compatible between the 1989 and 1998 I/O subprograms, but the file structures themselves are significantly different, and must be distinguished for proper operation of the SPA. Note that the `ARMNLIB` environment variable must be correctly set during the installation of the SPA for the RPN data format access routines to be loaded.

The GEneral Meteorological PAcKage (GEMPAK) is an open-source distribution made available by the University Center for Atmospheric Research. The `GEMLIB` environment variable (usually set by the `Gemenvi` script distributed with the package) must be correctly set during the installation of the SPA for the GEMPAK data format access routines to be loaded.

As much as possible, all of the data to be used as input should be contained in a single input file, unless regridding is required.

2 The SPA

The SPA is a set of FORTRAN90 subprograms stored in a series of source directories beneath the SPA root. All interactions between the driving program and each of the subprograms are strictly controlled through interface blocks stored in modules which are use-associated as needed in the code. This interfacing scheme allows for explicit error checking as well as making the subprogram interactions easily-understandable for the developer.

There are no common blocks or global variables in the SPA, although some common variables are declared within the lower-level modules responsible for file functions. These variables, however, are scoped only within the module and the subprogram by which it is associated. We are very insistent on the explicit passing of values between routines, since the data flow becomes rapidly untraceable once common blocks and global variables dominate in an extendible archive such as the SPA.

The subroutines contained in the SPA are primarily responsible for interactions with the datafiles and configuration scripts. Functions are generally used for calculation of diagnostics (the essence of the SPA) and are mostly generic, accepting either single value or array inputs for each argument, and producing single value or array output, respectively. Interfacing with array-valued functions is advisable, since they run many times faster than a nested single-valued function call.

2.1 Purpose of the SPA

The SPA is intended to provide an easy means by which to create, store, and access meteorological diagnostics routines in a non data-format-specific sense. The layer of abstraction between the user (and the tools developer) and the data file allows for enhanced focus on the extension of the diagnostics database.

The premise for the SPA is that of an extendible community library of meteorological functions. Accordingly, contributions from as many sources as possible are encouraged. The SPA allows meteorological programmers to focus on the development and enhancement of the package, rather than on the development of lower-level interfacing software and basic functions.

2.2 Structure of the SPA

The SPA source is arranged in a set of subdirectories in this distribution. The naming of the source for the routines follows strict guidelines that are required for some of the automated build processes of the SPA. The routines are easily accessible through a series of developer's utilities described in See [Section 4.1 \[Developers interface\]](#), page 10, although most development work will occur in a local directory belonging to the developer (much like an RCS or CVS modification). The source subdirectories of the archive are:

`'subroutines'`

This subdirectory is the repository for most of the generic initialization and setup utilities in the archive. The subroutines contained in this directory are usually accessed near the beginning of an SPA driver or when file I/O is required.

All subroutines in the archive begin with the `sub_` prefix, followed by the name of the subroutine with a `.f90` extension.

‘functions’

The source in this directory primarily handles the calculations of the SPA. Functions are usually referenced either by control subroutines or by the developer in the “calculations” section of the code. All functions in the archive begin with the `func_` prefix, followed by the name of the function with a `.f90` extension.

‘modules’

The modules contained in this directory contain interface blocks for the control routines of the SPA. The interfaces are grouped by type (i.e. “`mod_FST_File_Uutilities.f90`” holds all of the interfaces for RPN Standard File-specific code). All modules in the archive (with one exception noted in the next item) begin with the `mod_` prefix, followed by the name of the module with a `.f90` extension.

‘interfaces’

The interfaces for all of the calculation routines are contained in the interface file `Interface_MRG.f90`. This is the only source in the SPA whose name violates the rules described in the remainder of this table — and believe me I regret that every time I work with the code. A series of modules consisting of interface blocks are contained in the Mesoscale Research Group (MRG) primary interface, with subprogram access for each depends on broad groupings of functionality.

‘std’

The data file format-specific I/O subroutines are held below this subdirectory, with individual folders for each. Users with installed RPN, GEMPAK, and NCAR Graphics libraries (see [Section 3.3.8 \[Sounding\], page 8](#)) will have a fully-populated `std` subdirectory. Most of the files in this set of subdirectories are subroutines, although a few functions are also present for the GEMPAK interface.

‘opt’

This subdirectory has further subdirectories which allow the SPA to handle missing libraries without exceptions. For example, a user who has no need for GEMPAK data file structures, and who has no GEMPAK libraries installed, will see an `opt/gemlib` subdirectory filled with stub routines which warn the user about the library’s unavailability and exit smoothly. For users with the RPN, GEMPAK, and NCAR Graphics libraries installed see [Section 3.3.8 \[Sounding\], page 8](#) will see no `opt` directory in their distribution since all of the necessary routines are held below the `std` subdirectory.

‘templates’

The template files in this subdirectory are intended to ease mundane coding tasks for developers by providing easy access to initiation and control routines in a usable fashion. The `template.f90` program is intended for standard development tasks, while the `template_gui.f90` source allows for GUI driven development.

‘settings’

This directory holds an example set of configurations, again in an effort to ease the development process to the point where everyone will enjoy it and participate in it.

2.3 Extending the SPA

The easily-extendible nature of the SPA is one of the package's unique attributes. The SPA was originally designed by a devoted synoptician, and contained mostly large-scale atmospheric diagnostics. Other users have since requested (and often provided) very useful computational subprograms that have extended the utility of the package immensely. For example, you may want to calculate storm-relative helicity, a variable not currently supported in the SPA. The developer's interface provides a means by which you can build a helicity calculator with only a few read statements in the driver (template) program. Our hope, of course, is that any computational routine that you create (or, any set of interface routines developed by the very brave) will be incorporated into the next release of the SPA.

A series of utilities are provided as the developer's interface as described in [Section 4.1 \[Developers interface\]](#), page 10. Following the steps described with the utilities will make development of the archive an almost enjoyable process.

3 Metcal

The METeorological CALculator (METCAL) is a GUI between the end user and the SPA. It consists of a set of three notebook pages on which the user sets File I/O characteristics, operation mode requirements, and desired outputs. Each of these pages is described in detail in the following sections.

3.1 Purpose of MetCal

MetCal is intended as an abstraction for the end user. As such, it is limited in flexibility compared to the Developer's Interface. However, it is easy to learn, and will be expanded as the SPA continues to grow. For off-the-shelf diagnostics, this utility can be extremely useful.

3.2 File IO Page

The File I/O page (the first page that the user will see following the copyright notice) allows the user to define the input and output files to be processed by MetCal. For both RPN file formats, the user also has the option to view any of the input or output files. The **primary input** file should reference the file containing all of the necessary input variable, if possible. For the “Standard Outputs” setting (see [Section 3.3.1 \[Standard outputs\], page 7](#)), the **secondary input** file can be used for supplemental input; however, all other Major Modes refer only to the **primary input** file. The **output** file defines the data file into which to store the output from MetCal. If this line is left blank, then the output is piped directly back to the input file. The **grid** file allows the user to remap fields to a desired output grid. All values read from the **primary input** and **secondary input** files are remapped to the domain contained in the **grid** file before the execution of any calculations. If the **grid** file is left blank, then no remapping of the input fields occurs (the default). The file referenced in the **grid** entry must contain at least one two-dimensional field of the grid onto which the data will be remapped.

All of the data times and levels present in the input file will be operated on by MetCal. For this reason, try as much as possible to isolate the data times that you require so that MetCal does not continue to execute on the entire set.

3.3 Major Modes

The Major Modes notepad of MetCal (selected using the tab at the top of the page) controls the program's data and control flows during execution. As described in the following sections, each Major Mode has a distinct functionality that may be of more or less use to different users. The default Major Mode is for Standard Outputs. Selection of any other modes produces a pop-up window for input from the user. All of the entries in the window must be completed before MetCal will execute.

3.3.1 Standard Outputs

The Standard Outputs mode of MetCal provides access to many of the calculation functions of the SPA. The Standard Outputs page allows users to select the calculations to be performed. Each calculation represents a routine (usually a function) contained in the archive. See [Section 3.4 \[Standard page\], page 9](#), for details.

3.3.2 Difference Calculation

This simple field differencing algorithm can be used to compute the difference between a single variable at two different levels, or between two different variables at the same level. Of course, you could compute the difference between different variables at different levels, but you would really have to ask yourself why you are spending time doing that. The input and output identifiers must be the names of the variables contained in the `primary input` file on the page, See [Section 3.2 \[File io page\], page 6](#).

3.3.3 Simple Math Calculation

This mode allows the user to perform simple (+, -, *, /) operations on the selected two or three dimensional field. The quantity in the `value` box may be either an integer or real number, but the computation takes place with a floating point number. This mode is especially useful for copying fields (multiply the desired field by 1) and for changing units.

3.3.4 Advection Calculation

The Advection Calculation mode provides access to the advection routine of the SPA. Any level of any field (under the `Field Information` heading) may be advected by the horizontal wind at any level. The vertical advection calculation is not implemented in version 1.0.0 of this distribution. Although the user may select from either the `primary` or `secondary input` data files by pressing the `Input Variables` button, this usage is discouraged and will likely be disabled in future releases, thus allowing only `primary input` file entries.

3.3.5 Averaging

The field averaging algorithm accessed by this Major Mode computes the average (either arithmetic or pressure-weighted) of a variable in a column whose upper and lower bounds are defined by the `Top Level` and `Bottom Level` entries. The input data for this mode must be given in the `primary input` file.

3.3.6 Filtering

The Filtering mode of MetCal runs a 25-point smoothing box over the entirety of the domain for the two or three dimensional variable specified in the `Identifier` entry. The shape and weights of the cells in the filter are shown graphically, and may be modified by the user to smooth more heavily or lightly, as required. There is no “magic number” up to

which the cell-totals must add, so feel free to modify the filter to fit your requirements. One of the drawbacks of the MetCal interface distributed with version 1.0.0 of this distribution is that the output identifier is set equal to the `Identifier`. This oversight will be corrected in future versions of the interface.

3.3.7 Interpolation

With horizontal interpolation handled automatically through the `grid` file, the Interpolation Major Mode is responsible for implementation of cubic-spline interpolation in the vertical. The pop-up window defaults to the mandatory pressure levels, but this is just an example of how to specify multiple level outputs. As many levels as required may be specified in a comma-separated list.

3.3.8 Sounding

The Sounding mode of MetCal requires NCAR Graphics libraries for plotting. If these libraries were not installed when the SPA was built on your machine, then you will not have access to the sounder. However, if the NCAR Graphics libraries were (and still are) available to MetCal, then the only entries required in the pop-up window are the x,y coordinates of the gridpoint at which to take the sounding (it will be taken at this point for every time in your input file, in keeping with MetCal tradition). The `Plot Title` entry is one of the few optional entries in the Major Modes. The soundings produced by this algorithm are in “metafiles” produced by NCAR Graphics, and can be viewed with the `idt` utility distributed with the NCAR Graphics package, accessible through the `view output file` button on the File IO page (see [Section 3.2 \[File io page\], page 6](#)).

3.3.9 Cyclone Tracking

The Cyclone Tracking Major Mode of MetCal is the most complex in terms of usage. Any two dimensional variable (in the `Identifier` field) can be operated on in the region of a tracked cyclone. Note that all cyclones are tracked using the height field (on pressure levels, default 1000~hPa in the `Level` entry) and that it must therefore be present in the `primary input` file, along with the field of interest. The `Maximum Cyclone Phase Speed` entry allows the user to track cyclones of different speeds. You will probably only run into a problem if this is set too low, but the default value of 30 m/s is already pretty high. The `Radius of Interest` defines the low-centre-relative region over which to operate on the field of interest. The default value for this entry is 1000 km.

The Cyclone Tracking Major Mode produces text output files of any or all of the field of interest’s mean, standard deviation, minimum, or maximum near the low centre. These files are named with the `Text Output Root` prefix and `-mean`, `-sd`, `-min`, or `-max` suffixes. The output format is space-separated “`time value`” for forecast data, or “`date value`” for analysis data. These text files can be ingested directly into a plotting program such as `xmgrace` or (shudder) `Excel` to produce time-series of storm-following quantities.

Two execution modes are selectable by the user. The `automatic` start mode searches for the lowest height on the pressure level (i.e. the deepest low) in the initial field, and continues to track that feature. `Manual` start mode allows the user to input the x,y grid

coordinates of the cyclone to be tracked. Similarly, the `automatic` step mode allows the algorithm to track the cyclone automatically (or at least to attempt to). If the routine senses that it has lost the track, it will prompt the user for the x,y coordinate at a specific time in MetCal's parent shell. The `manual` step mode always asks the user for the location of the storm at every step.

Because of the unique interactive nature of this Major Mode, MetCal must be run in the foreground of the parent shell for correct functionality. We hope to rectify this problem with an interactive TopLevel window in future versions of the interface.

3.4 Standard Outputs Page

The `Output Fields` page is specific to the Standard Outputs Major Mode. Some of the more commonly used routines of the SPA are accessed by a series of check buttons. The `Input Variables` button allows the user to describe characteristics of the input fields contained in either the `primary input` or the `secondary input` data files. MetCal can, in general, accurately predict the fields that it requires to make the requested calculations. You should always check the input variables before execution. The `Check Variables` button allows the user to determine the output characteristics of the computed fields (essentially just the output variable identifier). Once you know these by heart for your usual file type, you probably will not need to use this button again. This page represents the beginnings of MetCal, and provides the most powerful functionality for the program.

3.5 Modifying the Interface

The GUI can be modified and updated by a developer who is familiar with perl/Tk. Access is provided via flags to the Developer's Interface (see [Section 4.1 \[Developers interface\]](#), [page 10](#)). In general, enhancements and upgrades should be distributed to the community and incorporated into the next release rather than being implemented in an ad-hoc fashion; however, if the development is just that good, then you have the flexibility to incorporate it between releases.

4 Other Tools

Several other tools are included with this distribution. Some of them relate exclusively to the RPN data file format, for which there is relatively little open-source software available. However, the Developer's Interface is central to the archive itself. Most of the tools distributed with version 1.0.0 of the SPA are described in the following sections.

4.1 Developer's Interface

The Developer's Interface is second only to MetCal as an avenue by which to access the SPA. This interface makes the job of developing routines almost enjoyable by relieving the developer from the tedious job of building a driver program. The basic steps involved in beginning a new development are:

1. Open a working directory and load template files and configurations;
2. Extract associated subprograms from the archive for verification of their functionality;
3. Determine dependencies and create a build environment (Makefile);
4. Write a new routine and interface through the template file; and,
5. Compile and test your new code.

Luckily, the SPA comes with a set of tools that make this process much simpler than it sounds. Each of these tools is described (in the standard order of their application) here. Each of these utilities is also supported by man pages. Please browse these pages for information not contained in this manual.

- '**spaopen**' This utility completes Step 1 above, by creating a working directory and pre-loading it with the desired template and configuration files. **spaopen** accepts an optional flag and a single argument. The argument is the root name of the subdirectory, to which will be appended `_f90`. This is a hold-over from the FORTRAN77 days of the archive, but serves to emphasize the development directory nicely, so it stays. The recognized flag is `-gui` which loads the GUI driven template, and a copy of the MetCal source code into the working directory. Running **spaopen** without a flag (but still with the directory name) will result in a directory containing the standard driver template, and is the common usage of the utility. Most
- '**spalist**' This utility allows the developer to look at a listing of the source code available in the SPA. The options for this tools are **subroutines**, **functions**, **interfaces**, and/or **modules** (**templates** may also work, but is of little practical use). These options of course refer to the subroutines of the archive. See [Section 2.2 \[Structure of the spa\], page 3](#), for a complete description of the archive's structure. The ability to list the existing source is useful in two ways. Firstly, it allows you to ensure that there is no repetition of subprogram names in your development, and secondly it provides information on the algorithms already coded in the SPA but not explicitly referenced (inasmuch as the subprogram names provide useful information about the functionality of the source).
- '**spainfo**' This tool allows the developer to obtain detailed information about the functionality and interface of the specific routines listed as command line arguments.

Note that the full name of the source file (including the prefix and suffix) must be included for `spainfo` to be able to find the appropriate source. A file header, contained in the source, is printed to `STDOUT` and contains package, interface, version, and functionality information. There are two primary uses for `spainfo`. The first is to scan the archive for pre-existing routines which may be of interest to you. The second is in interfacing to pre-existing routines, since a full list of the arguments and their properties are displayed in the header. Note that `spainfo` produces a warning message if it finds a generic subprogram.

'spaextract'

Moving onto Step 2 above, the `spaextract` utility allows the user to obtain source from the correct archive subdirectory. Any number of routines can be obtained simultaneously. Again, the full file name must be provided so that `spaextract` knows where to look in the archive for the required source.

'comp90.d'

Creating the build environment for your development project has never been easier. The `comp90.d` developer's interface tool allows the developer to painlessly develop a `Makefile` with a full dependency list and extract dependent files for recompilation in seconds (depending on the number of files to retrieve, of course). The only flag recognized by `comp90.d` is the `-opt` option, which compiles code in an optimized form. The compilation step can take quite a bit longer in this case, but optimization will yield returns at run time. `comp90.d` should be run any time that there is a modification to the source file listing of the directory so that all of its dependency lists are updated. Note that any SPA files that are automatically extracted by `comp90.d` are distribution cleanable. You should therefore run `make distclean` before re-running `comp90.d` in your working directory to clean out all of the automatically-retrieved code. Once the `Makefile` is built, it has the functionality of a standard GNU `makefile`.

The `comp90.d` utility requires some help in determining dependencies for function calls. If the source that you develop makes use of the functions `foo` and `foo2`, then your code should contain a line like:

```
! external :: foo,foo2
```

Although this is a comment line for the FORTRAN90 compiler, it will be correctly interpreted for the source's dependency list by `comp90.d`. The other anomaly associated with the `comp90.d` utility is the requirement of library loading, again displayed near the top of the template file:

```
! load FST      <- force comp90.d loading of ARMLIB \
                                     (FST library)
! load GEMPAK   <- force comp90.d loading of GEMLIB \
                                     (GEMPAK library)
! load NCARG    <- force comp90.d loading of NCARG \
                                     (NCAR Graphics library)
```

This will ensure that the proper libraries are loaded by the linker to create the executable. (Of course, only the 'load XXXX' is required - the comments are optional). Note that `NCARG` loading is not standard in the template file.

4.2 PlotFST Plotting Utility

The `plotFST` utility is a GUI which accesses the `sigma` plotting language developed by RPN. Clearly, this tool will only be useful for those using the RPN standard file data format. The interface is nearly entirely self-explanatory, and can be reformatted to either menus (default) or columns by selection in the `Display` menu. Both formats are functionally identical.

Up to three variables can be plotted simultaneously (one for each “set”). The sets are activated by pressing the button at the top of the set’s column. The initialized values are fairly common, so often there is not a lot of modification required before a usable product is obtained. Users familiar with the RPN standard file format will be used to dealing with the IP3 identifier, which can be used to specify fields beyond simple time and level stamps. The IP3 stamp is particularly useful for `plotFST` when running the utility on analysis data (in conjunction with `splitFST`).

The `Settings` menu allows the user to set map options and to create axis labels. However, it also aids in the quick production of plots. The `Locking` options refer to the propagation of time, level, and IP3 settings across sets. For example, if all three locks are engaged (as are, say, Sets 1 and 2) and the bar at the top of Set 1 is pressed, then the time, level, and IP3 settings of Set 1 are propagated to Set 2. If the hour lock is disengaged, then only the level and IP3 settings are propagated. This makes leveling and time changes much faster and practically bombproof.

Also under the `Settings` menu, the user is given the `Time Series` options of `Multiple Times` or `Multiple Files`. The time series output is engaged by entering a colon-separated list in the `Hour` entry box for the sets (i.e. `0:48:6`) of the form `start:end:step` in hours. This functionality is also available for the IP3 entry. The `Multiple Times` option produces a single metafile (the output default output from `sigma`) with multiple times in it; the `Multiple Files` creates a separate file for each time, especially useful for GIF generation.

4.3 splitFST file management utility

The `splitFST` file formatting utility is usable only for RPN Standard data files. A complete man page for the tool is available online and contains a more detailed description of the capabilities of the tool than that presented here. The most useful application of the `splitFST` utility is in the plotting of analysis files using `plotFST`. The command `splitFST -dates` will provide each individual analysis time in a combined analysis file with a unique IP3 identifier. This allows `plotFST` to use its IP3 entry to isolate analysis times within the file. A full listing of the options accepted by `plotFST` can be obtained with the `-h` option.

Appendix A Copying Conditions

A.1 GNU Free Documentation License

Version 1.2, November 2002

Copyright © 2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related

matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at

your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.1.1 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C) year your name.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover  
Texts. A copy of the license is included in the section entitled  
‘‘GNU Free Documentation License’’.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with  
the Front-Cover Texts being list, and with the Back-Cover Texts  
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index

(Index is nonexistent)