# MAPLAB Introduction and User's Manual

April 18, 2002 Version 1.49 1a

M. Potse and A.C. Linnenbank Medical Physics Department Academic Medical Center University of Amsterdam  $\mathbf{2}$ 

## Foreword

MA<sup>P</sup><sub>L</sub>i<sup>B</sup> is a Multichannel ECG toolbox for MATLAB. MA<sup>P</sup><sub>L</sub>AB is a Graphical User Interface written on top of it. The MA<sup>P</sup><sub>L</sub>i<sup>B</sup> toolbox by itself is useful for writing batch jobs and extensions to MA<sup>P</sup><sub>L</sub>AB. The two packages are described in a series of four manuals:

#### MapLab Introduction and User's Manual:

MapLib Reference Manual: Manuals for all MAPLiB routines

MapLib Sourcebook: Documented sources for the MAPLiB toolbox

#### MapLab Sourcebook: Documented sources for $MA_{L}^{P}AB$

These books are available via the world wide web from the server gnu.amc.uva.nl. Currently however, this computer is not accessible from outside the amc.uva.nl domain, because of a firewall. The books are updated about weekly, sometimes daily.

The MAPLAB package is changing rapidly. It gains new functionality and new bugs almost every week, and bug fixes are made almost daily. New versions can be requested at any time; a list of changes appears in chapter 6 of the user manual. It is recommended that users check this list regularly to see if an update of their software is worthwile.

This set of manuals is under construction. Some small parts of it are written in Dutch, other parts in English. Gradually, it will be translated into English. The User's Manual is now translated completely.

M. Potse

 $\mathbf{4}$ 

## Contents

1	$\mathbf{Intr}$	roduction 7			
	1.1	License			
	1.2	Warnings			
	1.3	System Requirements			
		1.3.1 Software requirements			
	1.4	Installation			
	1.5	About this Manual			
	1.6	Prerequisites			
	1.7	Problems			
<b>2</b>	Usa	ge 13			
	2.1	The "MapLab"-window 13			
		2.1.1 Menus			
		2.1.2 Buttons and such			
		2.1.3 Options			
	2.2	Loading Datafiles			
		2.2.1 Recordings			
		2.2.2 Very Large Recordings			
	2.3	The "Recording"-window 16			
		2.3.1 extended selections $\dots \dots \dots$			
		2.3.2 buttons and keys $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $17$			
		2.3.3 Menus			
	2.4	The "Channel"-window 18			
		2.4.1 Buttons			
		2.4.2 Markers			
		2.4.3 Zooming			
		2.4.4 Slaves			
		2.4.5 Multiple leads $\ldots \ldots 23$			
		2.4.6 Automatic QRS Detection			
		2.4.7 Automatic Activation Detection			
		2.4.8 Fourier Transforms			
		2.4.9 Signal Alignment			
		2.4.10 Signal Averaging			
		2.4.11 Other Transforms			
		2.4.12 Filters			
		2.4.13 Menus			
	2.5	Saving Things			
		2.5.1 $MA_{L}^{P}AB$ metafiles			
		2.5.2 Saving Data			

	2.6	The (standard) ECG	29
		2.6.1 Menus	30
	2.7	Printing Body Surface (Integral) Maps	31
		2.7.1 Menus	31
	2.8	Interactive Body Surface Maps	31
	2.9	Localization of Ectopic Beats	33
		2.9.1 The Localize Windows	33
	2.10	Cross-correlation of Maps	34
	2.11	Activation Maps	34
	2.11	2 11 1 Isochrones	36
		2.11.2 Activation Tracking	36
	9 1 9	Histograms	36
	2.12 9.12	The Fine Drint	36
	2.13	$\begin{array}{c} 1110 \text{ Fine Fine Fine } \\ 2 12 1 \text{ clossing} \end{array}$	30 27
		2.15.1 glossies	37 97
		2.13.2 SIZE	37
		2.13.3 greyscale and color graphics	37
9	Ont	iona	<b>9</b> 0
ა	0pt	Ontions for MAPAD	<b>30</b> 20
	3.1 0.0		38
	3.2	Options for the "Recording"-Window	39
	3.3	Options for the "Channel"-Window	39
	3.4	find $dV/dt$ options	40
	3.5	Options for the Standard ECG	41
	3.6	Options for Body Surface Maps	42
	3.7	Options for the "Fast Map"-window	44
	3.8	Options for the "Fast IntMap"-window	44
	3.9	Options for Activation Maps	44
	3.10	Datafile (MCHD) options	47
		3.10.1 MCHD options	47
		3.10.2 CONF options	47
		3.10.3 PDAT options	47
		3.10.4 RDAT options	47
<b>4</b>	Cus	tomization and Hacking	<b>49</b>
	4.1	variable naming conventions	49
	4.2	Global variables in $MA_{L}^{P}AB$	49
	4.3	Grid definitions	51
		4.3.1 Organization of grid definition files	51
		4.3.2 Grid definition file format	51
	4.4	Hook files	52
	4.5	Reading and changing the sources	52
	1.0		
5	$\mathbf{A} \mathbf{S}$	hort History of $\mathrm{MA}_{\mathrm{L}}^{\mathrm{P}}\mathrm{AB}$	55
6	Cha	nges to $MA^{P}_{L}iB$ and $MA^{P}_{L}AB$	57
Bi	bliog	raphy	72
In	$\mathbf{dex}$		73

## Chapter 1

## Introduction

Because life is short and the arts are long it would seem inadvisable either to require engineers and physicists to accumulate decades of biologic experience or to wrench physiologists and physicians away from their posts of observation for indoctrination of questionable applicability.

-Leo G. Horan and Nancy C. Flowers, [5]

MAP<sub>L</sub>AB is an experimental software package for analysis of multichannel ECG recordings. It has several aims, and we think it is instructive to express a few of them here.

- 1. In the first place, MAPLAB is a tool for offline analysis of multichannel ECGs, typically consisting of QRS detection or activation detection and subsequent creation of maps or isochrone charts, and maybe application of more sophisticated and experimental methods.
- 2. Some MAPLAB routines are able to produce professional quality graphics, such as body surface maps. Creating illustrations for publications has therefore become an important application.
- 3. MAP<sub>L</sub>AB is developed and used in an experimental environment. Because it is implemented in MATLAB, it is extensible, and it can easily be changed by users and developers. This makes MAP<sub>L</sub>AB an ideal environment for development of new algorithms for the analysis of ECGs. The functions that MATLAB offers for the creation of user interfaces make it possible to add a workable user interface to experimental routines with a reasonable amount of effort, so that these can be applied to biomedical and clinical studies. This may be the most important application of MAP<sub>L</sub>AB.
- 4. The package is currently under development to suit it for on-line usage in the catheterization laboratory.

MAPLAB is theoretically portable to every platform where MATLAB runs. Indeed, it is now used on PCs running Linux and several variants of MS-Windows, and Apple Macintosh systems. It has also been used on Sun 4 and Solaris systems. However, some bottleneck computations are written in the C language and dynamically linked with MATLAB, using MATLAB's "mex-file" mechanism; these functions must be compiled for each platform separately. This should be possible on every platform if a C compiler is present that is supported by MATLAB. Compiled files for MS-Windows are integrated in the distribution, so most users will not need to compile anything themselves. The screen previews shown in this document were created on a PC running the Linux operating system, and X Windows with the FVWM window manager. The windows look slightly different on other platforms.

 $MA_{L}^{P}AB$  is developed mainly on a Linux system, and to a lesser extent on a PC running MS-Windows. The source code is written and thoroughly documented in the CWEB and MWEB languages [8, 19]. The documentation, which includes the full source code, is freely available.

## 1.1 License

The MAP<sub>L</sub>AB package and the MAP<sub>L</sub>iB toolbox are copyright software. If you have obtained permission to use MAP<sub>L</sub>AB, you may make copies for personal use, compile it for personal use, and modify it for personal use. You may not redistribute original or modified versions of MAP<sub>L</sub>AB or MAP<sub>L</sub>iB, or advertise a modified version as "MAP<sub>L</sub>AB" or "MAP<sub>L</sub>iB." Requests for modifications or copies may be addressed to the authors.

## 1.2 Warnings

The authors do not claim that any aspect of the  $MA_{L}^{P}AB$  package, the  $MA_{L}^{P}B$  toolbox, or this manual, is correct or useful for any purpose. Anything may be changed without prior notice.

The authors disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness, in no event shall the authors be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

The MAP<sub>L</sub>AB software writes usage statistics and error messages in a log file. At startup, the software attempts to send this file to the authors. This may be undesirable, e.g. if your computer is connected to the internet via a telephone line. If you don't like this behaviour or suspect that it causes problems, please contact the authors for instructions on how to remove the automatic submission code. Complete documentation on this mechanism can be found in bs\_log.web.

## **1.3** System Requirements

Although MATLAB and the MAPLAB package will run on a very modest PC, better hardware can significantly improve performance. A minimum, recommended, and high-end setup are shown in table 1.1. The **required** setup works, but slowly; not suitable for daily use. The **recommended** setup works very well for common tasks, perhaps including on-line usage. The **high-end** setup is an indication of what we—today—consider really fancy equipment, perhaps necessary for tasks quite a few times larger than ours.

16-bits colour is very important for quickly drawing windows. Additional memory up to 32 megabytes is more important to speed than the CPU type and clock frequency. However, larger amounts of memory are only relevant if large files must be processed (larger than, e.g., 3 MB uncompressed or 1 MB compressed). MATLAB requires 40 MB of disk space to install; 100 to 300 MB for a full installation. The  $MAP_LAB$  package requires

#### 1.4. Installation

resource	required	recommended	high-end
memory	$16\mathrm{MB}$	$64\mathrm{MB}$	$256\mathrm{MB}$
cpu (PC)	80486	Pentium II 233	Pentium IV $2\mathrm{GHz}$
screen resolution	$800 \times 600$	$1152\times864$	$1600 \times 1200$
∫screen depth	8 bits	16 bits	24 bits
nr. of colours	256	65536	(truecolor)
fixed disk	$80\mathrm{MB}$	$1.6\mathrm{GB}$	$60\mathrm{GB}$
printer	see section	2.13	

Table 1.1: Hardware requirements

less than 2 MB of disk space. Many more space may be necessary to store your data. A full EP session with the 128-channel 2 kHz system generates 6 GB of data. Today, two or three such sessions would fit on a computer's fixed disk.

#### **1.3.1** Software requirements

MAPLAB requires MATLAB version 5.1 or later. MATLAB is distributed for several architectures and operating systems, among them:

- PC with MS-Windows '95/'98
- PC with MS-Windows NT
- PC with Linux version 2.0.18 or later, and X Windows (X11R6)
- several workstations, such as Sun Sparc, HP 9000, DEC Alpha, ...

MATLAB versions prior to 5.3 are known to have problems on MS-Windows '98 that make working with  $MA_{L}^{P}AB$  in this situation impossible. For more information please refer to MATLAB's own documentation.

## 1.4 Installation

Installation instructions can be found in the file README, which comes with the  $MA_{L}^{P}AB$  distribution. For convenience, its contents are copied here:

This file contains installation instructions for MapLab.

see also the file "README"

#### INSTALLATION

The purpose of the installation procedure is to copy the Matlab programs, mex files, and data files that are stored in the distribution file to a place where Matlab can find them. On platforms other than Linux, Windows, and Macintosh it is also necessary to compile the mex files. This is treated below.

The distribution archive file is called maplab-x.y.zv.zip, where x is the major version number, y the minor version number, and z an additional version number. v is a single letter which can be either 'l', 'm', or 's', for Large, Medium-size, and Small variants of the distribution, respectively.

The first step in the installation process is to unpack the distribution archive file, in such a way that its directory structure is retained. This can be done e.g. with WinZip (windows), pkunzip (dos/windows), or unzip (unix).

If you already have a version of MapLab installed, it is highly recommended that you make a copy of the existing MapLab directories. This copy can be used if you don't like the new version, or if you need to retain modified versions of distribution files.

The usual method of installation on windows systems is to unpack the archive to a new directory c:\maplab, and then to add the directories c:\maplab\dos, c:\maplab\matlab, and c:\maplab\data, in this order, to Matlab's search path. It is critically important that Matlab searches the "dos" subdirectory before the "matlab" subdirectory! See also the note on "elconf" directories below! You can install MapLab in any place you like, provided that Matlab will search the tree directories in the correct order.

On systems other than windows, the subdirectory "mex" takes the place of the "dos" subdirectory. The "dos" subdirectory contains compiled Matlab MEX files, which have a .dll extension on the windows platform. The extension for MEX files differs between platforms. MEX files must be compiled specifically for each platform. This can be done by executing one of the compile scripts found in the "mex" subdirectory, or doing something equivalent. A method that should work on all platforms (provided that Matlab and a supported C compiler are installed and configured) is to start Matlab, cd to the mex directory, and execute compile.m. After compilation, you must ensure that Matlab will find the compiled MEX files. This can be done by adding the "mex" subdirectory to Matlab's search path. Matlab must search this directory before it searches the "matlab" subdirectory!

For Linux and Macintosh systems, compiled MEX files are provided on demand.

The last step in the installation procedure is to configure MapLab:

- \* edit the file maplabrc.m as needed.
- \* If you have any database files (such as "NLV" and "RAA"), move them in a place where Matlab can find them, for example the "data" subdirectory.
- \* Most probably, you will need to use Grid definition files with MapLab. These files, which provide MapLab with information on electrode grid geometry, should reside in a directory called "ELCONF" that is on Matlab's path. On ms-windows systems, this is usually c:\maplab\elconf. A large set of grid definition files is included in the "ELCONF" directory in the distribution. For a first installation of MapLab, you may simply add this directory to Matlab's search path.

It is possible to have several "ELCONF" directories, which will all be used by MapLab, provided that they are mentioned individually in Matlab's path.

In addition, each "ELCONF" directory may have subdirectories. The names of subdirectories are used in MapLab to generate a menu structure for grid selection: for each subdirectory a submenu will

be created, in which all grids appear that were found in this subdirectory.

If you get the error message "unknown grid: ..." when loading a datafile, it means that MapLab could not find a grid definition file that matches the specification given in the datafile. This may be due to incorrect installation of the ELCONF directory. Alternatively, it may mean that the grid specification in the datafile is wrong, or that a matching grid definition file is still to be created.

#### BACKGROUND

MapLab consists of a set of m-files and a set of mex files (see the Matlab documentation for an explanation of these terms). The m-files are (or at least should be) platform-independent. The mex files are C programs that must be compiled for each platform. MapLab is distributed with compiled versions for DOS/Windows. Compiled versions for Linux and Solaris version 2 may be provided on request. We cannot compile them for other platforms, but if you use Matlab on an other platform, you should be able to if you possess a C compiler that can cooperate with Matlab's (c)mex script. Consult Matlab's "External Interface Guide" or "Application Program Interface Guide" to find out how to do this on your platform.

## 1.5 About this Manual

In this manual, several fonts are used to identify special things:

- *italics* are used for names of MATLAB variables
- typewriter type is used for filenames
- *slanted type* is used for emphasis
- **bold type** is used for key words

Keyboard keys, when they have a special meaning if pressed while some specific window has the focus, are indicated with a box like this: A. The capital letter A, usually typed using the "shift" key, is indicated as <u>SHIFT-A</u>. Likewise, we have <u>CTRL-A</u> and <u>ALT-A</u>.

Menu names are printed in sans serif type, like this: Actions|map. The vertical bar separates the submenu "map" from the menu "Actions" in this example. There can be multiple bars in a menu name, if a subsubmenu or subsubsubmenu is indicated.

Please note that this manual has an Index, which is meant to be correct and helpful. Important topics, window names, and variable names (in italics) can be found in this index.

#### **1.6** Prerequisites

We assume that the reader of this manual has some basic knowledge of MATLAB, such as can be obtained by reading the booklet "Getting Started with MATLAB" [12]. It is very well possible to read this manual and use MA<sup>P</sup><sub>L</sub>AB without this knowledge, but it may be annoying to find certain paragraphs hard to understand just because the lack of it. In order to change parts of  $MA_L^PAB$ , to add new routines, and to adapt  $MA_L^PAB$  to the specific requirements of a department, a moderate knowledge of MATLAB is absolutely necessary. Those who wish to undertake such tasks we strongly recommend to read at least "Using MATLAB" [13].

## 1.7 Problems

If you have a problem in installing or using MAPLAB, we suggest that you ask help from your colleagues before asking us. Bug reports may be sent to maplabbug%gnu@amc.uva.nl; suggestions and other discussion to maplab%gnu@amc.uva.nl or to the authors themselves.

## Chapter 2

## Usage

Cahn's Axiom: When all else fails, read the instructions.

To start working with  $MA_{L}^{P}AB$ , first startup the MATLAB program. This brings up a window where commands can be typed and results are printed. This window will be referred to as the "Matlab"-window. Type "maplab" in this window (followed by  $\leftarrow ENTER$ ), of course). This will in turn bring up the "MapLab"-window (see figure 2.1).

It is also possible to add maplab to the matlabrc.m file [13], to make it start up automatically when MATLAB starts.

## 2.1 The "MapLab"-window

The  $MAP_LAB$  package consists of many windows containing buttons, menus, etc. which control its actions. These windows can be created by pushing a button or selecting a menu in another window. There is a tree of windows, and the "MapLab"-window is at the root of this tree.

#### 2.1.1 Menus

The MAP<sub>L</sub>AB package is controlled by buttons and *menus*. An overview of the menus in the "MapLab"-window is given in table 2.1. The purpose of most of the menus is explained in later sections.

Menus are found in most MAPLAB windows. Generally, they affect only their own window and its hierarchy. Because the "MapLab"-window is at the top of the hierarchy, its menus have a global functionality.

#### 2.1.2 Buttons and such

 $MA_{L}^{P}AB$  is controlled with menus, buttons, and other such controls. The use of these may nowadays be assumed to be straightforward. The buttons in the "MapLab"-window are explained in sections 2.2 en 2.5.

#### 2.1.3 Options

 $MA_L^PAB$ 's behaviour is influenced by numerous options. Their values can be changed in options windows, which are accessed by selecting an Options menu in one of  $MA_L^PAB$ 's windows. All options windows in the  $MA_L^PAB$  package have the same general layout: for every option there is an edit box, checkbox, or something alike, and also a "reset" and a



Figure 2.1: A part of the computer screen, featuring the "MapLab"-window and, partially hidden, the "Matlab"-window.

"default" button. These buttons serve to retrieve the value of the option when the options window was created, and the default value, respectively. At the bottom of the window are 3 buttons: "OK" applies the (changed) values and closes the options window, "Apply" only applies them, and "Cancel" closes the window, discarding the changed values. A description of all options in the package is given in Chapter 3.

Many details of MAPLAB's behaviour are only described in Chapter 3. The descriptions in the current chapter are kept concise, and are limited to the most basic usage. We recommend reading Chapter 3 regularly when you have got used to MAPLAB, because there are many more options than one can remember, and the list of options grows continually.

The options can be saved with the Options|Save Options menu in the "MapLab"window. Values are stored in a file called bs\_opt.mat, somewhere along MATLAB's search path. You can keep a private options file if you use MATLAB in such a way that you have a private directory on top of your path [13], and store a bs\_opt.mat file in that directory. This is quite common on UNIX platforms, but not on MS-Windows. You can arrange for this by copying the standard (or any other) bs\_opt.mat into this directory, typing path at MATLAB's command prompt to force MATLAB to reset its file cache, and activating the menu Options|Save Options. The options file is loaded when MAPLAB starts.

The sizes and positions of some windows (like the "Recording"-window) are also considered options, and are saved with the Options|Save Options menu.

#### **Default values**

All options have a default value. In some cases this value depends on the data, or on the value of controls. When an options window is opened for the first time or a "default"

Menu	item	functionality	§
Actions	recording	open "Recording"-window	2.3
	channel	open "Channel"-window	2.4
	stdecg	create printable ECG	2.6
	map	create potential map	2.7
	intmap	create integral map	2.7
Options	Maplab	show global options	2.1
	std ecg options	options for standard ECGs	2.6
	map options	options for maps	2.7
	datafile options	options for MCHD files	3.10
	Save Options	save options in a file	2.1.3
About		$MA_{L}^{P}AB$ information	
Quit	quit this window	remove "MapLab"-window	
	maplab windows	remove all $MA_{L}^{P}AB$ windows	
	quit all windows	remove all windows	

**Table 2.1:** Overview of menus in the "MapLab"-window. The last column indicates in which section more information can be found.

button is pressed, this value is recomputed and shown. A changed value will only be used when the "apply" or "OK" button in an options window is pressed.

## 2.2 Loading Datafiles

The "load" button in the "MapLab"-window brings up a window where a datafile can be selected. The user interface is self-explanatory. If you press the "OK" button, or doubleclick on a filename, the window is closed and the "Recording"-window is opened to show the newly loaded file.

The currently loaded file can be reloaded using the "reload" button in the "MapLab"window. This is only necessary if some routine messed up the data. Routines that do this include FFT computation and signal averaging. Generally, the operations found in the Actions|Transforms menu in the "Channel"-window transform the recording irreversibly. Also, the effect of a baseline correction using more than 2 baseline markers cannot be undone. Using the "prev" and "next" buttons you can load the files that come previous/ next in the directory listing, relative to the current file.

MAP<sub>L</sub>AB automatically recognizes several data file formats that are currently in use. These include body surface mapping files recorded with equipment from the Medical Physics department, University of Amsterdam, and recordings made with equipment from the Experimental Cardiology department, University of Amsterdam. These formats are in use at several related centers. If a datafile is not recognized, a dialog window is presented to ask for the user's assistance.

#### 2.2.1 Recordings

Some datafiles contain multiple recordings. The current recording number is displayed in the edit box in the lower left corner of the "MapLab"-window. By editing this value, or pressing the "+" and "-" buttons next to it, you can load an other recording from the current file. The recording is loaded immediately if one of the buttons is pressed, if **ENTER** is pressed in the edit box, or if the focus leaves the window after the edit box was edited.

button	description
<	move to the first part
<<	move back one full range
<	move back half a range
R	reload the current part
>	move forward half a range
>>	move forward one full range
>	move to the last part

 

 Table 2.2: Buttons in the "MapLab"-window that change the currently loaded part of the recording.

#### 2.2.2 Very Large Recordings

A recording may very well be too large to fit in the computer's memory when loaded by MATLAB. It is now not uncommon to create datafiles that are hundreds of megabytes long, and even when the datafile is a few megabytes in size, the fact that MATLAB stores each number in 8 bytes makes that the loaded recording may be too large to handle quickly, or to handle at all.

 $MA_{L}^{P}AB$  has an upper limit for the recording size, which can be set in the "Maplab Options"-window (see chapter 3). The default value of this option is 32 megabytes. When loading a recording that would be too large,  $MA_{L}^{P}AB$  automatically loads just a part of it. It will notify you with an "Error"-window if this happens.

Using the controls in the lower 'frame' in the "MapLab"-window it is possible to change the loaded part. The first sample and number of samples can be set in edit boxes (if you set the number of samples too large, MAPLAB will reset it). The range can also be changed with the row of buttons next to these edit boxes; these are summarized in table 2.2.

These buttons change the loaded part, without affecting its size. The loaded size is only changed by  $MA_{L}^{P}AB$  if you set it so large as to make the recording larger than the limit.

## 2.3 The "Recording"-window

In the "Recording"-window (see figure 2.2), all leads are plotted together. If the datafile provides information on the electrode configuration, the positions of the ECGs will reflect this configuration. Otherwise, the ECGs will appear in a square, in the same order as the leads appear in the file. If the electrode configuration is known to  $MAP_LAB$ , but cannot be determined automatically from the file, it can be selected with the Options|Grid menu (more about grids in Chapter 4).

If the recording is long, only a part of it is shown. The maximum interval to be shown can be selected with the **Options** menu. By pressing the [F] or [D] keys, you can move forward and backward by half the amount shown in the ECGs. The [SHIFT-F] and [SHIFT-D] keys move by the whole range.

Exactly one 'channel' or lead at a time is known as the *current channel*. The current channel is marked with a black background in the "Recording"-window; it can be changed by clicking an ECG with the mouse.

Sometimes, it is necessary to tell  $MA_L^PAB$  that the signal quality of a lead is so bad that it should not be used for analysis. This is called "rejecting the channel". The current channel can be rejected by pressing  $\boxtimes$  in the "Recording"-window. By pressing  $\boxtimes$  again, it can be re-accepted. Rejected channels are marked with a red cross on top of their ECG.

By pressing  $\mathbb{P}$  or double-clicking on an ECG, the "Channel"-window can be popped up (see figure 2.3). In this window, a single large ECG is shown. This can be used to have a better look at a signal. The "Channel"-window can be closed using the "return" button.

#### 2.3.1 extended selections

The "Channel"-window can show more than one ECG at a time. You can select several channels by making an 'extend' selection with the mouse (using the middle button or the <u>SHIFT</u> key and the left button together) in the "Recording"-window. Channels that are part of the "extended selection" are marked with a blue border. They can be removed from the selection by 'extend'-selecting them again.

The extended selection can also be used in the "Standard ECG"-window. This way it is easy to print an arbitrary set of leads on paper. Another way to select channels for the 'standard' ECG is by dragging a rectangle with the mouse in the "Recording"-window. This creates a rectangular set, or "block" of leads, which can be printed in the same configuration in the "Standard ECG"-window (see section 2.6). The selected leads are marked with a yellow border.

#### 2.3.2 buttons and keys

Here is an overview of the buttons and equivalent keys in the "Recording"-window:

key(s)	runctionality
X,A	accept (rejected) channel
X,R	reject (accepted) channel
P,C	pop up "Channel"-window
F	move forward $\frac{1}{2}$ display
D	move backward $\frac{1}{2}$ display
SHIFT-F	move forward 1 display
SHIFT-D	move backward 1 display
	X,A X,R P,C F D SHIFT-F SHIFT-D

### 2.3.3 Menus

The following menus are found in the "Recording"-window:

Actions|refresh: redraws the "Recording"-window

Actions reject channel: marks the current channel as rejected

Actions accept channel: marks the current channel as accepted

Actions|toggle channel status: changes the status of the current channel between rejected and accepted

Actions|popup channel window: opens the "Channel"-window

- Actions|forward 1/2 interval: shows a different part of all leads in the "Recording"-window, moving forward by half the interval that is shown. The interval can be changed in the options window that corresponds to the "Recording"-window.
- Actions forward 1 interval: shows a different part of all leads in the "Recording"-window, moving forward by the interval that is shown.



Figure 2.2: The computer screen, featuring the "Recording"-window on the foreground, and, partially hidden, the "MapLab"-window and "Matlab"-window.

- Actions|backward 1/2 interval: shows a different part of all leads in the "Recording"window, moving backward by half the interval that is shown.
- Actions|backward 1 interval: shows a different part of all leads in the "Recording"-window, moving backward by the interval that is shown.
- Actions|quit window: closes the window.
- Options|grid: allows the user to select a grid; this is only necessary if the grid cannot be determined from the datafile.
- Options|Other options: opens an options window.

Manual: provides a short on-line manual.

Select |empty selection: makes the channel selection empty. (see section 2.3.1)

## 2.4 The "Channel"-window

The "Channel"-window is a natural basis for most work with  $MA_L^PAB$ . This window shows one or more electrograms in a single frame. Like the "Recording"-window, it may show just a part of a lead. If that is the case, you can move through the lead with the 6 buttons positioned below the ECG, or the D and F keys, as in the "Recording"-window. The window contains lots of buttons and menus which initialize actions, and it contains *markers*: vertical lines that identify special time instants in the ECG.

## 2.4.1 Buttons

An overview of the buttons and equivalent keys in the "Channel"-window is presented in table 2.3.

## 2.4.2 Markers

There are six kinds (or *classes*) of markers, called *Begin*, *End*, *Event*, *Point*, *Baseline*, and *Zero*. The "Point" class is special because point markers can be different for different channels. This class was invented to accommodate the analysis of intracardial recordings, where leads represent mainly local activities. The other markers are the same for all channels. There can be several *instances* of each marker class. It is common practice to have several Point and Baseline markers, but usually there is only one instance of the other classes.

The markers serve various purposes:

- The Begin- and End-markers delimit the interval for an integral map or activation map, or the interval where activation detection or Fourier transformation will take place.
- The Baseline markers are used to indicate samples where the potential may be assumed zero, for the purpose of baseline correction.
- The Point markers indicate activation times for activation maps and such.
- The Event markers can be used as helpers, to remember samples, or to indicate where, e.g., a potential map should be computed.
- The Zero markers are used to define 'time zero.' There can be multiple zero times; the time corresponding to a sample number that is between two Zero markers is usually computed relative to the earlier Zero marker.

The markers will be more thoroughly discussed in later sections on specific applications.

## **Controlling Markers**

Above the ECG in the "Channel"-window is a series of controls that deal with markers:

- leftmost is a popupmenu which selects the *current marker class*. This is important for the creation of markers, as we will see later. The current class is changed if you drag a marker with the mouse. This menu is referred to as the "*Class Menu*."
- next to it, is an edit box that shows, and sets, the *current marker of the current class*, also called the "current *instance*" of this class, or the *Current Marker*. This marker can be moved with the '<' and '>' buttons, as will be discussed later. This edit box is referred to as the "Instance Box."
- next to the instance box are two buttons that increment and decrement the current instance of the current class.
- the hand symbol can be used to start dragging the current marker.

button	key(s)	functionality
align		align complexes
unalign		return from aligned mode
goto		move to the current marker
baseline		do (piecewise) linear baseline correction
		reject current channel
del Chn		toggle current channel's rejected status
del Cpx		reject current complex (in aligned mode only) (tog-
find $dV/dt$		find activations (max. negative deflection) in the cur-
all $\mathrm{dV}/\mathrm{dt}$		find activations (max. negative deflection) in all
	Μ	find max. positive deflection
U		go up in grid
D		go down in grid
		go right in grid
R		redraw
Next	Ν	go to next channel
Prev	Ρ	go to previous channel
	SHIFT-N	go 5 channels forward
	G	go 5 channels forward
	SHIFT-P	go 5 channels backward
	Т	go 5 channels backward
return	SHIFT-R	go back to the "Recording"-window
	Q	quit this window
in		zoom in by $20\%$
out		zoom out by 20 %
ht		fit vertical axes limits
<		move to the beginning of this lead
>		move to the end of this lead
<<		move left by 1 block
>	し 同	move right by $\frac{1}{2}$ block
>>		move right by 1 block
	<u>C</u>	center horizontally on mouse position
+	$\pm$	increment current instance
_		decrement current instance
<		step current marker left and activate slaves
>		step current marker right and activate slaves
		drag current marker right
		go to next complex (in aligned mode only)
(hand)	<u></u>	start dragging the current marker
(hat)		create a new marker of the current class and start
. ,		dragging it
x		delete the current marker
	SHIFT-I	interchange Begin and Event markers
		interchange End and Event markers

Table 2.3: Buttons and keys in the "Channel"-window



Figure 2.3: The "Channel"-Window

- the hat symbol creates a new marker of the current class, and starts dragging it
- the 'x' button deletes the current marker.
- the larger edit box, after the text "at" shows the time of the current marker, and is referred to as the "*Time Box*." This box can be edited to move the current marker. You can make a marker the 'current marker' by clicking it with the mouse, or using the Class Menu and the Instance Box.
- The '<' and '>' buttons on the right of the Time Box move the current marker left and right by one sample, respectively.
- With the 'goto' button, you can move to the current marker, that is, shift the horizontal axes limits such that the current marker preferably appears in the middle.

#### Moving Markers

The markers in the "Channel"-window can be moved in four ways:

- dragging with the mouse: put the mouse pointer above the marker; press the left mouse button and keep it pressed; move the mouse to the desired sample and release the button. The marker will move with the mouse. However, it will not move until dragged by a considerable amount ( $\sim 1/50$  th of the axes width). This is to prevent accidental dragging of a marker when you just want to click it.
- numerically: click the Time Box with the mouse; edit the value, using <u>BACKSPACE</u> to delete, and press <u>ENTER</u>. The current marker will jump to its new location. The value in the edit box is given in *milliseconds*. The current marker can be set using the Class Menu and the Instance Box, or by clicking it with the mouse.
- with buttons: the "<" and ">" buttons move the current marker left and right by one sample and then activate the slaves (see section 2.4.4).

• with keyboard keys: the 
→ and → keys move the current marker without activating the slaves.

By starting a drag procedure on the hand sign you can drag the current marker. This may come in handy if the marker is out of the current range, or if you want to find out which marker of a class is 'current'.

#### **QRS** Detection and Activation Detection

The QRS Detection and Activation Detection procedures can move certain markers automatically. This is explained later.

#### **Creating and Deleting Markers**

New markers can be created by clicking the hat sign. This creates a new marker of the current class. The current class can be selected using the Class Menu. The marker comes into existence when you press the left mouse button on the hat, and you can start dragging it to the desired location immediately. If you press and release the button on the hat, without moving the mouse, the marker will be left at sample 0, and will be deleted when the "Channel"-window is rebuilt or switched to another channel.

Markers can be deleted by clicking the middle mouse button on them. With a twobutton mouse, use **SHIFT** with the left mouse button instead.

#### Point Marker Types

Point markers can have several types. The type of a point marker can influence its rendering in, for example, an activation map (see 2.11). The following types are defined:

- block: no activation can happen here
- *unsure:* the operator is unsure about this
- *dp*
- *tp*
- A1 first activation
- A2 second activation
- A3 third activation
- A4 fourth activation
- A5 fifth activation

#### 2.4.3 Zooming

By selecting a rectangle (pushing the left mouse button, dragging, and releasing) you can zoom in to a part of the ECG. Normally, only the vertical axis is zoomed, but this behaviour can be changed via the **options** menu, by setting the "enable x axis zoom" option and "enable y axis zoom" option items (see Chapter 3).

Zooming out is achieved with the middle mouse button. If you have a two-button mouse, you can use the <u>SHIFT</u> key with the left mouse button instead. This returns the

window to the previous 'zoom state'. If there are no more zoom states left, it zooms out by 20%.

If the right mouse button is pressed, the vertical range will be fitted in the same way as when switching channels or when moving to a different part of the same lead. This is done as if the "vertical fit method" option, (see chapter 3), was set to "fit global."

The 'in' and 'out' buttons in the lower left corner of the "Channel"-window zoom in and out by 20%, respectively, without changing the axes limits and the offsets of the lead(s). With the 'fit' button, the axes limits can be adapted to the new size.

#### 2.4.4 Slaves

The slaves of the "Channel"-window are actions that are performed when some of the markers are dragged or dropped. Some of them are associated with their own windows. There are slaves for potential maps (see section 2.8), integral maps (section 2.8), polar localization) maps (section 2.9), localization, baseline correction, activation maps (section 2.11), and histograms (section 2.12). The "Spy" slave is a small window that shows time, sample number, dV/dt, and amplitude (V) for the current sample.

#### 2.4.5 Multiple leads

The "Channel"-window can show more than one lead at a time. Extra leads can be selected in three different ways, depending on what kind of lead is required:

- regular leads are selected by making an "extended selection" in the "Recording"window (see section 2.3).
- extra leads, that is, leads that are linear combinations of regular leads, such as 'VF' in a Body Surface Map recording, are selected via the Select|Extra Channels menu.
- special leads are selected via the Select|Special Channels menu. Special leads are things like dV/dt of the current channel, SumAbs, and "the left neighbour of the current lead."
- A set of four channels, surrounding the current channel (based on the lead number), can be selected using the Select|Five in the "Channel"-window. This set will be updated when the current channel changes.
- The set of 4-connected neighbours of the current channel (up, down, left, and right) can be selected using the Select|Four+1 in the "Channel"-window. This set will be updated when the current channel changes.

Point markers are only shown for the current channel, which is shown by a black line; the other leads are blue.

Initially, all shown leads have different offsets, to make them all well visible. The offsets can be changed by dragging the triangular handles on the right of the ECG. If a handle is dropped, the vertical axes limits are automatically adjusted to make all ECG's fit.

The set of leads that is shown in the "Channel"-window can also be printed in the "Standard ECG"-window, by selecting the Options|Selection|Selection menu in that window. See section 2.6 for more information.

Leads can be unselected by selecting their menu entry again. The selection can be emptied at once with the Select|empty selection menu.

#### **Dirty Tricks with Five Channels**

If the lead selection in the "Channel"-window is set to "Five" (menu Select|Five) or "Four+1" (menu Select|Four+1), Points are marked in the extra leads as well. This is done with cyan dots that look like ordinary Point markers, but behave a little differently: before they are dragged they make their own channel the current channel. This means that you can drag them as usual, but while you are dragging the display will change such that the selected channel is in the middle.

It is also possible to select a different channel by clicking an ECG, or by clicking a (fake) Point marker without dragging it.

When displaying five leads at a time, it may also be useful to move by five leads at a time. This can be done with the G and T keys, or, alternatively, with the <u>SHIFT-N</u> and <u>SHIFT-P</u> keys. The first pair (g,t) was chosen for ergonomic reasons: an experienced typist may type these while keeping his right hand on the mouse. The second pair (N,P) was chosen for consistency with the <u>N</u> and <u>P</u> keys; see also table 2.3 on page 20.

#### **Special Channels**

The availability of special channels depends partly on the recording type and grid definition. The following may be available:

- null, an all-zero channel, useful as a baseline for other channels
- up, the upper neighbour of the current channel, or null if there is no upper neighbour
- down, the lower neighbour
- *left*, the left neighbour
- *right*, the right neighbour
- *ulft*, the upper left neighbour
- *urt*, the upper right neighbour
- *llft*, the lower left neighbour
- *lrt*, the lower right neighbour
- dV/dt, the time derivative of the current channel, computed with a symmetric twopoint method
- DD, the second time derivative of the current channel
- *MA*, the mean absolute value of all channels. If body surface channels are present, the mean absolute value of all body surface channels (this is a feature).
- *SAD*, SumAbsDiff, the sum of the absolute values of the time derivatives of all channels
- DSA, DiffSumAbs, the time derivative of the sum of the absolute values of all channels
- *DDSA*, DoubleDiffSumAbs, the second time derivative of the sum of the absolute values of all channels

#### 2.4. The "Channel"-window

- *SADD*, SumAbsDoubleDiff, the sum of the absolute values of the second time derivative of all channels
- QRS1, first experimental QRS detector signal
- QRS2, second experimental QRS detector signal
- dV/dx, spatial derivative (bipolar lead) in the horizontal direction
- dV/dy, spatial derivative (bipolar lead) in the vertical direction
- Lapl, until MAPLAB version 1.38 this was the true Laplacian, now it is  $-\frac{1}{4}\nabla^2 F$ . In an endo/epicardial recording, this is approximately proportional to the membrane current [11], and may be used for activation detection.

## 2.4.6 Automatic QRS Detection

Finding QRS complexes consists of *QRS detection*, by which we mean the recognition of a certain time instant as being part of a QRS complex, and *QRS expansion*, the delineation of a QRS complex.

## **QRS** expansion

QRS complexes can be expanded automatically, using an algorithm by Kemmelings and Linnenbank [6, 7]. If you double-click in a QRS complex, the current Event marker will move to the clicked sample, and the Begin- and End-markers will move to the begin and end of the complex. This is only tested for BSPM recordings, and is not fully reliable.

## **QRS** detection

Provisions for QRS detection are under development.

## 2.4.7 Automatic Activation Detection

In endocardial electrograms, activations—which are recognized as steep negative deflections can be detected automatically. With the "find dV/dt" button, detection is applied to the current channel, and activation times are indicated with Point markers (see section 2.4.2). With the "all dV/dt" button, activation detection is applied to all channels. This can take considerable time for long recordings; the detection routine still needs to be optimized. It is recommendable to use the "Only begin to end" option for activation detection (page 40). The algorithm is influenced by several parameters, which can be changed via the **Options** menu (see Chapter 3 for more information).

## 2.4.8 Fourier Transforms

If the Actions|Transforms|fft menu is selected, Fast Fourier Transformation is applied to the recording between the Begin and End markers, and the waveform is replaced by the Fourier transform. All future actions apply to the transformed data, so that you can make fft maps, fft-grams, and even fft's of fft's. To go back to the waveform, you will have to reload the recording using one of the "reload" buttons in the "MapLab"-window.

#### Warning

Fast Fourier Transforms *always* result in interesting patterns. None of us knows, however, what these patterns imply in case an ECG is transformed. Also, the use of Fourier Transforms on signals whose non-repetitive features provide real information, obscures real data. Fast Fourier Transforms are therefore highly attractive to the young and the ambitious; they intrude on the minds of the inquisitive, distract researchers from their research, and cause dizziness and glaucoma. Don't blame us if they even turn out to break your dishes.

## 2.4.9 Signal Alignment

Signal alignment is an intermediate step in signal averaging, but may also be useful by itself. With the "align" button, you can apply automatic signal alignment to the entire recording. The "Channel"-window shows all recognized complexes overlapping, and with the space bar you can select the *current complex*. This complex can be deleted, if necessary, with the "del cpx" button. Only the non-deleted complexes are averaged.

## 2.4.10 Signal Averaging

With the Actions|Transforms|average menu, complexes are averaged between the Begin and End markers. The recording is replaced by the averaged waveform. All future actions apply to the averaged recording, until the original recording, or another, is (re)loaded.

## 2.4.11 Other Transforms

Fourier transformation and signal averaging are examples of what we call *Transforms*: operations that convert the entire recording into something completely different. These transforms are, in general, irreversible; their effect can only be undone by reloading the datafile (which deletes the markers etc.!).

All transforms are accessible via the Actions|Transforms menu in the "Channel"-window. Additional transforms available from this menu are:

- dV/dt This takes the first temporal derivative of all leads, and replaces the recording by this data. This may be useful if derivatives of channels must be compared.
- **clip** This option clips the recording data symmetrically above and below the zero line, at a specified value. A dialog window is created to ask for the value.
- **invert** This option inverts the data, that is, it makes positive values negative and vice versa.
- Laplacian This option replaces each lead by the "Laplacian" or "coaxial" lead. The algorithm depends on the grid type; currently, only experimental support for Constellation basket is included, and only when the basket geometry has been taken into account, by an external program. If the geometry is unknown, or the recording is not from a Constellation basket, the two-dimensional electrode configuration is used to compute the Laplacian.

Other transforms may be added to this menu.

### 2.4.12 Filters

The Actions|Filters menu will access some often-used ECG filters. Currently, only a spike filter is implemented.

#### 2.4.13 Menus

The following menus are found in the "Channel"-window:

Actions|Figriddef.weblters: gives access to filters

- Actions|Transforms|average: signal-averages the recording, from the Begin marker to the End marker (see section 2.4.10).
- Actions|Transforms|fft: Fourier-transforms the recording, from the begin marker to the end marker (see section 2.4.8).
- Actions|Transforms|dV/dt: replaces the entire recording by its first temporal derivative.

Actions|Transforms|clip: clips the recording at a specified value.

Actions|Transforms|invert: inverts the recording.

Actions|Listfile|add to list: add markers to listfile

Actions|Listfile|add all: add all complexes to listfile

- Actions|shift Begin+End forward: shift the current Begin marker to the position of the End marker, and shift the End marker forward by the same amount.
- Actions|shift Begin+End back: shifts the current end marker to the position of the current begin marker, and moves the current begin marker backward by the same amount
- Actions|interchange Begin/Event: shifts the current event marker to the position of the current begin marker and vice versa
- Actions|interchange Event/End: shifts the current event marker to the position of the current end marker and vice versa
- Actions|label points by time: adds marker flags to all Point markers, based on the order in which they appear: the earliest is labelled 'A1', the second 'A2', etc.
- Actions|label points by instance: adds marker flags to all Point markers, based on their instance numbers: the Point marker with instance number 1 is labelled 'A1', the second 'A2', etc.
- Actions|histogram of dV/dt: Create the "dV/dt histogram"-window, which shows a histogram of the dV/dt values at the Point markers. This window features controls to delete Point markers whose dV/dt value is under an adjustable threshold.

Actions|Full Reset: re-creates the "Channel"-window

Actions|quit window: closes the "Channel"-window

- Options|find dV/dt options: opens an options window that controls the activation detection algorithm
- Options|other options: opens an options window that controls the "Channel"-window

Manual: provides a short on-line manual

- Flag: is used to add labels to Point markers. There are labels for activation block, for unsure activations, and for numbered activations (A1–A5). Numbered activations can be used to select one of several 'waves' in the same window in an activation map: see the options for the "Activation Map"-window.
- Slaves: enables you to switch the *slaves* of the "Channel"-window on or off (see section 2.4.4).

Select: selects channels to be shown in the "Channel"-window (see section 2.4.5).

## 2.5 Saving Things

Several kinds of information can be saved, after pressing the "save" button in the "MapLab"window. When this button is pressed, the "save"-window pops up, which gives the following choices:

- save markers A reference to the datafile, as well as the marker values and the rejected status of the channels are saved in a *metafile* (see section 2.5.1).
- **overwrite metafile** The same, but no file selector box is presented. Instead, the information is written in the most previously loaded metafile, which is usually where it came from. This facilitates making small changes to existing metafiles.
- **copy & append rejected** Copy the datafile, and append the rejected status of the channels to the copy.

save data save the actual—perhaps modified—recording data in a datafile.

cancel do nothing, just close the "save"-window.

A lower-level method to save data is using the rsave function. Refer to the MA<sup>P</sup><sub>L</sub>B reference manual for further information.

You can save any data using MATLAB's *save* command, in MATLAB's native "mat-file" format. Using the *fopen*, *fprintf*, and *fclose* functions, you can save data in ASCII files.

## 2.5.1 MAPLAB metafiles

It is often desirable to store information about a recording, such as marker values (see section 2.4.2) and rejected status in a file for later reference. The obvious place for this is the datafile itself; however, this would be impractical for two reasons. In the first place, datafiles are often stored on a read-only (or rather 'write-once') medium, such as a CD-ROM. Moreover, there are often several sets of markers related to the same datafile, for example if multiple complexes are analyzed or if unrelated experiments are performed with the same data.

Therefore,  $MA_L^PAB$  metafiles were introduced. These are generally very small files (a few kilobytes long) that contain a reference to a datafile, values for markers, and information on the signal quality of the leads. They can be loaded in  $MA_L^PAB$  like ordinary datafiles;  $MA_L^PAB$  will then load the referenced datafile and markers etc. will be set automatically. Metafiles can also be used in batch scripts for non-interactive processing. It is recommended but not required that maplab metafiles have the extension .mmf.



Figure 2.4: Computer screen, with the "Standard ECG"-window on the left-hand side.

A metafile can be created by pressing the "save" button in the "MapLab"-window, and then pressing the "save markers" button in the "save"-window which pops up. The filename for saving can then be selected using the standard directory browser window that pops up subsequently.

#### 2.5.2 Saving Data

The button "save data" in the "save"-window can be used to save the recording data in a datafile. This is rarely necessary; changing recording data in an interactive package like  $MA_L^PAB$  is dangerous because the changes are not documented, and it will be hard to know exactly what the saved datafile represents. If large amounts of data are to be processed using manually introduced guidelines, such as manually selected events, it is generally better to store the guidelines, and do the processing in a batch job using lower-level routines from the  $MA_L^PAB$  ib library.

## 2.6 The (standard) ECG

A standard 12-lead ECG can be created with the Actions|stdecg menu in the "MapLab"-window. The ECG will be plotted in the "Standard ECG"-window. An example is show in figure 2.4.

Instead of a standard ECG, this window can also show other selections of leads; in fact there are six methods to select leads for this:

- **Standard ECG** The standard 12-lead ECG. This will only work if the required channels are present, which is not the case for, e.g., an endocardial recording.
- **User-defined numbers** The lead numbers are given by the two-dimensional array *bso\_secg\_nrs*. The position of each ECG will correspond to its index in this matrix. The value zero means "no channel". For example:

bso\_secg\_nrs = [ 10, 11, 12; 13, 14, 0; 21, 22, 25 ];

A definition like this can be typed on the command line in the "Matlab"-window, or put in a script or function m-file that can be invoked from the command line or from a user-defined menu.

**User-defined names** The lead names are given by the two-dimensional cell array *bso\_secg\_names*. Each cell must contain a string that can be found as an element of *ElecName* or *ExtraChannelNames*. The position of each ECG will correspond to its index in the cell array. The empty element means "no channel". For example:

A definition like this can be typed on the command line in the "Matlab"-window, or put in a script or function m-file that can be invoked from the command line or from a user-defined menu.

Selection The "extended selection" (see sections 2.3, 2.4.5) is used.

Selection + Cur. Chan. The "extended selection" and the current channel are used.

Block The "block" from the "Recording"-window (see section 2.3) is used.

The selection method is chosen via the **Options**|selection menu in the "Standard ECG"-window.

#### 2.6.1 Menus

The following menus are found in the "Standard ECG"-window:

Print prints the window on the default printer

Actions quit window closes the window

Options|selection gives access to the five lead selection methods discussed above

Options|other options opens an options window

Repeat|update this window redraws the window, with the current status of markers and options

Repeat|new window creates another "Standard ECG"-window, with the current status of markers and options.

## 2.7 Printing Body Surface (Integral) Maps

A Body Surface Potential Map suitable for printing can be created with the Actions|map menu in the "MapLab"-window. A Body Surface Integral Map can be created with the Actions|intmap menu.

The time instant taken for a potential map is at the current Event marker. For integral maps, the interval from the Begin to the End marker is used.

A screen preview of the "Map"-window is given in figure 2.5a. Like the "Standard ECG"-window, it can be printed on the default printer. It can also be printed as a PostScript file, or any other graphics format, using MATLAB's *print* command. If such a file is to be incorporated in another document, it may be useful to set the *bso\_Output* option to 2 (see Chapter 3). An example of PostScript output is given in figure 2.5b. PostScript is a resolution-independent vector graphics format, and is very useful for final, high quality graphics (see section 2.13).

The "Map"-window and "Integral Map"-window also mention the nondipolar content of the map. Other indices of nondipolarity may be added in the near future.

The "Integral Map"-window features a polar diagram of the normal left ventricle, where a localization of the map is given. This is an experimental method [16], and only meaningful if the map is a QRS integral map originating from a structurally normal human left ventricle. The "Localize"-windows provide more localization methods.

#### 2.7.1 Menus

The following menus are found in the "Map"-window and "Integral Map"-window:

Print prints the window on the default printer

Manual provides a short on-line manual

Options opens an options window

- Repeat|update this window redraws the window, with the current status of markers and options
- Repeat|new window creates another "Map"-window or "Integral Map"-window, with the current status of markers and options

Quit closes the window

## 2.8 Interactive Body Surface Maps

The "Map"-window and "Integral Map"-window described in section 2.7 are intended for fine prints, and consequently slow. For interactive use, the "Fast Map"-window and "Fast Integral Map"-window are more suitable. These windows are 'slaves' of the "Channel"window (see section 2.4), and can be enabled with the Slaves menu in the "Channel"window. These windows have a print menu which activates the "Map"-window and "Integral Map"-window.

The following controls are found in these windows:



Figure 2.5: (a.) The "Integral Map"-window (left). On the right side of the window the "Fast IntMap"-window, "Maplab"-window and the "matlab"-window are visible; the "channel"-window is partially visible on the bottom of the screen. (b.) Example of a Body Surface Map. This picture has been created by writing the contents of the "Map"-window to a PostScript file with the command print -deps -f5 xmplmap.eps and incorporating it in this document.



Figure 2.6: The "Fast IntMap"-window and two "Localize"-windows. The integral map shown is used by the "Localize"-windows to compute an endocardial position.

button	key(s)	functionality		
reset	R	Reset the window, reconsidering the color scale.		
	Q	Quit the window. It will be re-opened when an update		
		would normally occur.		
	С	Close the window. No updates will take place until		
		the slave is enabled again.		

## 2.9 Localization of Ectopic Beats

The "Fast IntMap"-window can show, next to the map, the five best correlations with a database. The database is selected with the Database menu in that window. Nonstandard databases can be selected with the Database (select) menu. The commonly used databases [22, 23, 24] are copyrighted, they are not included in the MAP<sub>I</sub>AB package.

#### 2.9.1 The Localize Windows

Both Discrete and Continuous localization is provided by the "Localize"-windows, a series of similar windows which can be invoked through the Slaves|Localizer menu in the "Channel"-window, and can display continuous localization results in several endocardial representations. An arbitrary number of "Localize"-windows can be created, so that localization results can be shown in different representations at the same time. Two "Localize"-windows are shown in figure 2.6.

Continous localization results are shown with yellow numbers in a schematic representation of the left ventricle (right ventricle and atria will be added). After an update, which occurs when the Begin or End marker in the "Channel"-window is moved or baseline correction is performed, the highest number moves to a new location. With the "keep" button, the highest number can be fixed, and a new number will be used at the next update. With the "clear" button, all numbers are cleared. Every "Localize"-window has a "keep" and a "clear" button, and they apply to all "Localize"-windows simultaneously.

By double-clicking one of the numbers, the underlying data is loaded into  $MA_{L}^{P}AB$ . This way, you can resume the analysis of a previously created map. If you press the "keep" button again, however, it will be stored as a new map.

The controls for the "Localize"-windows are summarized in the following table:

button	key(s)	functionality
keep	K	Keep the last localization
reset	R	Reset the window, reconsidering the color scale.
	Q	Quit the window.
clear	С	Clear all points
fork	F	Create a new "Localize"-window with the same set-
		tings as this window

The "Localize"-windows each have an **Options** menu, which accesses an options window. The options apply to all "Localize"-windows. They also have a **Window** menu, that applies only to a single window. This menu has the following submenus:

- Window|View: choose a representation. Examples of representations are: a Left Ventricular (LV) polar diagram, LV cut-open diagram, LV and RV pseudo-3D diagrams, and RA Anteroposterior and Posteroanterior diagrams. More views may be implemented later.
- Window|Database: choose a database for comparison. Common databases are NLV, NRV, AMI, and IMI [22, 23]. Recently, the RA database for the right atrium has become available [24].
- Window|Algorithm: choose an algorithm for continuous localization. Currently, only an algorithm for the normal left ventricle is available [16]. This can also be of use in infarcted left ventricles.
- Window|quit: close this window.

## 2.10 Cross-correlation of Maps

Provisions for map correlation and other comparison methods will be made soon. Instead of computing a cross-correlation, you may apply continuous localization as discussed in section 2.9.1, which also gives an impression of the (dis)similarity between maps.

## 2.11 Activation Maps

Activation maps are created with the Actions|activation map menu in the "Channel"window, and displayed in the "Activation Map"-window (see figure 2.7). The routines for creating activation maps are under development. The "Activation Map"-window has an Options menu that gives access to three options windows:

- Options|lead selection: Options for lead selection. Not all leads in a recording may need to be used for an activation map. The selection of leads for the activation map is based on the lead types: The electrode configuration file defines the type of each lead or electrode as 1, body surface; 2, endocardial; 3, intracardial; 4, epicardial; 5, subcutaneous; 6, extremity. The controls in this options window determine which types are used. The default is to use types 2, 3, and 4. This is of course only useful for mixed-type recordings, e.g. combined body surface map and endocardial basket catheter recordings [21, 25].
- Options|other options: Most of the options that determine the appearance of the activation map. Data for a pseudocolor map, texts, and contours (isochrones) can

							Ę
File Window Help	for Activation Map		in	Com	in	Con	/
isochrone interval		reset default	1 Martin	TER	YET	TOX	9
activation time limit	40 + -	re	Fig 4	: Activation Map	0001005	501950	न
do updates		re File Window Help A	Actions Options	•			
pseudocolor map	activation time 💷	16 D16074 499 B 12 (D160741		000			
texts	milliseconds 💷	window = 5:136 zero = 268, iint = 5 cdata = activation tin	, limits = 4e+01ms 0.2V/s , ne_text = milliseconds	550			
time/sample nr	from zero 💷	16-May-2000 12:43:10	ino, torre inimocoorrido				
isochrones	isomap(c) 💷		6 7 8 9 10	11 12 13 14 1	15 16 17 18		
isochrone levels	fixed interval 🖃		18 475 106 114 121	122		-120	
number of levels	7 + -			125		- 100	
and are drively				125		-80	
activation selection	25 + -	5 52 59 41 34 58	131 40 101 109 113	127		60	
color scale		6 60 58 45 39 45	75 87 99 10 720	Rax x x	× × × ×	100	
show colorbar		re 7 X 62 53 46 67	6) 9/ 100 h 12/123	135		-40	
		8 63 63 53 50 51	ty 88 116 131	132		-20	
<u> </u>		9 63 63 63 54 54	64 116 129	131		-0 bik	
		10 64 64 64 5 <del>7 57</del>	64 64 116 31	131		~~~	1
cancel	apply	11 64 65 64 64 60	63 64 131 131	132	-	-bg refresh	
							Ĩ
	3.0	Ma all	All a	20	All a	199	
			1		1 .		

**Figure 2.7:** The "Activation Map"-window, displaying a circular tachycardia with a functional line of block above and an anatomical zone of block below. Partially shown is an options window associated with the "Activation Map"-window.

be selected, as well as several parameters that have an influence on these, such as the time unit and reference, cell size, and activation time limit.

• Options more options: Rarely used options.

A complete list of options is given in  $\S3.9$ .

If the "do updates" option is on (it is by default) the activation map is updated if the Begin or End marker in the "Channel"-window is moved. The window does not react on changes in the Point markers. An update can be forced using the 'update' button in the "Activation Map"-window.

The following controls are found in the "Activation Map"-window:

button	key(s)	functionality
update	R	Update (refresh) the window, considering the current
		values of the Point markers
	SHIFT-R	Do a full reset.
	С	Quit the window.
	Q	Quit the window.
	Х	Toggle the rejected status of the current channel

If there are multiple Point markers in the current window, at most two values will be printed in the activation map. MAPLAB has to select one of them in order to choose a color for the cell. This choice is based on the "activation selection" option (see section 3.9). If it is necessary to distinguish independent activation fronts that overlap in time, it may be helpful to use *marker flags*. Using the Actions|label points by time menu in the "Channel"-window, flags A1, A2, ... can be attached to all Point markers. These flags can be hand-edited using the Flag menu in the same window.

#### 2.11.1 Isochrones

Isochrones can be drawn in the activation map by selecting a contour algorithm in the pull-down menu for the "isochrones" option. Currently, three algorithms are available:

simple uses the standard MATLAB function contourc.

- isomap uses a specialized function for isochrones, which is able to detect lines of block. It shows these with a special block contour, instead of drawing crowding contour lines. Block is considered present if the propagation speed in ms per unit distance is less than the activation time limit, which is set with the "activation time limit" option.
- **tricky** uses *contourc* with a special trick to allow multiple activations in a lead within the time window. This enables crossing contour lines.

#### 2.11.2 Activation Tracking

If the "pseudocolor map" option for the "Activation Map"-window is set to "flood fill", an activation tracking algorithm is invoked that colours electrode positions different if their activations are independent. If independent regions meet, they will be joined and get the same colour. The program tracks from the Begin marker to the End marker. By moving (and dropping) the End marker, you can see what is going on.

## 2.12 Histograms

There are presently three windows in the MAPLAB package that show histograms of some data. Two of these will be eliminated in the near future, and their functionality will be incorporated in the third: the "Histogram"-window.

In the "Histogram"-window, two kinds of data can be shown simultaneously. One kind of data is shown as a histogram, where the data are displayed along the horizontal axis, and the vertical axis indicates the frequency. An optional second kind of data is shown with color: The histogram consists of "cells," each corresponding to a data element, and the second data value of each element is indicated using the color of the cell.

This is somewhat complicated because a cell can represent three different data elements: a Point marker, a vertex of the current grid, or a face of the current grid. If a cell represents a Point marker, the special color data type "compare" can be used to find Point markers with the same activation time or in the same channel.

The "Histogram"-window can have several instances, so that histograms of various kinds of data can be shown simultaneously. It has no Options window (yet).

The "Histogram"-window is currently in the Alpha testing stage. It will be documented more completely when it has reached the Beta stage.

## 2.13 The Fine Print

If used with care,  $MA_L^PAB$  is able to produce professional quality graphics ready for print or incorporation in a graphics program. The authors use METAPOST and  $T_EX$  if further processing is required. These are powerful but radically novice-unfriendly programs and therefore not an option for everyone. Some people may simply hand output to their illustration department. For these purposes, the "Standard ECG"-window and "Map"-window are
particularly useful. The "FastMap"-window, "Channel"-window, "Recording"-window, and such are less suitable. A good window for isochrones must still be created.

MATLAB is able to send graphics to a variety of printers, and to files, in any desirable format. For final output, the PostScript format may be the best choice, because it is resolution independent, which means that it provides the best possible results on any device, be that a modest 300 dpi laserprinter or a 2540 dpi digital phototypesetter.

## 2.13.1 glossies

Journals often require graphics on photographic ("glossy") paper. This is useless for blackand-white graphics, but, being at their mercy, you may want to create "glossies". Digital phototypesetters can produce these directly. However, it is rather uncommon to have access to this kind of equipment; a good alternative—for black/white graphics, that is—is printing on a 600 dpi laserprinter and photographing on graphic (black-and-white) paper.

## 2.13.2 size

It is usually best to produce a graphic in the intended size in  $MA_L^PAB$ , and print it enlarged as much as possible, to make optimal use of the hardware's resolution. The journal's compositor will scale the graphic anyway, to make the pages look nice, and probably won't object to some extra scaling. If you let  $MA_L^PAB$  itself produce oversized graphics, the relation between line width and graphics size will be wrong; the lines will be too thin when the graphic is reduced and may disappear or look grey after reproduction. Also, the relation between font size and objects like maps may be wrong, resulting in unreadably small texts after reduction.

# 2.13.3 greyscale and color graphics

The typical grey area in Body Surface Maps, when printed on a black-and-white laser printer and submitted to a journal, is often rendered badly [3]. This is due to differences in the raster of the graphic and the raster used by the printer. Reproduction on photographic paper, sometimes demanded by journals, does not help. Better results are obtained with real greyscale graphics. Digital photography is fine for grey patches, but produces hairy lines [14]. The best results we have seen were obtained with a dye sublimation printer, which produces real halftones and sharp lines [15].

For pseudocolor maps, a dye sublimation printer or better is strongly recommended. Ink-jets and thermal wax printers are inappropriate, because they cannot really mix colours; they create the illusion of mixed colours by drawing many tiny spots with a limited set of colours, and the colour impression is determined by the number of spots in the different colours.

Of course, any device is suitable for previews. An inkjet printer is fine for, e.g., activation maps, because it prints on normal paper and one can easily draw on the print.

# Chapter 3

# Options

One should absorb the colour of Life, but one should never remember its details. Details are always vulgar.

-OSCAR WILDE, The Picture of Dorian Gray

The following sections discuss the options found in all options windows of the  $MA_{L}^{P}AB$  package. Each section discusses one options window. The options are listed with their names, as they appear in the windows. After the description of an option, the name of the associated variable and its default value are given. The use of this will be explained in Chapter 4.

# 3.1 Options for $MAP_LAB$

The options in the "Options for MapLab"-window, which can be accessed through the Options|Maplab options menu in the "MapLab"-window, have a global impact on the MAPLAB package:

- use colours Use colours or grey values for  $MA_L^PAB$  windows and their contents. (*bsoi\_common.ColourMode* = 2)
- figure scale Size of a4-size windows, relative to the physical size of a4 paper (only correct if MATLAB knows the screen size correctly). (bsoi\_common.FigScale = 0.75)
- **sampling rate** The sampling frequency of the recording (Hz). In normal circumstances, this is determined correctly from the input file, and should not be changed. Particularly, the aspect ratio of ECGs, the values in integral maps, and the activation map algorithms depend on this. ( $bs\_SamplingRate = ...$ )
- **baseline algorithm** Code for the baseline correction algorithm to be used. Possibilities are:
  - 1: Linear interpolation with two baseline points.
  - 2: Only offset correction, using only one baseline point.

For each baseline point, the mean value of a small interval is computed.  $(bsoi\_common.BaselineAlgorithm = 1)$ 

baseline width time interval to use when computing baseline zero points, in milliseconds.

 $(bsoi\_common.BaselineWidth = 20)$ 

- window style This options determines if hardcopy maps are drawn in A4-sized windows (code 1) or in windows that are just large enough (code 2). The latter may be useful if one wants to print to a graphics file instead of a printer. ( $bsoi\_common.output = 1$ )
- font size Font size in printer's points (pt).
   (bsoi\_common.fontsize:) depends on the screen size, but is at least 10 pt.
- font weight (bsoi\_common.fontweight = 'demi')
- max recording size Maximum size of the internal recording matrix in MegaBytes. If internal storage of the recording would be larger than this,  $MA_L^PAB$  loads only a part of the recording. As a rule of thumb, his size should be set to about half the amount of the computer's physical RAM to ensure rapid loading of recordings. If this value is larger than the physical RAM size, the computer may attempt to load recordings larger than its RAM and consequently start swapping heavily. If it is really too large, some operating systems may crash when a really big recording is loaded. (*bsoi\_common.maxsize* = 32)
- **debug mode** If debug mode is off,  $MA_L^PAB$  will attempt to catch unexpected errors and give a short error message in a special popup window. If debug mode is on,  $MA_L^PAB$  will not catch errors; no error window will appear, but instead the full error message (with context) will be printed in the MATLAB command window. Presently this is the only way to obtain such information from MATLAB. Obviously, this option is meant for hackers only. (*bsoi\_common.debug* = 0).
- file wildcard Filter specification in UNIX shell style or DOS style, which is used by the window for selection of a file to be loaded. The default value for *bsoi\_common.wildcard* is \* on UNIX systems and \*.\* on others. By changing this into d18\*.\*, for example, you can specify that only files whose names start with "d18" should be shown.

# 3.2 Options for the "Recording"-Window

The options discussed in this section apply to the "Recording"-window. They are accessed through the Options|Other options menu in the "Recording"-window.

**X-axis width** Maximum length of the horizontal (time) axis of the ECG in milliseconds.  $(bsoi\_sreg.limit = 2048)$ 

#### $(0301_{3})(030$

# 3.3 Options for the "Channel"-Window

The options discussed in this section apply to the "Channel"-window, and some of its descendants. They are accessed through the Options|other options menu in the "Channel"-window.

- **X-axis width** The maximum length of the horizontal (time) axis of the ECG in milliseconds.  $(bsoi_chn.limit = 2048)$
- Auto baseline Whether or not to apply baseline correction when a baseline marker is dropped. ( $bsoi\_chn.autoBL = 0$ )
- **Enable x axis zoom** When the mouse is used to zoom, zoom in the horizontal direction. This is generally not desired. ( $bsoi_chn.zoomx = 0$ ).
- **Enable y axis zoom** When the mouse is used to zoom, zoom in the vertical direction.  $(bsoi\_chn.zoomy = 1)$ .
- **Range** Vertical range in millivolts. If the vertical range is fixed (see the next option), the vertical axes limits are, whenever they need to be reconsidered, set to the mean value of the lead plus or minus this amount. The range is determined as the maximum of the absolute values of the current limits, and can also be changed by zooming in or out. (*bsoi\_chn.range* = 3)
- **Vertical fit method** Determines how the vertical range for the ECG frame is chosen. Choices are "fixed" (uses the Range option), "fit global" and "fit local". ("global" refers to the entire loaded part of the file, "local" is the visible part)  $(bsoi\_chn.FitMethod = 2)$
- **Chop Spikes** If this option is on, the program tries to ignore spikes and other short artifacts when vertical limits must be chosen.  $(bsoi\_chn.ChopSpikes = 0)$
- **Sticky Markers** If this option is on, the markers will stick to the mouse pointer when clicked, until clicked again. If it is off, the markers only move with the mouse as long as the mouse button is pressed. ( $bsoi_chn.sticky = 0$ )
- Marker Linewidth Thickness of all markers except Point markers, in pixels. (bsoi\_chn.MarkerLineWidth = 3)

# $3.4 \quad \text{find } dV/dt \text{ options}$

These options apply to the activation detection algorithm. They are accessed through the Options|find dV/dt options menu in the "Channel"-window.

**min interval** Minimum interval between activations. The interval is specified in milliseconds.

 $(bsoi_fdt.tint = 40)$ 

- **Threshold** Minimum absolute value of the slope, for a negative deflection to be detected as an activation. This value is given in V/s.  $(bsoi_fdt.threshold = 3)$
- **spike width** Maximum width of spikes, used in the spike filter that is applied before the automatic activation detection routine.  $(bsoi_fdt.nspike = 5)$
- **Only begin to end** If true, activation detection is only applied between the Begin and End markers. This may speed up the computation significantly.  $(bsoi_fdt.b2e = 0)$

detect in ENDO channels The MAP<sub>L</sub>AB package knows about lead types. By default, it will try to find activations in endocardial leads, and it will not try to find them in, e.g., body surface leads. Set this option off if you don't want activation detection in endocardial leads. ( $bsoi_fdt.ut_endo = 0$ )

```
detect in INTRA channelsDetect in intracardiac leads. (bsoi\_fdt.ut\_intra = 1)detect in EPI channelsDetect in epicardiac leads. (bsoi\_fdt.ut\_epi = 1)detect in BSPM channelsDetect in body surface leads. (bsoi\_fdt.ut\_bspm = 0)detect in SUBC channelsDetect in subcutaneous leads. (bsoi\_fdt.ut\_subc = 0)detect in extremitiesDetect in extremity leads. (bsoi\_fdt.ut\_extra = 0)
```

# 3.5 Options for the Standard ECG

The following options apply to the "Standard ECG"-window, and are accessed through the **Options** menu in that window. The options are split in two options windows.

- **begin of interval** The beginning of the interval to be plotted, in milliseconds. This option is set to the position of the first Baseline marker when the options window is opened. The value is changed when a baseline marker is dragged; the new value will show up when a new options window is opened. ( $bsoi\_secg.begin = ...$ )
- **end of interval** The end of the interval to be plotted, in milliseconds. This option is set to the position of the last Baseline marker when the options window is opened. The value is changed when a baseline marker is dragged; the new value will show up when a new options window is opened. ( $bsoi\_secg.end = ...$ )
- **plot boxes** Plot a box around the ECG. ( $bsoi\_secg.box = 0$ )
- **plot baseline** Plot the baseline.  $(bsoi\_secg.baseline = 0)$
- fill interval Fill the interval between the first and the last point, and plot the baseline (it looks silly without a baseline).  $(bsoi\_secg.fill = 0)$
- **mark points** Mark the current begin and end samples in the ECG.  $(bsoi\_secg.markpoints = 0)$
- **aspect ratio** Aspect ratio of the ECG in millivolts per second.  $(bsoi\_secg.AspectRatio = 2.5)$
- width (cm) Maximum width of the entire plot in centimeters.  $(bsoi\_secg.width = 24)$
- height (cm) Maximum height of the entire plot in centimeters.  $(bsoi\_secg.height = 15)$
- **separation** Part of the plot that should be allocated to white space between the ECGs.  $(bsoi\_secg.sep = 0.1)$
- **stack tightly** Position the ECGs exactly with the specified separation. If this option is off, the ECGs are placed such that the maximum width and height are used, which means that the leads have a fixed position. This can be useful if multiple standard ECGs are printed for comparison. If a single standard ECG is needed, it is usually desired that the ECGs are placed close together. ( $bsoi\_secg.fit = 0$ )

**legend** Print reference lengths in the lower right corner. (*bsoi\_secg.legend* = 1)

- **paper speed** Horizontal scale in mm/s. The vertical scale is computed from the 'speed' and the aspect ratio. The default is to adapt the 'speed' to the available room iand the length of the ECG interval: (bsoi\_secg.speed = 1) The list of speed values in this pull-down menu is affected by the "paper speeds" option below.
- **name placement** How the lead names should be positioned. You can choose top inward, top outward, bottom inward, bottom outward, baseline, and 'none' (no labels at all). (*bsoi\_secq.NamePlacement* = 1)
- fontsize The fontsize in points of all texts in the "Standard ECG"-window. Sensible values are 8-12. (*bsoi\_secg.FontSize* = 8)
- window orientation Landscape or portrait orientation of the figure window can be chosen here.  $(bsoi\_secg.PaperOrientation = 1)$
- clip top at Maximum value for all ECGs, in millivolts. If this value is larger than the actual maximum, the actual maximum is used.  $(bsoi\_secg.max = \infty)$
- clip bottom at Minimum value for all ECGs, in millivolts. If this value is smaller than the actual minimum, the actual minimum is used.  $(bsoi\_secg.min = -\infty)$

#### paper speeds

 $(bsoi\_secg.SpeedList = `[1,5,10,25,50,100,500]')$ , in millimeter/second. This should be an array of speed values; it is used to generate the pulldown menu for the "paper speed" option above. This allows you to select your own favourite speeds.

#### legend width

 $(bsoi\_secg.LegendX = 500)$  length of the horizontal legend line in milliseconds.

**legend height**  $(bsoi\_secg.LegendY = 1)$  height of the vertical legend line in millivolts.

# 3.6 Options for Body Surface Maps

The following options apply to the "Map"-window:

- **map height** Maximum height of the map, in centimeters (on paper). (*bsoi\_map.ysize* = 8)
- **map type** The main types are pseudocolor and contour, but this menu provides many subtypes, that can be used for specific purposes:
  - contours (b) normal black and white contour plot, with the positive area filled grey. This is the default.
  - contours (bn) black and white contour plot without extrema values under it
  - **contours (bd)** black and white contour plot without grey and without zero line, especially for departure maps.

**map width** Maximum width of the map, in centimeters (on paper). (*bsoi\_map.xsize* = 8)

- **contours (bs)** black and white contour plot especially for very Small maps: no dashed lines, no extrema values under the map, no white area surrounding the zero contour.
- contours (bsx) like 'bs', with extrema values.
- coloured contours (c) contour lines with colours chosen as in pseudocolor maps.
- **pseudocolor (p)** normal pseudocolor map: positive red, negative blue, and zero line green.
- **pseudocolor** (**pn**) pseudocolor map without extrema values under it.
- **pseudocolor (pgn)** pseudocolor map without extrema values and without a green line.
- pseudocolor (pg) pseudocolor map with extrema values but without a green line.
- **pseudocolor (ps)** for very small pseudocolor maps: with green line but without extrema values.
- **pseudocolor database (pd)** for very small database maps in pseudocolor: one color on either side of the zero line, with a green line, without extrema values.
- filled contours (f) filled contours, using an asymmetric grey colormap, intended for maps of special data that has only positive values, but can also be used with signed data.

the letters between parentheses indicate the exact option given internally to the bsm function. ( $bsoi\_map.type = 1$ ) The option variable is a numeric index.

- **plot shoulders** Indicates whether the shoulders, sternum, and spine above the map should be drawn. (*bsoi\_map.shoulders* = 1)
- size of extrema markers The size of the  $\oplus$  and  $\ominus$  signs that indicate the maximum and minimum of the map, specified in 'map units' (interelectrode distances). Zero means no marker. Negative values indicate that the plus and minus signs should not be encircled. (*bsoi\_map.PlusMinSize* = 1.5)
- **positive part grey** Whether or not to paint the positive part of a contour map grey.  $(bsoi\_map.grey = 1)$
- **grid refinement factor** Resolution of the grid for pseudocolour maps, relative to the electrode grid. Higher resolutions print slower. (*bsoi\_map.fine* = 8)
- nr of contour levels Number of contour levels on each side of the zero line, for contour maps. (bsoi\_map.Nlevels = 7)
- **lead number** Lead number of a lead that is to be plotted below the map. The position of the lead in the map is indicated with a black dot, unless a negative number is used. The absolute value is the real lead number.  $(bsoi\_map.leadnr = [] (empty))$
- lead begin First sample of the lead. This value is changed when a baseline marker is dragged; the new value will show up when a new options window is opened. (bsoi\_map.LeadBegin = bs\_BaseLn(1) minus 50 milliseconds)
- lead end Last sample of the lead. This value is changed when a baseline marker is dragged; the new value will show up when a new options window is opened. (bsoi\_map.LeadEnd = bs\_BaseLn(2) plus 50 milliseconds)

- ecg box Position of the ECG below the map. This will be changed.  $(bsoi\_map.ecgbox = [2, 8, \emptyset, 5, 5]).$
- **plot polar map** Whether or not to print a QRS integral map-based localization in polar coordinates in the lower right corner of the window.  $(bsoi\_map.polar = 0)$

# 3.7 Options for the "Fast Map"-window

The following options apply to the "Fast Map"-window. They are accessed through the **Options** menu in the "Fast Map"-window.

grid refinement factor Grid refinement factor for the map. Higher values give finer maps, which take longer to print. ( $bsoi_fmap.fine = 2$ )

# 3.8 Options for the "Fast IntMap"-window

The following options apply to the "Fast IntMap"-window.

grid refinement factor Grid refinement factor for the map. Higher values give finer maps, which take longer to print. ( $bsoi_fimap.fine = 2$ )

# 3.9 Options for Activation Maps

For the "Activation Map"-window, there are three options windows. The following options are accessed through the **Options**|**other options** menu in that window.

- **isochrone interval** Contour interval in milliseconds.
- activation time limit Used by the flood fill algorithm. This is the maximum allowed activation time difference between neighbouring cells for inheritance of a state, in milliseconds.  $(bsoi\_tmap.ActLimit = 40)$
- do updates Update the activation map when a Begin or End marker in the "Channel"window is moved.
  (bsoi\_tmap.update = 1)
- **pseudocolor map** This option determines what kind of data is used for a pseudocolor map. Choices are:
  - activation time colors encode the activation time as indicated by the Point markers
  - **flood fill** all leads whose last activations seem to be connected are marked with the same color
  - **diff time** colors encode the time difference between the last activation in a lead and the mean of the last preceding activations in its neighbours
  - dV/dt at Point colors encode the value of dV/dt at the selected Point markers, i.e., for a different time in each lead.
  - $d\mathbf{V}/dt$  at Event colors encode the value of dV/dt at the current Event marker, i.e., at the same time for all leads.
  - V at Point colors encode the potential value at the selected Point markers

V at Event colors encode the potential value at the current Event marker, i.e., at the same time for all leads.

**bs\_tmap\_extern** data from the variable *bs\_tmap\_extern*.

white no data, but grey is used for areas and lines of block.

off no pseudocolor map at all. Block lines have the same colour as normal contours, and areas of block have the same colour as normal electrodes.

 $(bsoi\_tmap.colormode = 1)$ 

**texts** This option determines what kind of text is printed in the cells. Possibilities are:

milliseconds activation time(s) in milliseconds

samples activation time(s) in samples

grid numbers sequential number within the grid

lead numbers sequential number within the recording

**lead names** lead names (as in the "Channel"-window. For experts: the names from the —ElecName— array).

clabels no text in the cells, but contours will be labeled.

color data values of the data that is selected for the colormap (see above).

off no texts at all

 $(bsoi\_tmap.txtmode = 1)$ 

- time/sample nr What time is regarded as zero for printing activation times. Choices are: the preceding zero marker, the current Begin marker, or the first Point marker in the window.  $(bsoi\_tmap.timmode = 1)$
- isochrones This option chooses the algorithm used for isochrones. These algorithms are under development. "simple" uses the standard MATLAB function *contourc.* "isomap" uses a specialized function for isochrones, which is able to detect lines of block. It shows these with a special block contour, instead of drawing crowding contour lines. Block is considered present if the propagation speed in ms per unit distance is less than the activation time limit *bsoi\_tmap.ActLimit.* "tricky" uses *contourc* with a special trick to allow multiple activations in a lead within the time window. This enables crossing contour lines. (*bsoi\_tmap.contourmode* = 4)
- isochrone levels This option determines how the isochrone levels are chosen.  $(bsoi\_tmap.levelmode = 1)$
- **number of levels** The number of isochrone levels between the first and last activations in the current window, if the "isochrone levels" option is set at "fixed number".  $(bsoi\_tmap.Nlevels = 7)$
- show axes labels Show row and column labels next to the activation map.  $(bsoi\_tmap.vis = 1)$
- cell size The horizontal and vertical size of the cells (electrodes/channels) in the activation map. Changing this size is the only way to change the size of the "Activation Map"-window. This also affects the size of hardcopy. ( $bsoi\_tmap.px = 20$ )

- activation selection This option indicates on which Point marker the pseudocolor map and contours will be based. This can be set to the steepest deflection, the earliest, or one of the marker flags A1–A5. Point markers can be 'flagged' with the Flag menu in the "Channel"-window. Automatic flagging is also possible: see the manual on the "Channel"-window.  $(bsoi\_tmap.cs = 1)$
- **color scale** This option determines which data values are included in the color scale: "all" indicates all Point markers in the time window, "selected" indicates the selected Point markers in the time window (selection takes place with the "activation selection" option above), and "window" indicates the time window itself (from Begin marker to End marker). (*bsoi\_tmap.cscale* = 1).
- **show colorbar** This option determines if the colorbar on the right of the activation map is displayed.  $(bsoi\_tmap.colorbar = 0)$ .

The following options for the "Activation Map"-window are accessed through the Options|lead selection menu in that window.

show BSPM leads The MA<sup>P</sup><sub>L</sub>AB package knows about lead types. By default, it will not try to find activations in body surface leads, and it will not show body surface electrodes in activation maps. If you want to see them in the activation map, enable this option.  $(bsoi\_tmap.ut\_bspm = 0)$  The window size may change if room needs to be made for extra electrodes.

show ENDO leads	Show endocardial leads. $(bsoi\_tmap.ut\_endo = 1)$
show INTRA leads	Show intracardiac leads. $(bsoi\_tmap.ut\_intra = 1)$
show EPI leads Sh	how epicardiac leads. $(bsoi\_tmap.ut\_epi = 1)$
show SUBC leads	Show subcutaneous leads. $(bsoi\_tmap.ut\_subc = 0)$
show EXTR leads	Show extremity leads. $(bsoi\_tmap.ut\_extra = 0)$

The following options for the "Activation Map"-window are accessed through the Options|more options menu in that window.

- **begin of window** Determines how the beginning of the window is selected: either the current Begin marker (default) or the beginning of the recording. (The 'window' is the part of the recording where Point markers are seen) ( $bsoi\_tmap.winbegin = 1$ )
- end of window Determines how the end of the window is selected: either the current Begin marker (default) or the beginning of the recording. (The 'window' is the part of the recording where Point markers are seen) ( $bsoi\_tmap.winbegin = 1$ )
- **decimal places** Determines how fractional activation times (in milliseconds) are printed. The value 0, which is default on windoze and Unix platforms, means printing  $\frac{1}{2}$ ,  $\frac{1}{4}$ , and  $\frac{3}{4}$  signs, or no decimal places at all for other fractions. Other values indicate the number of places to use. The default value for the mac is 1.
- **contour level offset** Difference between selected contour levels and the levels actually fed to the contour routine (*contourc* or *isomap*). This can be used to prevent contours at data levels, which look strange when several adjacent electrodes have the same data level. The offset should be much smaller than one, and may be positive or negative. The value zero provides honest contours. (*bsoi\_tmap.cc\_offset = 0*).

# 3.10 Datafile (MCHD) options

The MCHD information that is read from datafiles is also used when saving datafiles. This information can be changed via the Options|MCHD|PDAT, Options|MCHD|RDAT, Options|MCHD|ENDO, and Options|MCHD|BSPM menus in the "Channel"-window. For more information, see the documentation in iff.web.

# 3.10.1 MCHD options

## Filename

The "MCHD Options"-window contains buttons that open option windows where data for the various MCHD chunks can be changed.

# 3.10.2 CONF options

- File Nr Optional file identification number. (*bsg\_mchd.conf.FileNr* is obtained from the datafile)
- **Elec Array** Grid identification code. This is used by the MAPLAB package to display the leads in the same configuration as the electrodes. (*bsg\_mchd.conf.ElecArray* is obtained from the datafile)

# 3.10.3 PDAT options

- **Patient** Patient name. (*bsg\_mchd.pdat.Name* is obtained from the datafile)
- **Sex** Patient's gender. (*bsg\_mchd.pdat.Sex* is obtained from the datafile)
- **Birthdate** Patient's date of birth. (*bsg\_mchd.pdat.BirthDate* is obtained from the datafile)
- **Diagnosis** To give a hint on what this file is about. (*bsg\_mchd.pdat.Diagnosis* is obtained from the datafile)

# 3.10.4 RDAT options

- **Nchannels** Number of channels or "leads" in the recording. (*bsg\_mchd.rdat.NChannels* is obtained from the datafile)
- **NSamples** Number of samples in the recording. (*bsg\_mchd.rdat.NSamples* is obtained from the datafile)
- **NBits** Number of bits to be used for values. (*bsg\_mchd.rdat.NBits* is obtained from the datafile)
- **Configuration** How the data is ordered in the file. (*bsg\_mchd.rdat.Config* is obtained from the datafile)
- **Compression** Compression type. (*bsg\_mchd.rdat.Compression* is obtained from the datafile)
- **Sample Time** Sampling interval in nanoseconds. (*bsg\_mchd.rdat.SampTime* is obtained from the datafile)

- **Unit Per Bit** The value associated with the lowest bit; i.e. the resolution. (*bsg\_mchd.rdat.UnitPerBit* is obtained from the datafile)
- **Unit** Name of the unit. (*bsg\_mchd.rdat.Unit* = 'mV')
- **Date** Recording date. (*bsg\_mchd.rdat.Date* is obtained from the datafile)
- **Time** Recording time. (*bsg\_mchd.rdat.Time* is obtained from the datafile)

# Chapter 4

# **Customization and Hacking**

"My dear Watson, try a little analysis yourself," said he, with a touch of impatience. "You know my methods. Apply them, and it will be instructive to compare results."

-CONAN DOYLE, The Sign of the Four

This chapter discusses how to extend MAP<sub>L</sub>AB, and how to suit it for your particular purposes. For example, it discusses some internal data structures, and it describes how to define new electrode configurations. It is assumed that the reader of this chapter has a moderate knowledge of MATLAB, such as can be obtained by reading the guide "Using MATLAB" [13].

# 4.1 variable naming conventions

- global variables have the prefix  $bs_{-}$
- global option variables have the prefix bso\_
- global variable struct arrays (such as used in locwin.web) have the prefix bsg.
- script and function m-files have the prefix bs\_, and their name has at most 8 characters
- hook scripts, which are also m-files, have the prefix *bsh\_*, and their name has at most 8 characters

# 4.2 Global variables in MAPLAB

All global variables of the  $MA_{L}^{P}AB$  package have the prefix  $bs_{-}$ . This should prevent the user from accidentally changing them. This section lists the most important gobal variables in the  $MA_{L}^{P}AB$  package. The following variables are the main containers for data:

bs\_Reg — the recording matrix: each row is a channel, each column is a sample.

*bs\_rejected* — the rejected vector: status of all channels.

These are for time instants and are closely connected with the *markers* in the "Channel"-window.

- bs\_Points sample values of all Point markers; this is a matrix with Nchannels rows; each row corresponds to a channel; zeros in this matrix indicate 'no marker'.
- bs\_PointType matrix with the same dimensions as bs\_Points; the elements contain the point type codes corresponding to the Point markers. The meaning of these codes is defined in tmap.web. These codes must be integers in the range 0-255.
- bs\_Events sample values of all Event markers
- bs\_Begins sample values of all Begin markers
- $bs\_Ends$  sample values of all End markers
- $bs\_BaseLn$  sample values of all Baseline markers
- bs\_Zeros sample values of all Zero markers
- bs\_CurPoint sample value of current Point marker
- bs\_CurEvent sample value of current Event marker
- bs\_CurBegin sample value of current Begin marker
- bs\_CurEnd sample value of current End marker
- $bs\_CurBL$  sample value of current Baseline marker
- bs\_CurZero sample value of current Zero marker
- *bs\_CurSample* sample value of current marker (the current instance of the current class)
- Other important variables are:
- bs\_CurChannel currently selected channel. This is the channel shown in the "Channel"window. Per-channel operations should by default operate on this channel.
- *bs\_ChanSelection* extended channel selection. Operations on a group of channels, such as a 'standard ecg' can use this set.
- bs\_ChanBlock a set of channels that are delimited by two rows and two columns. This variable is set by bs\_sreg if the user selects a block in the "Recording"-window. It is a struct, with the following elements:
  - raw: corners of the rbbox in the "Recording"-window
  - *leads*: numbers of selected leads
  - row1: first row
  - *col1*: first column
  - *Nrows*: number of selected rows
  - *Ncols*: number of selected columns
- bs\_filename name of the current datafile
- bs\_shortfilename bs\_filename without directory part
- *bs\_elconf\_names* names of the known nonstandard grids, as found in the *elconf* directories at startup.
  - Also of great importance are the variables defined in a grid definition file.

# 4.3 Grid definitions

When MAP<sub>L</sub>AB loads a file, it may infer from the data what grid is associated with it. Information about the grid may be built-in, or may be contained in a grid definition file. If the name of this file consists of the name of the grid as mentioned in the datafile, with extension .m, and the grid definition file can be found on MATLAB's search path, MAP<sub>L</sub>AB will use it automatically.

# 4.3.1 Organization of grid definition files

Several electrode grids are included in the distribution. These grids can be selected with the menu Options|Grids|Standard in the "Recording"-window.

It is recommended that the grid definition files are stored in one or more directories called **elconf**, that can be anywhere on MATLAB's search path. Files found there when MAPLAB starts up will be stored in the global variable *bs\_elconf\_names*, and will automatically be added to the menu Options|Grids|Other in the "Recording"-window. Directories can be added to the path using MATLAB's *addpath* function.

Alternatively, grid definition files can be applied from the command line. By itself, a grid definition file does not reset functions and re-create windows that depend on it, so if a grid definition is applied directly from the command line it may be necessary to close and re-open the "Recording"-window manually, for example. A better way to call a private grid is using the *bs\_ugrid* function; the call

```
bs_ugrid('foogrid')
```

where foogrid.m is the name of the grid definition file, which exists on the MATLAB path, handles everything that is needed for a new grid, as if the grid is selected from the grids menu in the "Recording"-window.

# 4.3.2 Grid definition file format

A grid definition file is an m-file script that defines an electrode configuration and sets the necessary variables to let  $MA_{L}^{P}AB$  work with it. A grid definition file should define the following variables:

- *ElecPos* Matrix containing the electrode index numbers in their grid positions. The value zero means 'no channel'.
- DefaultChannel The most-often used channel, like V1 in BSPM recordings.

*ExtraChannels* — Matrix that specifies commonly used linear combinations of leads.

ExtraChannelNames — Names of these 'extra channels'

NextraChannels — Number of extra channels.

*RefChannels* — Numbers of leads that are not part of the grid.

IgnoreChannels — Unused channels.

- *ElecName* Names of the electrodes. Each row of this matrix is a name.
- ElecGrid Grid index of the electrodes. This matrix has the same dimensions as ElecPos.

*ElecType* — Types of the electrodes. This matrix has the same dimensions as *ElecPos*. Possible electrode types are:

- 1. Body Surface
- 2. Endocardial
- 3. Intracardial
- 4. Epicardial
- 5. Subcutaneous
- 6. Extremities

For more information, see the documentation on the functions *bs\_sgrid* and *bs\_ugrid* in the MAP<sub>L</sub>AB sourcebook.

If any of these is not present, it will be left empty. After defining these variables, the file should call the script  $bs\_sgrid$ , which takes care of internal MA<sup>P</sup><sub>L</sub>AB variables that deal with the electrode configuration. An example grid definition is given in the file facegrid.m. This file is shown in figure 4.1.

# 4.4 Hook files

Hook files are m-file scripts that allow you to let MATLAB perform certain actions at certain events. These scripts are executed by MAPLAB functions, and by default contain little or no code. By adding code to them, you can arrange that, for example, a program of your own creation is executed whenever a marker in the "Channel"-window is dropped. It is recommended that the *bs\_hook* function is used to add code to hook files (see hooks.web). The following tables gives an overview of the hook files that are available:

hook script	where called	document	when called
bsh_bl.m	callback	channel.web	at baseline correction
bsh_chc.m	bs_chn.m	channel.web	at change of current channel
bsh_map .m	bs_map.m	maps.web	when a map is plotted for hardcopy
bsh_drop.m	bs_mdrop.m	channel.web	when a marker is dropped
bsh_rst.m	bs_rstat.m	channel.web	at change of rejected state

# 4.5 Reading and changing the sources

Source documentation of MAPLB and MAPLAB is available in PDF format at

```
http://gnu.amc.uva.nl.
```

The source files themselves are provided in the distribution, in the web or src directory. Sources come in several languages:

- plain m-files. These will gradually disappear; all m-files will eventually be documented in MWEB format.
- plain C files and .h files. These will gradually disappear; all C programs will eventually be documented in CWEB format.
- CWEB files [8] are used to document sources in the C language. To create files that the compiler understands, the **ctangle** program is needed. Contact us for more information.

```
ElecPos = [
     0 \quad 0 \quad 35 \quad 36 \quad 01 \quad 02 \quad 03 \quad 04 \quad 05 \quad 06 \quad 07 \quad 0
                                         0
     0 34 0
                0 0 0 0 0
              0
                               0 0 08
                                         0
     0 33 0 37 38 0 0 44 41
                                0 0 09
                                         0
           0 40 39
                       0 43 42
    32 0
                    0
                                0
                                   0 10
                                         0
                   0
                      0
                         0
    31 0 0
              0
                0
                             0
                                0
                                  0 0 11
    30 0
           0
              0
                0
                   0
                       0
                          0
                             0
                               0
                                   0 0 12
    29
           0
              0
                0 0 0 0
                             0 0 0 0 13
       0
    28 0 45 0 0 0 0 0 0 53 0 0 14
    27 0 0 46 0 0 0 0 0 52 0 15
                                        0
     0 26 0 0 47 48 49 50 51 0 0 16 0
     0 0 25 0 0 0 0 0 0 0 17 0
                                         0
     0 0 0 24 23 22 21 20 19 18 0 0 0];
ElecName = ' '; % not an empty string!
for i = 1:53
   ElecName(i,1:3) = sprintf('E%2d', i);
end
IgnoreChannels = [54:bs_Nchannels];
ElecGrid = ones(size(ElecPos));
ElecType = ones(size(ElecPos));
for i = 1:36
   e = find(ElecPos==i);
   ElecGrid(e) = 1;
                          % grid 1: outline
   ElecType(e) = 1;
                          % outline type
end
for i = 37:40
   e = find(ElecPos==i);
   ElecGrid(e) = 2;
                          % grid 2: left eye
   ElecType(e) = 2;
                          % eye type
end
for i = 41:44
   e = find(ElecPos==i);
   ElecGrid(e) = 3;
                          % grid 3: right eye
   ElecType(e) = 2;
                          % eye type
end
for i = 45:53
   e = find(ElecPos==i);
   ElecGrid(e) = 4;
                          % grid 4: mouth
   ElecType(e) = 3;
                          % mouth type
end
bs_sgrid;
```

Figure 4.1: The facegrid.m file (see text).

- MWEB files [18] are used to document m-files. To create files that the compiler understands, the mtangle program is needed. This program is only available with us, but we give it away for free.
- nuweb files [1] are used to document files in other languages, such as parsers written in *lex* and *yacc* [9].

# Chapter 5

# A Short History of MAPLAB

"Mine is a long and a sad tale!" said the Mouse, turning to Alice, and sighing. "It is a long tail, certainly," said Alice, looking down with wonder at the Mouse's tail; "but why do you call it sad?"

-LEWIS CARROLL, Alice's Adventures in Wonderland

 $MA_L^PAB$  was born in 1993, when Mark Potse decided to use MATLAB for the analysis of Body Surface Mapping data. Version 4 of MATLAB was then just available, which was the first version with sophisticated graphics and graphical user interface capabilities. Created in the first place as a personal tool set, it soon became a coherent software package. André Linnenbank pointed out that it should be given a name, and the name "MAPLAB" was coined shortly afterwards. The name stands for Map Laboratory, just like MATLAB stands for Matrix Laboratory, and the strange formatting was modeled after the logo of  $LAT_FX$ , the text processing system that was used to typeset the documentation.

The first official documentation was printed in July 1994[17]. During 1994 and 1995, Auke Latour became the first user and co-author. With his help, the package, which worked only on UNIX computers, was made suitable to run also under the MS Windows version of MATLAB.

After André Linnenbank got his PhD, and Mark Potse got his MSc degree, they both happened to be employed in a new Body Surface Mapping project. They continued working on the package, much improving the user interface, using the new functionality offered by MATLAB version 5, and adding support for long recordings. By that time, Mark migrated from the Sun to a new PC running Linux, and it became possible to export filesystems in order to share files between UNIX and Windows systems so that both could be used for development. A revision control system was employed to bring some order in the chaos of changes, and a real multi-programmer development system was set up.

After prof. Van Hemel asked for a tool to view multichannel endocardial ECG's and to indicate activation times in them, it was decided to extend  $MA_L^PAB$  for this purpose, and support for endocardial data was added, including an activation detection algorithm, a better interface for ECG handling, and a tool to display activation maps. New hardware capable of recording combined body surface map and endocardial maps necessitated the handling of both types of data at the same time.

Because of MAP<sub>L</sub>AB's new capabilities to handle endocardial data, researchers from the Experimental Cardiology department became interested, and their habit of inventing a new electrode grid for almost every new experiment made it necessary to thoroughly

parameterize MAP<sub>L</sub>AB's grid handling. The availability of *users*, on the other hand, made it possible to find and fix hundreds of bugs in the package.

A publication at the IEEE-EMBS conference in Hong Kong, 1998 [20], generated interest from researchers abroad. Due to these contacts, and cooperation that existed earlier, in the year 2000 the software is in use in Amsterdam, Utrecht, Nieuwegein, San Francisco, Bordeaux, Münster, Tampere, Tessaloniki, and Graz.

Several new algorithms, such as for activation detection, VT localization, and isochrone drawing have been tested with and made generally available through the  $MA_{L}^{PAB}$  package.

The next big step in the development of  $MAP_LAB$  will be the incorporation of tools that are capable of handling three-dimensional electrode grids[4, 10]. This will include a completely new grid definition system, as well as important changes to the map-displaying tools.

# Chapter 6 Changes to MA<sup>P</sup>LiB and MA<sup>P</sup>LAB

Change begets change. Nothing propagates so fast.

-CHARLES DICKENS, Martin Chuzzlewit

This chapter describes all changes to MAPAB, starting at version 1.37.

- **264** (1/2/99) Made transforms in **rconvert** more efficient.
- **265** (3/2/99) Made output difforder adjustable ("huff.DiffOrder") in rconvert.
- **266** (3/2/99) Fixed memory leak in rconvert.
- **267** (4/2/99) Used the new *readdir* mex function for the 'next' and 'prev' file buttons in the "MapLab"-window; this should make them work under windoze.
- **268** (5/2/99) Reduced debugging output for several functions.
- **269** (5/2/99) Bug fix in *eplot* that caused errors in the "Localize"-window.
- **270** (5/2/99) Choose a default database in the "Localize"-window if the current database becomes invalid due to a change of view.
- **271** (6/2/99) Added some forgotten globals to declarations in maplab.m.
- **272** (9/2/99) Removed simple variable names in some scripts.
- **273** (19/2/99) Bug fixes in texts for the "ActMap"-window.
- **274** (19/2/99) Write errors in log file.
- **275** (19/2/99) Case-insensitive compare strings for windoze in *readdir*, to make the 'next' and 'prev' file buttons work.
- **276** (9/2/99) Removed simple variable names from scripts (MAPLAB programs that don't have their own namespaces should always use variables with underscores in them, to prevent clashes with user variables).
- **277** (5/3/99) Added the *laplace* function for computation of weights for computation of a surface Laplacian.

- **278** (9/3/99) Made clockwise order (instead of numeric) when computing *raster\_neighbours* in *bs\_sgrid*, to make it usable by the *laplace* function.
- **279** (17/3/99) Fix for Matlab-5 compatibility in getmap.
- **280** (17/3/99) In *bs\_load*, remove trailing semicolons from grid names of endocardial files, before feeding them to *bs\_ugrid*.
- **281** (1/4/99) Use  $bs\_LapW$ , now only generated for Constellation Baskets by cblap, to compute the Laplacian lead in  $bs\_lead$ .
- **282** (1/4/99) Added the *laplace* function.
- **283** (1/4/99) Added the *bs\_note* and *bs\_dial* functions.
- **284** (2/4/99) Added the *bs\_panic* and *bs\_panf* functions.
- **285** (6/4/99) Made *bs\_log* capable to write a string to the log file.
- **286** (6/4/99) Log file loading and use of Grid menu; submit log file in *bs\_panic*.
- **287** (7/4/99) Give times in "Map"-window in milliseconds as well as samples.
- **288** (7/4/99) Put also name of m-file in *bs\_panf*.
- **289** (7/4/99) Bug fix in next/prev file buttons in the "MapLab"-window that occurred if a metafile was loaded.
- **290** (7/4/99) The "Spy"-window now shows values for all leads in the Selection (the leads that show up in the "Channel"-window).
- **291** (7/4/99) The *rarith* function has got an alternative syntax that allows for extra optional arguments.
- **292** (9/4/99) ECG position is now reset correctly after vertical dragging if the vertical fit method in the "Channel"-window is "fixed".
- **293** (9/4/99) The file selection window for saving metafiles now proposes the directory where the datafile comes from.
- **294** (9/4/99) New option "Marker Linewidth" for the "Channel"-window replaces the fixed *bs\_MarkerLinewidth* variable.
- **295** (9/4/99) Default localization algorithm is now set correctly in the "Localize"-window.
- **296** (9/4/99) The MeanAbs (MA) signal in the "Channel"-window is computed using only surface leads, if any are present, and over all leads otherwise. It is also made faster, using the new capabilities of the *rarith* function.
- **297** (9/4/99) Made SumAbsDiff lead in the "Channel"-window faster.
- **298** (12/4/99) ECG offsets in the "Channel"-window can now be set by the user even if the "vertical fit method" is "fix". They are not automatically adapted if the vertical range is changed. This allows faster display of the ECGs without corrupting the vertical position.
- **299** (12/4/99) The Laplacian signal is now inverted and divided by the number of neighbours so that it can easier be interpreted as a measure of the membrane current.

- **300** (21/4/99) Split the options window for the "Activation Map"-window.
- **301** (21/4/99) Added another options window for the "Activation Map"-window, with an option to select the kind of begin for the activation window: Begin marker, or begin of recording, and an analogous option for the end of the window.
- **302** (22/4/99) All transient windows (Options, Warnings, ...) get a special background colour (kind of yellow).
- **303** (23/4/99) Implemented the "Filter"-window, a user interface to filters, containing moving average filter, mains interference filters, and spike filter.
- **304** (24/4/99) The 'q' key, pressed in any options window, will make the window quit (like the "cancel" button), just like in most other MA<sup>P</sup><sub>L</sub>AB windows.
- **305** (27/4/99) Sanitized the code for resizing the "Activation Map"-window: resizing is made impossible instead of ignored. This makes refreshing and printing a little faster.
- **306** (28/4/99) dV/dt values at Point and Event markers can now be shown in the "Activation Map"-window.
- **307** (28/4/99) The colour of rejected channels in the "Activation Map"-window is now computed by interpolating the largest group of neighbours that can, according to the "activation time limit" option, belong to the same front. (it used to be interpolated from all neighbours, which gives wrong results in case of colliding fronts)
- **308** (28/4/99) Hackers only: *bs\_ActMap* now always contains Not-A-Numbers at positions where no electrodes or no data is present.
- **309** (29/4/99) Added maps for dV/dt at Event, and V at Point and Event to the "Activation Map"-window.
- **310** (29/4/99) Fixed a bug in the call to *ffill*, for flood fill maps in the "Activation Map"-window.
- **311** (3/5/99) Bug fixes in the "Activation Map"-window: remove menus when freezing, don't remove contours when freezing, and call also the new options functions at initialization.
- **312** (3/5/99) Reset X (time) tick marks in the "Channel"-window when applying the Channel Window Options.
- **313** (4/5/99) Built-in the needle-105 grid.
- **314** (4/5/99) Made "Recording"-window more robust to incorrect grid setting.
- **315** (4/5/99) Take care of leads without valid neighbours when interpolating rejected channels in the "Activation Map"-window.
- **316** (5/5/99) Bug fix for block selection in the "Recording"-window.
- 317 (5/5/99) Set font sizes correctly in "Recording"-window and "MapLab"-window.

**318** (5/5/99) Better progress indicator and a slightly better progress for "find dV/dt in all channels".

## version 1.39

- **319** (5/5/99) Guess "naald105" grid if the number of channels is 222.
- **320** (10/5/99) Made "dV/dt at Event" and "V at Event" maps in the "Activation Map"-window show all channels if there are no Point markers; if there are any Point markers, they will show only channels with valid Points.
- **321** (14/5/99) Added "11x11" epicardial grid; made it the default for recordings with 121 channels.
- **322** (14/5/99) Used *bs\_busy()* in code for "Recording"-window and "Standard ECG"-window, to display a watch pointer or even a progress indicator if redrawing takes long.
- **323** (20/5/99) Attempt to work with incorrect grid definition with not too many error messages.
- **324** (9/6/99) Bug fix in legend placement for the "Standard ECG"-window.
- **325** (9/6/99) New options window for the "Standard ECG"-window, with adjustable paper speed list, legend width, and legend height.

- **326** (10/6/99) Oops, call new option scripts for initialization in the code for "Standard ECG"-window.
- **327** (10/6/99) Added forgotten option variable *bso\_chn\_b2e* to the globals, to make it work with option saving.
- **328** (11/6/99) Reduced terminal output for *bs\_secg*; changed the header format for the "Standard ECG"-window to show also the metafile name if a metafile was loaded.
- **329** (11/6/99) Bug fix in options saving code, for non-UNIX platforms.
- **330** (11/6/99) Set fontsizes correctly in the "MapLab"-window.
- **331** (14/6/99) Keep existing Point markers if "find dV/dt" button is pressed, and  $bs\_Points$  has not enough columns; this is necessary if points are loaded from a metafile that is not generated by MAP<sub>L</sub>AB.
- **332** (14/6/99) Fixed off-by-one bug in activation detection.
- **333** (14/6/99) Set fontsizes correctly in the "Activation Map"-window.
- **334** (14/6/99) At startup, MAPLAB will look for directories named "elconf" in the MATLAB path. M-files found there will be shown as electrode configurations in the menu Options|Grids|Other in the "Recording"-window.
- **335** (15/6/99) Made activation detection work again even if the number of defined leads differs from the number of actual leads.

- **336** (17/6/99) Prevented crash in "Channel"-window when there is no Begin marker at startup.
- **337** (17/6/99) Bug fix in balloon64 grid.
- **338** (18/6/99) Another bug fix in balloon64 grid.
- **339** (18/6/99) Guess ballon64 grid in case of 65, 66, 67, 68, and 69 channels.
- 340 (21/6/99) Removed incorrect warning about NRPART parameter from *rload*.
- **341** (6/8/99) Fixed bug for non-native byteorder rewriting of datafiles.
- **342** (18/8/99) Bug fix in ECG-redrawing in aligned mode.
- **343** (19/8/99) Several bug fixes in signal-alignment/averaging code; it seems to work now.

- 344 (26/8/99) Bug fixes and improvements in code for special channels SAD and DDSA; these are now compatible with the signals in the paper on QRS detection by Kemmelings [7].
- **345** (27/8/99) Added wrappers to some callback functions in order to catch more unforeseen errors.
- **346** (30/8/99) Bug fixes in error reporting functions.
- **347** (30/8/99) Decent error message if the 'goto' button is pressed while a nonexistent marker is selected.
- **348** (3/9/99) Implemented matrix convolution and signal-averaged convolution in MAP<sub>L</sub>iB function *rarith*.
- **349** (4/9/99) Made computation of the SAD signal (which can be used for QRS detection) in the "Channel"-window 3 times faster, and the SADD signal 8 times faster. For both, the recording isn't copied anymore, so that they require much less memory in case of large recordings. Put a noise filter in the code for SADD, and magnified it by 4 to give it the same order of magnitude as BSM leads.
- **350** (9/9/99) Initialize "Channel"-window at minimal width (632 pixels) on screens with 640 pixels or less.
- **351** (15/9/99) Handle missing internal datafiles gracefully.
- **352** (22/9/99) Bug fix in "Integral Map"-window.
- **353** (24/9/99) Let *rload* return error if the recording number is larger than one in an ATTL or IFF-MCHD file.
- **354** (5/10/99) Accept wrong number of channels in 2-spoons grid, just like in any other grid.
- **355** (6/10/99) Removed Tooltips from "Channel"-window, to allow working with MATLAB version 5.1

## version 1.41.68

- **356** (7/10/99) Changes in *rload* to recognize files created by Van Dessels raw2epwb, and to correct the bit step.
- **357** (8/10/99) Added Laplacian transformation in the Actions—Transforms menu of the "Channel"-window.
- **358** (8/10/99) Made genintrp less verbose.
- **359** (8/10/99) Show complete version number at startup.
- **360** (8/10/99) Set all font sizes in the "Activation Map"-window depending on the *bso\_fontsize* option in the Maplab Options.
- **361** (12/10/99) Improved handling of gridnames in endo files: less "unknown grid" errors!
- **362** (13/10/99) Remember contour interval option.
- **363** (13/10/99) Several fixes for bugs detected by MetroWerks compiler for the Power-Mac. Also added many predeclarations to prevent warnings. Made *bs\_log* do nothing on the mac, until we know how to do it.
- **364** (14/10/99) Improved error messages from *rload* in case of damaged endo files.
- **365** (15/10/99) More information in the "Activation Map"-window.

## version 1.42.0

- **366** (15/10/99) Fix in code for *getmap* for Borland compiler.
- **367** (16/10/99) Bug fixes in check for the Dessel bug.
- **368** (16/10/99) Reduced screen output from the grid-selection code.
- **369** (18/10/99) Even less "unknown grid" errors.
- **370** (18/10/99) Corrected title position in "Activation Map"-window.
- **371** (18/10/99) Removed some debugging output for  $bs\_tmap$ .
- 372 (10/11/99) Many, many fixes to make MAPLAB work on a Power Macintosh as well.
- **373** (10/11/99) Fixed bugs in code for reading endo2 files on big-endian machines.
- **374** (10/11/99) Set *RDAT\_Time* empty when reading an endo2 file.
- **375** (10/11/99) Recognize ExpCard gridnames again, after last october's changes to the *rload* function.
- **376** (11/11/99) Correct byte-swapping for signed shorts in order to read endo2 files on big-endian machines (Macintosh!).
- **377** (15/11/99) Bug fixes in metafile writing for the Macintosh.
- **378** (15/11/99) Bug fix in drawing the rejected-channel cross in the "Recording"-window that became evident on the Mac.

**379** (15/11/99) New option "decimal places" for the activation map, to control the printing of activation times. Default value for Windoze and Unix is 0, which means print  $\frac{1}{2}$ ,  $\frac{1}{4}$ , and  $\frac{3}{4}$  signs if possible; default value for the mac is 1, which means printing 0.5, 0.2, and 0.7 in these cases.

#### version 1.43.0

- **380** (16/11/99) Bug fix in metafile reading code that became evident on the mac.
- **381** (18/11/99) Fixed new bug in "Standard ECG"-window for filenames with backslashes. Also, the filename is now shown with the correct file separator.
- **382** (18/11/99) Show dV/dt limit in title of activation map.
- **383** (23/11/99) Bug fixes in *mload* for big-endian machines.
- **384** (23/11/99) Another bug fix in *mload*.
- **385** (24/11/99) Another bug fix in *mload* for big-endian machines.
- **386** (24/11/99) Bug fix in *bs\_mdrop* to handle selection of a nonexistent marker in the "Channel"-window gracefully.
- **387** (21/01/2000) External data can be shown in "ActMap"-window using *bs\_tmap\_extern*.
- **388** (25/01/2000) André started work on derksen.web.
- **389** (10/02/2000) New menus and keys in the "Channel"-window for interchanging Begin/Event or Event/End markers. This facilitates switching between QRS, QRST, and ST integral maps. The new menus and keys, as well as the existing menus for shifting Begin and End, will automatically invoke the slaves.
- **390** (14/04/2000) Incorrect number of channels in ballon64 grid is now handled gracefully.
- **391** (26/04/2000) and earlier: bug fixes in  $13 \times 16$  and  $11 \times 11$  grids; added  $16 \times 13$  grid as nonstandard to the distribution.
- **392** (26/04/2000) more bug fixes in generic grid stuff, probably only important for experimental code.
- **393** (2/5/2000) made updates of the "Activation Map"-window faster.
- **394** (12/5/2000) implemented the new *isomap* function in MAPLAB, also tested and debugged for Windows. Available via the Isochrones menu in the options for the "Activation Map"-window.
- **395** (12/5/2000) more efficiency improvements for updates of the "Activation Map"-window, now tested and debugged for Windows.

- **396** (15/5/2000) New text option "off" for "Activation Map"-window.
- **397** (15/5/2000) New colormap option "white" for "Activation Map"-window shows regions and lines of block with grey, and all other areas with white.

- **398** (15/5/2000) excluded web files from small distribution zip file, in order to keep it fitting on a single floppy.
- **399** (16/5/2000) Added an optional colorbar for the "Activation Map"-window, with a switch in the options window.

#### version 1.44.15

- 400 (17/5/2000) Replaced new code that required at least Matlab version 5.2 to work; all of MAP<sub>L</sub>AB should work in version 5.1.
- **401** (17/5/2000) Fixes for block contours in triangular grids in *isomap*.
- **402** (18/5/2000) Options for the "Activation Map"-window are remembered again (if "Save Options" is used).
- **403** (18/5/2000) Added Sippens IMI database positions for LV pseudo-3D view in the "Localize"-window.
- 404 (18/5/2000) Added support for Maastricht datafile format.
- **405** (18/5/2000) Made *isomap* work in "Activation Map"-window with mixed-type grids (such as BSM+Basket).
- 406 (18/5/2000) Use 30 colours for activation map (used to be 45, then 15).

## version 1.44.34

- 407 (22/5/2000) Fixed new bug in options saving code.
- 408 (22/5/2000) Much work on *isomap* in strange circumstances.
- 409 (23/5/2000) New DefBas slave released.
- **410** (23/5/2000) Prevented gruesome *isomap* error that occurred in case no or very few activations were present.
- 411 (23/5/2000) Added button for manual contour labeling in "Activation Map"-window.
- 412 (23/5/2000) MAP<sub>L</sub>AB no longer rings the bell for "notices" (it does for warnings and errors).
- **413** (23/5/2000) Added *cc\_offset* (Contour level offset) option for "Activation Map"-window.
- **414** (23/5/2000) Put all options in the Index of the user manual, referring to the appropriate page in the Options chapter.
- **415** (23/5/2000) Bug fix in *isomap* for block in new edges.
- **416** (23/5/2000) Button for manual contour labelling in the "Activation Map"-window.
- 417 (24/5/2000) Many more fixes, particularly for triangular grids, in *isomap*.
- **418** (25/5/2000) Fixed bug in *to3dgrid* that caused errors in contours created by *isomap* in the "Activation Map"-window.

- **419** (25/5/2000) Fixed bugs in new code for the "clabel" button in the "Activation Map"-window.
- **420** (25/5/2000) Bug fix for saving markers in case the number of channels in the datafile was set wrong.
- 421 (25/5/2000) A few more fixes for *isomap*.
- **422** (16/6/2000) Several fixes in *isomap* routine.
- **423** (28/8/2000) Don't use togglebutton yet; not everybody has it.
- 424 (28/8/2000) Fixed Macintosh-specific problems in elconf directory scanning.
- **425** (12/12/2000) Do not execute, only syntax check elconf files at startup. This prevents problems with elconf files that depend on information on the datafile, like the number of channels.
- **426** (9/4/2001) Update tickmarks in "Channel"-window when zooming in or out and in all other circumstances.
- **427** (24/4/2001) Automatically jump to current marker (like the "goto" button does) when loading a metafile.

## version 1.44.136

- **428** (24/4/2001) No longer automatically pop up the "Recording"-window if a new file is loaded; only update it if it exists. (This was an accidental feature.)
- **429** (24/5/2001) Bug fix in the "Fast Map"-window and "Fast intmap"-window: start up correctly even if the map is flat (all channels the same value).
- **430** (24/5/2001) Bug fix in the "fast map" slave (for body surface maps); the integral map is now correctly computed also if there is one sample between the begin and end markers.
- **431** (24/5/2001) Introduced bs\_import script for datafile loading.
- 432 (5/2001) Various new features in rconvert.
- **433** (5/6/2001) Bug fix in rconvert that prevented header skipping in endo files.

# version 1.44.147

- **434** (9/8/2001) Search for datafile first in current directory if file not found in metafilepath.
- **435** (3/10/2001) Bug fix for Matlab v6. More v6 bugs exist.

# version 1.44.149

**436** (5/11/2001) Removed syntax check of elconf files at startup.

- 437 (11/12/2001) Button for easy replacement of metafile in the "save"-window.
- **438** (19/12/2001) Many internal changes for Matlab v6 compatibility. The user shouldn't notice, except that it should now work with version 6.
- **439** (16/01/2002) Added "lead names" as a text mode for the "Activation Map"-window, to facilitate inspection of *ElecName* settings in grid definition files.
- **440** (21/01/2002) Added "Histogram"-window, intended to replace the other two histogram windows. It now incorporates the functionality of the Activation Time histogram, as well as the new MaxPhaseDif histogram (which is defined on the faces of the grid).
- **441** (21/01/2002) The "Activation Map"-window can now also show data that is defined on the faces of the grid, instead of the vertices. At this time, face data can only be shown with texts, and two data types are defined on the faces: face numbers and MaxPhaseDif.

#### version 1.46

- 442 (23/01/2002) Several improvements to "Histogram"-window. Optimized the code for MaxPhaseDif computation (used for both activation map and histogram).
- 443 (23/01/2002) Automatically clean up hookfiles when MAPLAB starts. This should prevent some strange errors that occur after a crash, due to remaining hooks.
- 444 (23/01/2002) Several minor bugs fixed in code for "Activation Map"-window and "Histogram"-window.
- 445 (24/01/2002) Added error checking in code for hookfile handling.
- **446** (24/01/2002) Always pop up the "Recording"-window the first time a file is loaded in a session. If the user deletes it, it is not opened again when another file is loaded. If it remains, it will be updated each time a file is loaded.
- 447 (24/01/2002) Made updates of "Activation Map"-window slightly faster.
- **448** (24/01/2002) Made Options windows more compact, and modified colours to make them easier on the eye.
- 449 (25/01/2002) Fixed bug in default setting for options of the "Activation Map"-window.
- 450 (28/01/2002) Added option checkbox for debug mode in MAPLAB's Options window. Improved presentation of options.
- 451 (28/01/2002) Check for missing mex (dll) files.
- **452** (29/01/2002) Allow spaces in the name of the options file.

## version 1.47

**453** (30/01/2002) Hackers only: moved option variables of *bs\_chn* and *finddvdt* into two new structs: *bsoi\_chn* and *bsoi\_fdt*, respectively.

- **454** (31/01/2002) Hackers only: moved option variables of *bs\_secg* into struct *bsoi\_secg*, except for *bso\_secg\_Points*, *bso\_secg\_names*, and *bso\_secg\_nrs* because they are part of the documented user-interface.
- **455** (31/01/2002) Bug fixes in code for the Fast Map window: startup without panic even if the Event marker is absent, and display the correct sample in the header.
- **456** (31/01/2002) Hackers only: moved option variables of *bs\_map* into struct *bso\_secg*. These options dated from a very early version of MAPLAB and had nonsystematic names: *bso\_Fine*, *bso\_Grey*, *bso\_Leadnr*, *bso\_MapXSize*, *bso\_MapYSize*, *bso\_Nlevels*, *bso\_PlusMinSize*, *bso\_Shoulders*, *bso\_Type*, *bso\_map\_ecgbox*, *bso\_Lead\_Begin*, *bso\_Lead\_End*, and *bso\_Polar*.
- **457** (31/01/2002) Hackers only: moved all variables with prefix *bs\_chn\_* into new global struct *bsg\_chn*.
- **458** (01/02/2002) Hackers only: Initialization work that was formerly done by the maplab.m script is now placed in the new function *bs\_init*. Temporary variables used in this code will therefore no longer appear in the base workspace. This makes a better overview of the workspace possible.
- **459** (01/02/2002) Hackers only: Several new parameters for the *bsm* function.
- **460** (01/02/2002) Made all body surface maps clickable; if an electrode position is hit, the corresponding channel will be selected (in the "Channel"-window, "Recording"-window, "Histogram"-window, etc.).
- **461** (01/02/2002) Fixed a bug in x-axis tick labels of the "Histogram"-window, and one in updates. The latter manifested itself especially when the number of bins decreased.
- **462** (03/02/2002) Hackers only: global variables for  $bs\_busy$ ,  $bs\_sreg$  and  $bs\_listb$  moved into structs.
- 463 (03/02/2002) Bug fix in "Channel"-window for marker deletion.
- 464 (03/02/2002) Bug fixes in "Histogram"-window.
- 465 (04/02/2002) Bug fixes in the listfile browser.

#### version 1.47.66

- **466** (05/02/30) Fixed recent bug that manifested itself when a channel's rejected status was changed before the first time the "Channel"-window appeared.
- **467** (05/02/2002) Temporary "omit zero" (omz) option implemented in "Histogram"-window.
- **468** (06/02/2002) Hackers only: New interface made for the creation of options windows, consisting of the *bs\_optwu* and *bs\_opta* functions.
- 469 (06/02/2002) Revised the old "Movie"-window, and added an options window for it.
- **470** (06/02/2002) Fixed minor bugs in handling of the "omit zero" option in "Histogram"-window.

- 471 (07/02/2002) Fixed bug in printing negative and near-zero times in the "Activation Map"-window.
- **472** (08/02/2002) Updates of the "Fast Map"-window and "Fast IntMap"-window (body surface mapping only) are now much faster, and on modern computers (500 MHz or more) fast enough for movie-like behaviour. The > and < keyboard keys in the "Channel"-window have been adapted to update these windows, so that the automatic repeat function of the keyboard can be used for constant movement.
- **473** (08/02/2002) Database localization removed from "Fast IntMap"-window, and removed "Polar Map"-window; their functionality is now taken over by the Localizer windows. "Fast Map"-window and "Fast IntMap"-window can now be made smaller.
- 474 (11/02/2002) Hackers only: lots of global variables moved into global structs. Check the MAP<sub>L</sub>AB source files if your add-on scripts fail!
- 475 (13/02/2002) Hackers only: Variables that are dumped in the workspace by the *rload* function (such as *rload\_total*) are no longer global in MAP<sub>L</sub>AB, but local to the *bs\_load* function. They are no longer used in bs\_import.m. The values of all variables dumped by *rload* are available in the *bsg\_mchd* struct.
- **476** (13/02/2002) The variables with prefixes *RDAT\_*, *PDAT\_*, and *CONF\_* are replaced by a single struct array *bsg\_mchd* with fields *rdat*, *pdat*, and *conf*, which are in turn struct arrays. This affects the **bs\_import** script. If you use a customized version of this script (to load local file formats) you may have to adapt it. Ask the local guru.

#### version 1.48.0

- 477 (14/02/2002) Hackers only: Avoided putting temporary variables in the base workspace as much as possible. If such variables are used in MA<sup>P</sup><sub>L</sub>AB, their name ends with an underscore character.
- 478 (15/02/2002) Hackers only: grid definition variables like *ElecPos* are no longer global in MAP<sub>L</sub>AB; they are kept local to *bs\_ugrid*, which executes the grid definition script. These variables are now available as fields of *bs\_ElConf*. The *defghi* and *bs\_sgrid* scripts were modified to make this possible. There is a small chance that this breaks existing electrode definition scripts.
- **479** (15/02/2002) Hackers only: MA<sup>P</sup><sub>L</sub>AB's pre-digested grid variables *raster\_row* etc. are now available as *bs\_ElConf.rrow*, *bs\_ElConf.rcol*, *bs\_ElConf.rgrid*, *bs\_ElConf.rtype*, and *bs\_ElConf.rneighbours*.
- **480** (15/02/2002) Hackers only:  $bs\_ColourOpt$  and  $bs\_FigScale$  are replaced by fields of  $bsoi\_common$ . Note that  $bsoi\_common.ColourMode = bs\_ColourOpt + 1$  (so it can have values 1, 2, and 3.)
- **481** (05/03/2002) Handle case of no selected sample neatly in the Spy slave.
- 482 (11/03/2002) New map type 'd' for Departure maps in bsm.
- **483** (11/03/2002) New bugs fixed in code for loading new recordings.
- **484** (14/03/2002) Take proper care of (ENDO) files in which the recordings have different numbers of channels.

- **485** (14/03/2002) Hackers only: Repaired the *nspike* argument of *finddvdt*.
- **486** (15/03/2002) Added "file wildcard" option. Used a suitable default for windows and UNIX systems.
- **487** (15/03/2002) Grouped the new keypad buttons in the "Channel"-window using new features of *bs\_pad*.
- **488** (15/03/2002) Handle missing *bs\_tmap\_extern* data neatly, without a panic message.
- **489** (05/04/2002) Read also PruckaMapp file format.
- **490** (05/04/2002) Added grid definition files for the Krannert institute.
- **491** (10/04/2002) MAPLAB is now distributed with a separate **elconf** directory for grid definition files. This directory contains subdirectories corresponding to institutes. These subdirectories contain the grid definition files. The distributed **elconf** directory can be used as a starting point for a customized **elconf** tree. The tree structure is used by MAPLAB to create a menu structure for grid selection.
- **492** (11/04/2002) Bug fix: don't clear the point markers when a grid is defined with more channels than available in the recording, and make internal representation consistent.
- 493 (11/04/2002) Catch and report errors that occur when saving a metafile.
- **494** (11/04/2002) Use as much as possible of a rejected vector specified in a metafile, even if its length is incorrect.
- **495** (11/04/2002) Bug fix in code for recognizing metafiles.
- **496** (11/04/2002) Hackers only: Moved several variables into structs. Local *bs\_import* scripts **must be updated to reflect these changes!**
- 497 (12/04/2002) Hackers only: bs\_tmap\_extern is replaced by several fields of the bsg\_tmap.extern struct: VertexColorData, VertexTextData, VertexTextString, VertexContourData FaceColorData, FaceTextData, FaceTextString, FaceContourData, Title, SubTitle. Not all of them are implemented at this time; see tmap.web for the actual situation.
- **498** (12/04/2002) Hackers only: the *ml\_colour* function can now be used to obtain the specifications of the colours that the current version of MAP<sub>L</sub>AB uses for specific purposes.
- **499** (15/4/2002) Changed the colours of buttons and background.
- 500 (15/4/2002) Hackers only: the number of channels in each grid that is found while scanning elconf directories is stored, and an array of grids for each possible number of channels is stored to allow grid guessing with nonstandard grids.
- 501 (15/04/2002) MA<sup>P</sup><sub>L</sub>AB can now guess the correct grid, based on the number of channels in the recording, even for 'nonstandard' grids. In addition, it learns from manual grid choices.

#### version 1.49

502 (17/04/2002) Hackers only: MCHD chunk names that serve as fieldnames in *bsg\_mchd* and *bsoi\_mchd* are now uppercase.

# Bibliography

- [1] Preston Briggs. Nuweb Version 0.87b; A Simple Literate Programming Tool. Available from the CTAN archives [2] in the directory web/nuweb.
- [3] Jan-Henk E. Dambrink, Arne SippensGroenewegen, Wiek H. van Gilst, Katinka H. Peels, Cornelis A. Grimbergen, and J. Herre Kingma. Association of left ventricular remodeling and nonuniform electrical recovery expressed by nondipolar QRST integral map patterns in survivors of a first anterior myocardial infarction. *Circulation*, 92:300–310, 1995.
- [4] Mélèze Hocini, Jacques M. T. de Bakker, André C. Linnenbank, Yen Ho, Michiel J. Janse, Michel Haissaguerre, and Jacques Bonnet. High-resolution mapping of the left atrium and its afferent vessels during pacing and atrial fibrillation in canine hearts. In *Lille*, April 1998.
- [5] L. G. Horan and N. C. Flowers. The principle of waveform correlation in electrocardiographic research. J. Electrocardiol., 1:43–50, 1968.
- [6] J. G. C. Kemmelings, A. C. Linnenbank, S. L. C. Muilwijk, A. SippensGroenewegen, A. Peper, and C. A. Grimbergen. Automatic QRS detection during body surface mapping of ventricular tachycardia. In Proc. 14th Annu. Int. Conf. IEEE EMBS, pages 561–562, 1992.
- [7] J. G. C. Kemmelings, A. C. Linnenbank, S. L. C. Muilwijk, A. SippensGroenewegen, A. Peper, and C. A. Grimbergen. Automatic QRS onset and offset detection for body surface QRS integral mapping of ventricular tachycardia. *IEEE Trans. Biomed. Eng.*, 41(9):830–836, 1994.
- [8] Donald E. Knuth and Silvio Levy. The CWEB System of Structured Documentation. Addison-Wesley, 1993.
- [9] John Levine, Tony Mason, and Doug Brown. Lex & Yacc. O' Reilly, 2nd edition, October 1992.
- [10] André C. Linnenbank, Jeroen G. Snel, Mélèze Hocini, Mark Potse, Jacques M. T. de Bakker, and Cornelis A. Grimbergen. Display of activation fronts on a 3D reconstruction of the left atrium of a dog. In Proc. 20th Annu. Int. Conf. IEEE EMBS, pages I–90–93, October 1998.

- [11] Nicos Maglaveras, Jacques M. T. de Bakker, Frans J. L. van Capelle, Costas Pappas, and Michiel J. Janse. Activation delay in healed myocardial infarction: A comparison between model and experiment. Am. J. Physiol., 269:H1441–H1449, 1995.
- [12] The Math Works, Inc., 24 Prime Park Way; Natick, MA 01760-1500; USA. Getting Started with Matlab, August 1998.
- [13] The Math Works, Inc., 24 Prime Park Way; Natick, MA 01760-1500; USA. Using Matlab, August 1998.
- [14] Heidi A. P. Peeters, Arne SippensGroenewegen, Bas A. Schoonderwoerd, Eric F. D. Wever, Cornelis A. Grimbergen, Richard N. W. Hauer, and Etienne O. Robles de Medina. Body surface QRST integral mapping: Arrhythmogenic right ventricular dysplasia versus right ventricular tachycardia. *Circulation*, 95:2668–2676, 1997.
- [15] Heidi A. P. Peeters, Arne SippensGroenewegen, Eric F. D. Wever, Mark Potse, Marcel C. G. Daniëls, Cornelis A. Grimbergen, Richard N. W. Hauer, and Etienne O. Robles de Medina. Electrocardiographic identification of abnormal ventricular depolarization and repolarization in patients with idiopathic ventricular fibrillation. J. Am. Coll. Cardiol., 31(6):1406–1413, May 1998.
- [16] M. Potse, A. C. Linnenbank, A. SippensGroenewegen, and C. A. Grimbergen. Continuous localization of ectopic left ventricular activation sites by means of a twodimensional representation of body surface QRS integral maps. In Proc. 18th Annu. Int. Conf. IEEE EMBS, 1996.
- [17] Mark Potse. MA<sup>P</sup><sub>L</sub>AB; datareductie van veelkanaals ECG-registraties onder MATLAB. Vakgroep Medische Fysica en Informatiekunde, Faculteit Geneeskunde, Universiteit van Amsterdam, July 1994.
- [18] Mark Potse. Implementation of the MWEB System, A Literate Programming Tool for Matlab, September 1994. Alpha test-version.
- [19] Mark Potse. MWEB (Matlab web) version 2.10, 2000. Free software. Available from the CTAN archives [2] in the directory web/matlabweb.
- [20] Mark Potse, André C. Linnenbank, and Cornelis A. Grimbergen. MA<sup>P</sup><sub>L</sub>AB: An extensible software package for analysis of multichannel ECG recordings. In Proc. 20th Annu. Int. Conf. IEEE EMBS, pages I–59–62, October 1998.
- [21] Mark Potse, André C. Linnenbank, Pascal F. H. M. van Dessel, Norbert M. van Hemel, and Cornelis A. Grimbergen. The use of simultaneous multichannel endocardial and surface electrocardiograms for verification of exit site localization using body surface mapping. In Proc. 20th Annu. Int. Conf. IEEE EMBS, pages I–75–78, October 1998.
- [22] A. SippensGroenewegen, H. Spekhorst, N. M. van Hemel, J. H. Kingma, R. N. W. Hauer, M. J. Janse, and A. J. Dunning. Body surface mapping of ectopic left and right ventricular activation: QRS spectrum in patients without structural heart disease. *Circulation*, 82:879–896, 1990.
- [23] A. SippensGroenewegen, H. Spekhorst, N. M. van Hemel, J. H. Kingma, R. N. W. Hauer, M. J. Janse, and A. J. Dunning. Body surface mapping of ectopic left ventricular activation: QRS spectrum in patients with prior myocardial infarction. *Circ. Res.*, 71(6):1361–1378, 1992.

- [24] Arne SippensGroenewegen, Heidi A. P. Peeters, Emile R. Jessurun, André C. Linnenbank, Etienne O. Robles de Medina, Michael D. Lesh, and Norbert M. van Hemel. Body surface mapping during pacing at multiple sites in the human atrium: P-wave morphology of ectopic right atrial activation. *Circulation*, 97:369–380, February 1998.
- [25] Pascal F. van Dessel, Norbert M. van Hemel, Jacques M. de Bakker, Mark Potse, Andre C. Linnenbank, Emile R. Jessurun, and Jo A. Defauw. Comparison of simultaneous body surface mapping and 64-electrode endocardial mapping of postinfarction ventricular tachycardia exit. *Circulation*, 100 Suppl. I(18):655–656, November 1999. (abstract).
# Index

#### ?, 28

### <u>A</u>

absolute value, 24 activation detection, 22, 25 options, 27, 40 activation map, 19 activation maps, 34, 36, 37 activation selection (option), 35, 46 activation time limit (option), 36, 44 activation tracking, 36 activations, 25 addpath, 51 Alice, 55 alignment of signals, 26 Apple Macintosh, 7 aspect ratio (option), 41 Auto baseline (option), 40 averaging, 26

## $\mathbf{B}$

baseline algorithm (option), 38 baseline correction, 19 baseline width (option), 39 batch job, 29 begin of interval (option), 41 begin of window (option), 46 Birthdate (option), 47 block activation, 28, 35, 36, 45 of leads, 17, 30 blue border, 17 Body Surface Map options, 42 Body Surface Maps, 31 bs\_, 49 bs\_ActMap, 59 bs\_BaseLn, 50  $bs\_BaseLn(1), 43$  $bs\_BaseLn(2), 43$ bs\_Begins, 50

bs\_busy, 67  $bs_busy(), 60$ bs\_ChanBlock, 50 bs\_ChanSelection, 50  $bs\_chn, 66$  $bs\_chn.m, 52$ bs\_chn\_, 67 bs\_ColourOpt, 68 bs\_CurBegin, 50 bs\_CurBL, 50 bs\_CurChannel, 50 bs\_CurEnd, 50 bs\_CurEvent, 50 bs\_CurPoint, 50 bs\_CurSample, 50 bs\_CurZero, 50 bs\_dial, 58 bs\_ElConf, 68 bs\_ElConf.rcol, 68 bs\_ElConf.rgrid, 68 bs\_ElConf.rneighbours, 68 bs\_ElConf.rrow, 68 bs\_ElConf.rtype, 68 bs\_elconf\_names, 50, 51 bs\_Ends, 50 bs\_Events, 50 bs\_FigScale, 68 bs\_filename, 50 bs\_hook, 52 bs\_import, 69 bs\_init, 67  $bs\_LapW, 58$ bs\_lead, 58 bs\_listb, 67 bs\_load, 58, 68 bs\_log, 58, 62 bs\_map, 67 bs\_map.m, 52 bs\_MarkerLinewidth, 58 bs\_mdrop, 63 bs\_mdrop.m, 52 bs\_note, 58

#### INDEX

bs\_opta, 67 bs\_optwu, 67 bs\_pad, 69 bs\_panf, 58 bs\_panic, 58 bs\_Points, 50, 60 bs\_PointType, 50 bs\_Reg, 49 bs\_rejected, 49 bs\_rstat.m, 52 bs\_SamplingRate, 38 bs\_secq, 60, 67 bs\_sqrid, 52, 58, 68 bs\_shortfilename, 50 bs\_sreg, 50, 67 bs\_tmap, 62 bs\_tmap\_extern, 45, 63, 69 bs\_uqrid, 51, 52, 58, 68 bs\_Zeros, 50 bsg\_, 49 bsg\_chn, 67 bsg\_mchd, 68, 69 bsg\_mchd.conf.ElecArray, 47 bsg\_mchd.conf.FileNr, 47 bsg\_mchd.pdat.BirthDate, 47 bsg\_mchd.pdat.Diagnosis, 47 bsg\_mchd.pdat.Name, 47 bsg\_mchd.pdat.Sex, 47 bsg\_mchd.rdat.Compression, 47 bsg\_mchd.rdat.Config, 47 bsg\_mchd.rdat.Date, 48 bsg\_mchd.rdat.NBits, 47 bsg\_mchd.rdat.NChannels, 47 bsg\_mchd.rdat.NSamples, 47 bsg\_mchd.rdat.SampTime, 47 bsg\_mchd.rdat.Time, 48 bsg\_mchd.rdat.Unit, 48 bsg\_mchd.rdat.UnitPerBit, 48 bsg\_tmap.extern, 69 bsh\_, 49 bsh\_bl.m, 52 bsh\_chc.m, 52 bsh\_drop.m, 52 bsh\_map .m, 52 bsh\_rst.m, 52 bsm, 43, 67, 68 bso\_, 49 bso\_chn\_b2e, 60 bso\_Fine, 67 bso\_fontsize, 62

bso\_Grey, 67 bso\_Lead\_Begin, 67 bso\_Lead\_End, 67 bso\_Leadnr, 67 bso\_map\_ecqbox, 67 bso\_MapXSize, 67 bso\_MapYSize, 67 bso\_Nlevels, 67 bso\_PlusMinSize, 67 bso\_Polar, 67 bso\_secg, 67 bso\_secg\_names, 30, 67 bso\_secq\_nrs, 30, 67 bso\_secq\_Points, 67 bso\_Shoulders, 67 bso\_Type, 67 bsoi\_chn, 66 bsoi\_chn.autoBL, 40 bsoi\_chn. ChopSpikes, 40 bsoi\_chn.FitMethod, 40 bsoi\_chn.limit, 40 bsoi\_chn.MarkerLineWidth, 40 bsoi\_chn.range, 40 bsoi\_chn.sticky, 40 bsoi\_chn.zoomx, 40 bsoi\_chn.zoomy, 40 bsoi\_common, 68 bsoi\_common.BaselineAlgorithm, 38 bsoi\_common.BaselineWidth, 39 bsoi\_common.ColourMode, 38, 68 bsoi\_common.debug, 39 bsoi\_common.FigScale, 38 bsoi\_common.fontsize, 39 bsoi\_common.fontweight, 39 bsoi\_common.maxsize, 39 bsoi\_common.output, 39 bsoi\_common.wildcard, 39 bsoi\_fdt, 66 bsoi\_fdt.b2e, 40 bsoi\_fdt.nspike, 40 bsoi\_fdt.threshold, 40 bsoi\_fdt.tint, 40 bsoi\_fdt.ut\_bspm, 41 bsoi\_fdt.ut\_endo, 41 bsoi\_fdt.ut\_epi, 41 bsoi\_fdt.ut\_extra, 41 bsoi\_fdt.ut\_intra, 41 bsoi\_fdt.ut\_subc, 41 bsoi\_fimap.fine, 44 bsoi\_fmap.fine, 44

#### $\mathbf{74}$

#### INDEX

bsoi\_map.ecqbox, 44 bsoi\_map.fine, 43 bsoi\_map.grey, 43 bsoi\_map.LeadBegin, 43 bsoi\_map.LeadEnd, 43 bsoi\_map.leadnr, 43 bsoi\_map.Nlevels, 43 bsoi\_map.PlusMinSize, 43 bsoi\_map.polar, 44 bsoi\_map.shoulders, 43 bsoi\_map.type, 43 bsoi\_map.xsize, 42 bsoi\_map.ysize, 42 bsoi\_mchd, 69 bsoi\_secg, 67 bsoi\_secg.AspectRatio, 41 bsoi\_secq.baseline, 41 bsoi\_secq.begin, 41 bsoi\_secg.box, 41 bsoi\_secq.end, 41 bsoi\_secg.fill, 41 bsoi\_secq.fit, 41 bsoi\_secq.FontSize, 42 bsoi\_secq.height, 41 bsoi\_secg.legend, 42  $bsoi\_secg.LegendX, 42$  $bsoi\_secq.LegendY, 42$ bsoi\_secq.markpoints, 41 bsoi\_secg.max, 42 bsoi\_secq.min, 42 bsoi\_secq.NamePlacement, 42 bsoi\_secq.PaperOrientation, 42 bsoi\_secg.sep, 41 bsoi\_secq.speed, 42 bsoi\_secg.SpeedList, 42 bsoi\_secq.width, 41 bsoi\_sreg.limit, 39 bsoi\_tmap.ActLimit, 44, 45 bsoi\_tmap.cc\_offset, 46 bsoi\_tmap.colorbar, 46 bsoi\_tmap.colormode, 45 bsoi\_tmap.contourmode, 45 bsoi\_tmap.cs, 46 bsoi\_tmap.cscale, 46 bsoi\_tmap.levelmode, 45 bsoi\_tmap.Nlevels, 45 bsoi\_tmap.px, 45 bsoi\_tmap.timmode, 45 bsoi\_tmap.txtmode, 45 bsoi\_tmap.update, 44

bsoi\_tmap.ut\_bspm, 46 bsoi\_tmap.ut\_endo, 46 bsoi\_tmap.ut\_epi, 46 bsoi\_tmap.ut\_extra, 46 bsoi\_tmap.ut\_intra, 46 bsoi\_tmap.ut\_subc, 46 bsoi\_tmap.vis, 45 bsoi\_tmap.winbeain, 46

#### <u>C</u>

Carroll, Lewis, 55 cblap, 58cc\_offset, 64 cell size (option), 45 change, 57 channel, 18 Chop Spikes (option), 40 Chuzzlewit, Martin, 57 Class Menu, 19 clip bottom at (option), 42 clip top at (option), 42 clipping, 26, 27 coaxial lead, 26 *col1*, 50 color scale (option), 46 compile script, 9 Compression (option), 47 Conan Doyle, Arthur, 49 conf, 68CONF\_, 68 Configuration (option), 47 Constellation basket, 26 contour level offset (option), 46 contourc, 36, 45, 46 contours, see isochrones current channel, 16, 23, 30 current class, 19 current instance, 19 current marker, 19, 21, 22 current marker class, see current class customization, 49 CWEB, 8

#### D

data saving, 29 database, 33 atrial, 34 installation, 9 ventricular, 34

#### $\mathbf{76}$

databases, 33 Date (option), 48 DD (special lead), 24 DDSA (special lead), 24 debug mode (option), 39 DEC, 9decimal places (option), 46 DefaultChannel, 51 defghi, 68 derivative, 24, 26, 27 detect in BSPM channels (option), 41 detect in ENDO channels (option), 41 detect in EPI channels (option), 41 detect in extremities (option), 41 detect in INTRA channels (option), 41 detect in SUBC channels (option), 41 Diagnosis (option), 47 Dickens, Charles, 57 dll files, 9 do updates (option), 35, 44 down (special lead), 24 dragging, 21 DSA (special lead), 24 dV/dt (special lead), 24 dV/dx (special lead), 25 dV/dy (special lead), 25 dye sublimation printer, 37

## $\mathbf{E}$

ecg box (option), 44ectopic beats, 33 Elec Array (option), 47 ElecGrid, 51 ElecName, 30, 51, 66 ElecPos, 51, 52, 68 ElecType, 52 Enable x axis zoom (option), 40 enable x axis zoom (option), 22 Enable y axis zoom (option), 40 enable y axis zoom (option), 22 end of interval (option), 41 end of window (option), 46 endocardial electrograms, 25 eplot, 57 Experimental Cardiology department, 15 extended selection, 17, 18, 23, 30 extra leads, 23 ExtraChannelNames, 30, 51 ExtraChannels, 51

#### $\mathbf{F}$

FaceColorData, 69 FaceContourData, 69 FaceTextData, 69 Face TextString, 69 fclose, 28 ffill, 59 figure scale (option), 38 file load, 15file formats, 15 File Nr (option), 47 file wildcard (option), 39 Filename (option), 47 fill interval (option), 41 filter, 27 filters, 27 find dV/dtoptions, 40 finddvdt, 66, 69 five channels, 23, 24 flag markers, 27, 28 flood fill. 36 font size (option), 39 font weight (option), 39 fontsize (option), 42 fopen, 28 Fourier transforms, 25 fprintf, 28

### <u>G</u>

genintrp, 62 getmap, 58, 62 glossies, 37 glossy paper, 37 graphics, 36, 37 grid definition, 9, 51 grid definition files, 9 grid refinement factor (option), 43, 44

### $\mathbf{H}$

hacking, 49 halftones, 37 hand, 19, 22 hat, 21, 22 height (cm) (option), 41 Hemel, prof. dr. N.M. van, 55 hooks, 52

#### HP 9000, 9

### I

IgnoreChannels, 51 Index, 11 ink-jet printers, 37 installation, 9 Instance Box, 19 instances of markers, 19 integral map, 19 inversion, 26, 27 irreversible transforms, 26 isochrone interval (option), 44 isochrone levels (option), 45 isochrones, 36 isochrones (option), 36, 45 isomap, 46, 63-65 isomap (contour algorithm), 36

# <u>J</u>\_\_\_\_

journals, 37

# <u>K</u>

Kemmelings, Jurgen, 25

# $\mathbf{L}$

Lapl (special lead), 25 laplace, 57, 58 Laplacian lead, 26 laserprinter, 37  $LAT_FX, 55$ Latour, Auke, 55 lead begin (option), 43 lead end (option), 43 lead number (option), 43 leads, 50 left (special lead), 24 legend (option), 42 legend height (option), 42 legend width (option), 42 Linnenbank, André, 25, 55 Linux, 7-9, 55 listfile, 27 llft (special lead), 24 local activities, 19 localization, 33 continuous, 33 lrt (special lead), 24

## $\underline{\mathbf{M}}$

MA (special lead), 24 map height (option), 42map type (option), 42 map width (option), 42 MAP<sub>I</sub>AB, 3, 7–9, 11–16, 18, 25, 28, 29, 33, 35-39, 41, 46, 47, 49, 51, 52, 55-57, 59, 60, 62–64, 66–69 MAP<sub>1</sub>iB, 3, 8, 28, 29, 52, 61 mark points (option), 41 Marker Linewidth (option), 40 markers, 19, 49 classes, 19 controls for, 19 creating, 22 deleting, 22 instances, 19 moving, 21, 22 Point, 22 types, 22 Matlab, 3, 7-9, 11-13, 16, 28, 31, 37, 55 max recording size (option), 39 Medical Physics department, 15 menus, 13 metafiles, 28-29  $\max$  files, 9 min interval (option), 40 ml\_colour, 69 mload, 63 Mouse, 55 MS-Windows, 7–9, 55 MS-Windows '95, 9 MS-Windows '98, 9 MS-Windows NT, 9 MWEB, 8

### N

name placement (option), 42 NBits (option), 47 Nchannels, 50 Nchannels (option), 47 Ncols, 50 neighbouring channels, 24 NextraChannels, 51 nondipolar content, 31 nondipolarity, 31 nr of contour levels (option), 43 Nrows, 50 NSamples (option), 47

#### INDEX

#### **7**8

nspike, 69 null (special lead), 24 number of levels (option), 45

# 0

Only begin to end (option), 25, 40 options, 13, 38 for activation detection, 40 for standard ECG, 41 general, 13 MapLab, 38 saving, 14

## $\mathbf{P}$

paper speed (option), 42 paper speeds (option), 42 Patient (option), 47 pdat, 68 $PDAT_{-}, 68$ phototypesetter, 37 plot baseline (option), 41 plot boxes (option), 41 plot polar map (option), 44 plot shoulders (option), 43 positive part grey (option), 43 PostScript, 31, 37 potential map, 19 Potse, Mark, 3, 55 previews, 37 printers, 36 pseudocolor map (option), 36, 44

# $\mathbf{Q}$

QRS detection, 22 qrs detection, 25 qrs expansion, 25 QRS1 (special lead), 25 QRS2 (special lead), 25

# $\mathbf{R}$

Range (option), 40 rarith, 58, 61 raster\_neighbours, 58 raster\_row, 68 raw, 50 rdat, 68 RDAT\_, 68 RDAT\_, 68 RDAT\_Time, 62 readdir, 57 recording load, 15 overview, 16 RefChannels, 51 rejected channel, 16 revision control, 55 right (special lead), 24 rload, 61, 62, 68 rload\_total, 68 row1, 50 rsave, 28

### S

SAD (special lead), 24 SADD (special lead), 25 Sample Time (option), 47 sampling rate (option), 38 save data, 28, 29 markers, 28-29 rejected, 28-29 save, 28 separation (option), 41 Sex (option), 47show axes labels (option), 45 show BSPM leads (option), 46 show colorbar (option), 46 show ENDO leads (option), 46 show EPI leads (option), 46 show EXTR leads (option), 46 show INTRA leads (option), 46 show SUBC leads (option), 46 signal alignment, 26 signal averaging, 26 simple (contour algorithm), 36 size of extrema markers (option), 43 slaves, 21, 23, 28 Solaris, 7 sources reading, 52 special channels, 23, 24 special leads, 23 spike width (option), 40 stack tightly (option), 41 standard ECG, 17, 29 options, 41 Sticky Markers (option), 40 SubTitle, 69 Sun, 55 Sun 4, 7

Sun Sparc, 9

## $\mathbf{T}$

texts (option), 45 thermal wax printers, 37 Threshold (option), 40 Time (option), 48 Time Box, 21 time/sample nr (option), 45 *Title*, 69 *to3dgrid*, 64 transforms, 26 tricky (contour algorithm), 36

# $\overline{\mathrm{U}}$

ulft (special lead), 24 Unit (option), 48 Unit Per Bit (option), 48 University of Amsterdam, 15 UNIX, 55 up (special lead), 24 urt (special lead), 24 use colours (option), 38

# $\overline{\mathbf{V}}$

VertexColorData, 69 VertexContourData, 69 VertexTextData, 69 VertexTextString, 69 Vertical fit method (option), 40 vertical fit method (option), 23

# $\overline{\mathbf{W}}$

Watson, 49 width (cm) (option), 41 Wilde, Oscar, 38 window orientation (option), 42 window style (option), 39 Windows Maplab windows "ActMap", 57, 63 "Activation Map", 28, 34–36, 44– 46, 59, 60, 62–66, 68 "Channel", 15, 17-19, 21-24, 26-28, 31, 33-35, 37, 39, 40, 44-47, 49, 50, 52, 58, 59, 61-63, 65, 67-69 "Error", 16 "Fast IntMap", 32, 33, 44, 68

"Fast Integral Map", 31 "Fast Map", 31, 44, 65, 68 "Fast intmap", 65 "FastMap", 37 "Filter", 59 "Histogram", 36, 66, 67 "Integral Map", 31, 61 "Localize", 31, 33, 34, 57, 58, 64 "MCHD Options", 47 "MapLab", 13-16, 18, 25, 28, 29, 31, 38, 57-60 "Maplab Options", 16 "Maplab", 32 "Map", 31, 36, 42, 58 "Matlab", 13, 14, 18, 30 "Movie", 67 "Options for MapLab", 38 "Polar Map", 68 "Recording", 14-20, 23, 30, 37, 39, 50, 51, 59, 60, 62, 65-67 "Spy", 58 "Standard ECG", 17, 23, 29-31, 36, 41, 42, 60, 63 "channel", 32 "dV/dt histogram", 27 "matlab", 32 "save", 28, 29, 66

# X

X Windows, 8, 9 X-axis width (option), 39, 40

#### $\mathbf{Y}$

yellow border, 17

### $\mathbf{Z}$

zero time, 19 zooming, 22