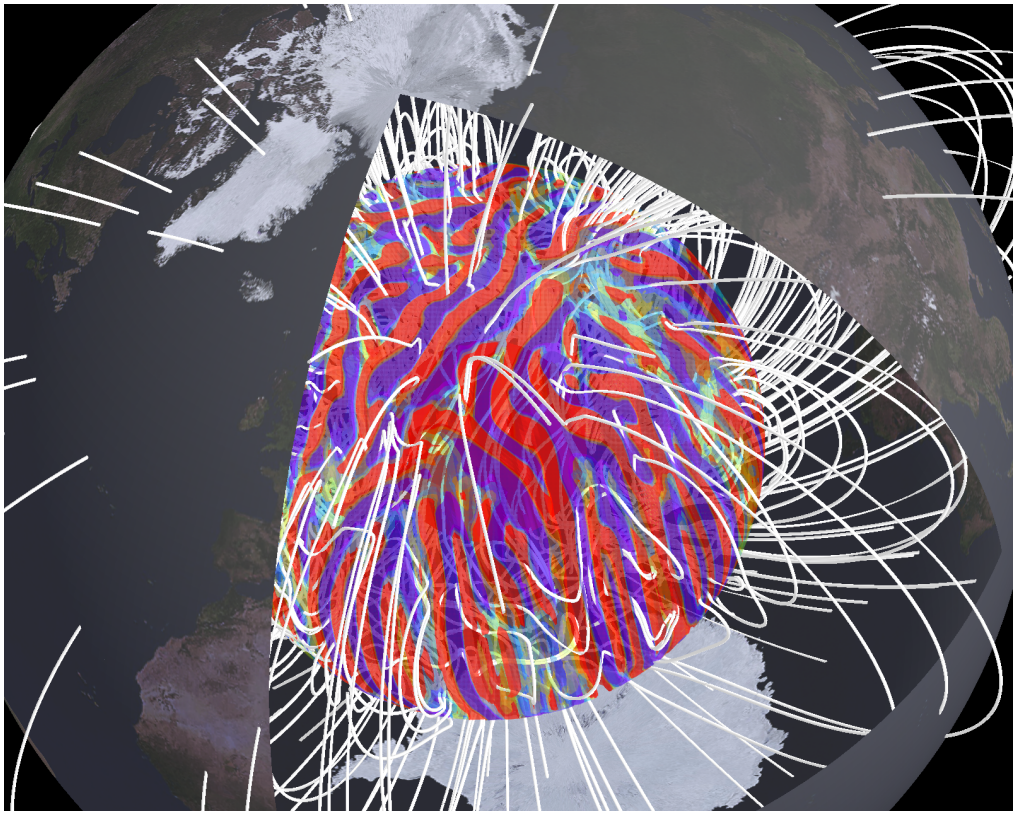
COMPUTATIONAL INFRASTRUCTURE FOR GEODYNAMICS (CIG)

Calypso

User Manual
Version 2.0



Hiroaki Matsui

www.geodynamics.org

Preface

CalypsoView is a date viewer program for sections and isosurfaces generated by Calypso. This program is intended to run on a desktop computer.

Contents

1 Introduction

CalypsoView is a date viewer program for sections and isosurfaces generated by Calypso. Calypso is a program package for magnetohydrodynamics (MHD) simulations in a rotating spherical shell for geodynamo problems. This program is intended to run on a desktop computer with a single process.

To make the program simple and small as posible, CalypsoView can only visualize cross section and isosurface results and saves image data. To visualize the data with whole volume, please use more powerful visualizaion program such as ParaView or VizIt.

This user guide provides instructions for the configuration and execution of CalypsoView.

2 History

Calypso has its origins in two earlier projects. One is a dynamo simulation code written by Hiroaki Matsui in 1990's using a spectral method. This code solves for the poloidal and toroidal spectral coefficients, like Calypso, but it calculates the nonlinear terms in the spectral domain using a parallelization for SMP architectures. The other project is the thermal convection version of GeoFEM, which is Finite Element Method (FEM) platform for massively parallel computational environment, originally written by Hiroshi Okuda in 2000. Under GeoFEM Project, Lee Chen developed cross sectioning, iso-surfacing, and volume rendering modules for data visualization for parallel computations.

In GeoFEM project, Yoshitaka Wada developed GppView, which is a mesh and surface viewer for FEM mesh data with GeoFEM format and cross sections and isosurfaces obtained by sectioning module from GeoFEM. In 2012, Hiroaki Matsui has developed the data viewer with a same features as GPPView because of the halting of the development of GPPView.

CalypsoView Ver. 0.1 supports the following features and capabilities

- Visualize cross sections and isosurface data from Calypso.
- Visualize on map using Aitoff projection for contour map.
- CalypsoView can display up to 10 sectioning and isosurface data in total.
- Ouptput sequential image files with rotating along with $x-$, $y-$, or $z-$axis (or movie file for Mac OS version)
- Ouptput time sequential image files (or movie file for Mac OS version)

- Attach image data as a texture using a spherical coordinate.

CalypsoView DOES NOT SUPPORT the following features and capabilities.

- Input result data with whole domain and construct isosurface or cross sections.

3 Acknowledgements

Calypso was primarily developed by Dr. Hiroaki Matsui in collaboration with Prof. Bruce Buffett at the University of California, Berkeley. The following NSF grants supported the development of Calypso,

- B.A. Buffett, NSF EAR-0509893; Models of sub-grid scale turbulence in the Earths core and the geodynamo; 2005 - 2007.
- B.A. Buffett and D. Lathrop, NSF EAR-0652882; CSEDI Collaborative Research: Integrating numerical and experimental geodynamo models, 2007 - 2009
- B.A. Buffett, NSF EAR-1045277; Development and application of turbulence models in numerical geodynamo simulations ; 2010 - 2012

4 Citation

Computational Infrastructure for Geodynamics (CIG) and the Calypso developers are making the source code to Calypso available to researchers in the hope that it will aid their research and teaching. A number of individuals have contributed a significant amount of time and energy into the development of Calypso. We request that you cite the appropriate papers and make acknowledgements as necessary. The Calypso development team asks that you cite the following papers:

Matsui, H., E. King, and B.A. Buffett, Multi-scale convection in a geodynamo simulation with uniform heat flux along the outer boundary, *Geochemistry, Geophysics, Geosystems*, **15**, 3212 – 3225, 2014.

5 Installation

CalypsoView has a MacOS version and Linux implimentaion. Installation procedure and look and feel are different between these implimentaion.

5.1 MacOS

CalypsoView has a binary application bundle in the package. This application needs simply to drag and drop into `/Application` folder (see Figure 1).

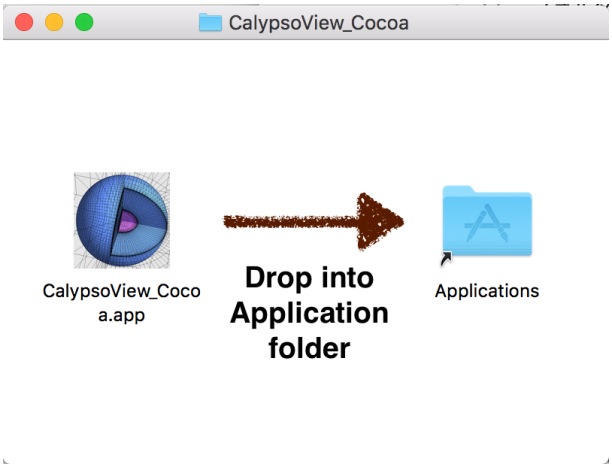


Figure 1: Folder of the MacOS binary

5.1.1 Compiler Requirements

Source code of CalypsoView_Cocoa is written in C and Objective-C. To hack and build the program, Xcode is required.

5.1.2 Library Requirements

Calypso requires the following libraries for MacOS version.

- Xcode (https://developer.apple.com)
- zlib (https://www.zlib.net)

zlib and other foundations are pre-installed in MacOS. Consequently, program should work witout installing any other libraries.

5.2 Linux

5.2.1 Compiler Requirements

Source code of CalypsoView_GLFW is written in C. Fortran90 is also used to make a C source code including shader program. Consequently, Fortran compiler is required. GCC, the GNU Compiler Collection (`https://gcc.gnu.org`) includes gfortran compiler in the most of Linux distributions. For MacOS, any fortran compiler needs to be installed because Xcode does not have fortran compiler.

5.2.2 Library Requirements

- GNU make
- zlib (https://www.zlib.net)
- libpng (http://www.libpng.org/pub/png/libpng.html)
- gtk+3 (https://developer.gnome.org/gtk3/stable/)
- OpenGL (https://www.opengl.org)
- GLFW (https://www.glfw.org)

Linux and MacOS use GNU make as a default 'make' command, but some system (e.g. BSD or SOLARIS) does not use GNU make as default. `configure` command searches and set correct GNU make command.

In the most of Linux distributions have these libraries as a default excepet for GLFW, but they sholud have a package to install GLFW.

5.3 Known problems

OpenGL on MacOS

Apple set OpenGL as a deprecated library. OpenGL is still supprted, but some alternative needs to be considered.

5.4 Directories

The top directory of Calypso (ex. `[CALYPSO_HOME]`) contains the following directories.

```
% cd [CALYPSO_HOME]
% ls
CMakeLists.txt Makefile.in configure.in examples
INSTALL bin doc src
LICENSE configure doxygen work
```

`bin`: directory for executable files

`cmake`: directory for cmake configurations

`cmake`: directory for document generated by doxygen

`doc`: documentations

`examples`: examples

`src`: source files

`work`: work directory. Compile is done in this directory.

5.5 Doxygen

Doxygen (`http://www.doxygen.org`) is an powerful document generation tool from source files. We only save a configuration file in this directory because thousands of html files generated by doxygen. The documents for source codes are generated by the following command:

```
% cd [CALYPSO_HOME]/doxygen
% doxygen ./Doxyfile_CALYPSO
```

The html documents can see by opening `[CALYPSO_HOME]/doxygen/html/index.html`. Automatically generated documentation is also available on the CIG website at `http://www.geodynamics.org/cig/software/calypso/`.

5.6 Install using `configure` command

5.6.1 Configuration using `configure` command

Calypso uses the configure script for configuration to install. The simplest way to install programs is the following process in the top directory of Calypso.

```
%pwd
[CALYPSO_HOME]
% ./configure
...
% make
...
% make install
```

After the installation, object modules can be deleted by the following command;

```
% make clean
```

./configure generates a Makefile in the current directory. Available options for `configure` can be checked using the `./configure --help` command. The following options are available in the `configure` command.

```
Optional Features:
--disable-option-checking  ignore unrecognized --enable/--with options
--disable-FEATURE       do not include FEATURE (same as --enable-FEATURE=no)
--enable-FEATURE[=ARG]  include FEATURE [ARG=yes]
--enable-silent-rules   less verbose build output (undo: "make V=1")
--disable-silent-rules  verbose build output (undo: "make V=0")
--enable-cocoa          Use Cocoa framework
--enable-dependency-tracking
do not reject slow dependency extractors
--disable-dependency-tracking
speeds up one-time build

Optional Packages:
--with-PACKAGE[=ARG]    use PACKAGE [ARG=yes]
--without-PACKAGE       do not use PACKAGE (same as --with-PACKAGE=no)
--with-zlib=DIR root directory path of zlib installation defaults to
/usr/local or /usr if not found in /usr/local
--without-zlib to disable zlib usage completely
--with-x                use the X Window System

Some influential environment variables:
CC          C compiler command
CFLAGS      C compiler flags
LDFLAGS     linker flags, e.g. -L<lib dir> if you have libraries in a
nonstandard directory <lib dir>
LIBS        libraries to pass to the linker, e.g. -l<library>
```

```
CPPFLAGS    (Objective) C/C++ preprocessor flags, e.g. -I<include dir> if
you have headers in a nonstandard directory <include dir>
PKG_CONFIG  path to pkg-config utility
CPP         C preprocessor
ZLIB_CFLAGS C compiler flags for ZLIB, overriding pkg-config
ZLIB_LIBS   linker flags for ZLIB, overriding pkg-config
PNG_CFLAGS  C compiler flags for PNG, overriding pkg-config
PNG_LIBS    linker flags for PNG, overriding pkg-config
GL_CFLAGS   C compiler flags for GL, overriding pkg-config
GL_LIBS     linker flags for GL, overriding pkg-config
GTK3_CFLAGS C compiler flags for GTK3, overriding pkg-config
GTK3_LIBS   linker flags for GTK3, overriding pkg-config
GLFW_CFLAGS C compiler flags for GLFW, overriding pkg-config
GLFW_LIBS   linker flags for GLFW, overriding pkg-config

Use these variables to override the choices made by 'configure' or to help
it to find libraries and programs with nonstandard names/locations.
```

At the end of the configuration, The following message can use to check if libraries can be refered correctly:

```
-----   Configuration summary   --------------------------------

host:        "x86_64-apple-darwin16.7.0"

Use Cocoa...           no
Use X Window...       yes

Use zlib ...          yes
Use PNG output...     yes

Use GTK3+...          yes
Use GLFW...           yes

-----------------------------------------------------------------
```

5.6.2 Compile

Compile is performed using the `make` command. The Makefile in the top directory is used to generate another Makefile in the `work` directory, which is automatically used to complete the compilation. The object file and libraries are compiled in the `work` directory.

Finally, the executive files are assembled in `bin` directory. You should find the following programs in the `bin` directory.

`kemoviewer_GLFW:`
: Viewer program

The following library files are also made in `work` directory.

`libcalypsoview.a:` CalypsoView library

5.6.3 Clean

The object and fortran module files in `work` directory is deleted by typing

```
% make clean
```

This command deletes files with the extension `.o`, `.mod`, `.par`, `.diag`, and .

5.6.4 Distclean

To revert the files and directory to the original package, use make distclean as

```
% make distclean
```

5.6.5 Install

The executive files are copied to the install directory `$(INSTDIR)/bin`. The install directory `$(INSTDIR)` is defined in Makefile, and can also set by `${--prefix}` option for `configure` command. Alternatively, you can use the programs in `${SRCDIR}/bin` directory without running `make install`. If directory `${PREFIX}` does not exist, `make install` creates `${PREFIX}`, `${PREFIX}/lib`, `${PREFIX}/bin`, and `${PREFIX}/include` directories. No files are installed in `${PREFIX}/lib` and `${PREFIX}/include`.

5.6.6 Construct dependecies (only for developper)

C source files need dependency among include files. Consequently, list of dependency of source files are saved in the file `Makefile.depends` in each directory. When you modify the source files with changing the module usage, `Makefile.depends` files need to be updated. To update the `Makefile.depends`files, use the `make` command at the `[CALYPSO_HOME]` directory as

```
% make depends
```

The dependency is generated by the gcc with `-MM -w -DDEPENDENCY_CHECK` option. Consequently, the dependencies need to be generated by the environment with gcc or compatible compiler. After generating the dependency, you can transfer the modified package and build without using gcc.

5.7 Install without using configure

It is possible to compile Calypso without using the `configure` command. To do this, you need to edit the `Makefile`. First, copy `Makefile` from template `Makefile.in` as

```
% cp Makefile.in Makefile
```

In Makefile, the following variables should be defined.

`SHELL` Name of shell command.

`SRCDIR` Directory of this Makefile.

`INSTDIR` Install directory.

`MPICHDIR` Directory names for MPI implementation. If you set fortran90 compiler name for MPI programs in `MPIF90`, you do not need to define this valuable.

`F90` Command name of local Fortran 90 compiler to make a C source file including GLSL shader sources.

`AR` Command name for archive program (ex. `ar`) to generate libraries. If you need some options for archive command, options are also included in this valuable.

`RANLIB` Command name for `ranlib` to generate index to the contents of an archive. If system does not have `ranlib`, set `true` in this valuable. `true` command does not do anything for libraries.

`OPTFLAGS` Optimization flags for C compiler

`BLAS_LIBS` Library lists for BLAS (ex. `-lblas`)

`ZLIB_CFLAGS` Option flags for zlib (ex. `-I/usr/include`)

`ZLIB_LIB` Library lists for zlib (ex. `-L/usr/lib -lz`)

`PNG_CFLAGS` Option flags for libpng (ex. `` `pkg-config --cflags libpng` ``)

`PNG_LIBS` Library lists for libpng (ex. `` `pkg-config --libs libpng` ``)

`X_CFLAGS` Option flags for X window (ex. `|pkg-config --cflags x11—`)

`X_LIBS` Library lists for X window (ex. `` `pkg-config --libs x11` ``)

`OPENGL_INC` Option flags for OpenGL (ex. `` `pkg-config --cflags glfw3` ``)

`OPENGL_LIBS` Library lists for OpenGL (ex. `` `pkg-config --libs gl` ``)

`GTK3_CFLAGS` Option flags for gtk+-3 (ex. `` `pkg-config --cflags gtk+-3.0` ``)

`GTK3_LIBS` Library lists for gtk+-3 (ex. `pkg-config --libs gtk+-3.0` `)

`GLFW_CFLAGS` Option flags for GLFW (ex. `` `pkg-config --cflags glfw3` ``)

`GLFW_LIBS` Library lists for GLFW (ex. `` `pkg-config --libs glfw3` ``)

5.8 Install using cmake

CMake is a cross-platform, open-source build system. CMake can be downloaded from http://www.cmake.org. The following procedure is required to install.

1. Create working directory (you can also use `[CALYPSO_HOME]/work`).
2. Generate Makefile and working directories by `cmake` command.
3. Compile programs by `make` command.

In this section, `[CALYPSO\_HOME]/work` is used as the working directory. Options for CMake can be checked by `cmake -i [CALYPSO_HOME]` command at `[CALYPSO_HOME]/work`. There are a number of options can be found, but the following valuables are important settings for installation:

- Install directory

  `CMAKE_INSTALL_PREFIX`
  : Install directory

- Compiler settings

`CMAKE_Fortran_COMPILER`
: Fortran90 compiler.

`CMAKE_c_COMPILER` C compiler.

`CMAKE_Fortran_FLAGS`
: Optimization flags for Fortran90 compiler.

`CMAKE_c_FLAGS`
: Optimization flags for C compiler.

- Manual settings for optional features

  `CMAKE_LIBRARY_PATH`
  : CMake library search paths. This directory is used to search FFTW3 library.

  `CMAKE_INCLUDE_PATH`
  : CMake include search paths. This directory is used to search include file for FFTW3.

The easiest example of using CMake on Mac OS X with gcc9 is the following:

```
% cd build
% cmake ~/CALYPSO/ -DCMAKE_Fortran_COMPILER=/opt/local/bin/gfortran-mp-9
? -DCMAKE_c_COMPILER=/opt/local/bin/gcc-mp-9 \
? -DCMAKE_Fortran_FLAGS="-O3 -g" -DCMAKE_c_FLAGS="-O3"
```

After configuration, compile and install are started by

```
% make
...
% make install
```

After running `make` command, execute files are built in `[CALYPSO_HOME]/work/bin` directory.

6 Start the program

6.1 Mac OS

The program will start by double clicking the application icon. Viewer window and menu window will open as shown in Figure 2.

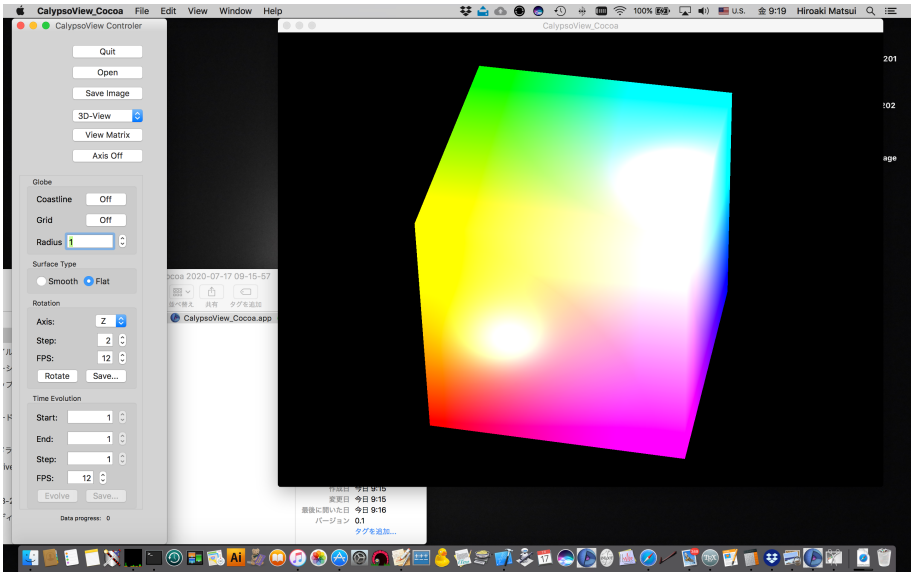


Figure 2: Windows of CalypsoView at starting.

6.2 Linux

The program is started by input the command in terminal as

```
% make
...
% [BINDIR]/CalypsoView_GLFW
```

The viewer window and menu window will be displayed as shown in Figure 3.

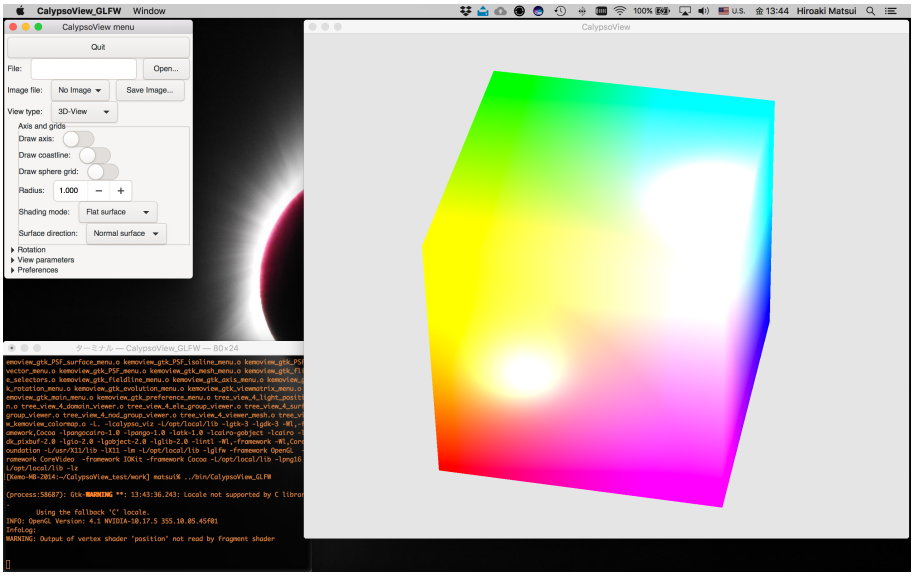


Figure 3: Windows of CalypsoView at starting.

7 Usage of program

The menu window at starting is shown in Figure 4. The program can finish by pushing ”Quit” button (1 in Figure 4).

7.1 Open file

First of all, data for visualization is required to be loaded. The file menu is opend by clicking ”Open” bottun (2 in Figure 4). In Mac version, the file can drag and drop into the viewer window. In Linux version, file name can input the file name box. CalypsoView can treat a unstructured grid data with triangle element or quadrature elements. The file name has to have a step number and extension as `[file_prefix].[step #].[extension]`. The avaiable data formats and extensions are listed in Table 1. For (compressed) sectioning binary data, the file for grid data `[file_prefix].0.sgd` or `[file_prefix].0.sgd.gz` is required in the same directory as the data file.

After loading the data, the menu is expanded to control surface visualization parameters as shown in Figure 6. CalypsoView can load up 10 surfacing data. The detailed control of the sectionig data is described in section 7.6.

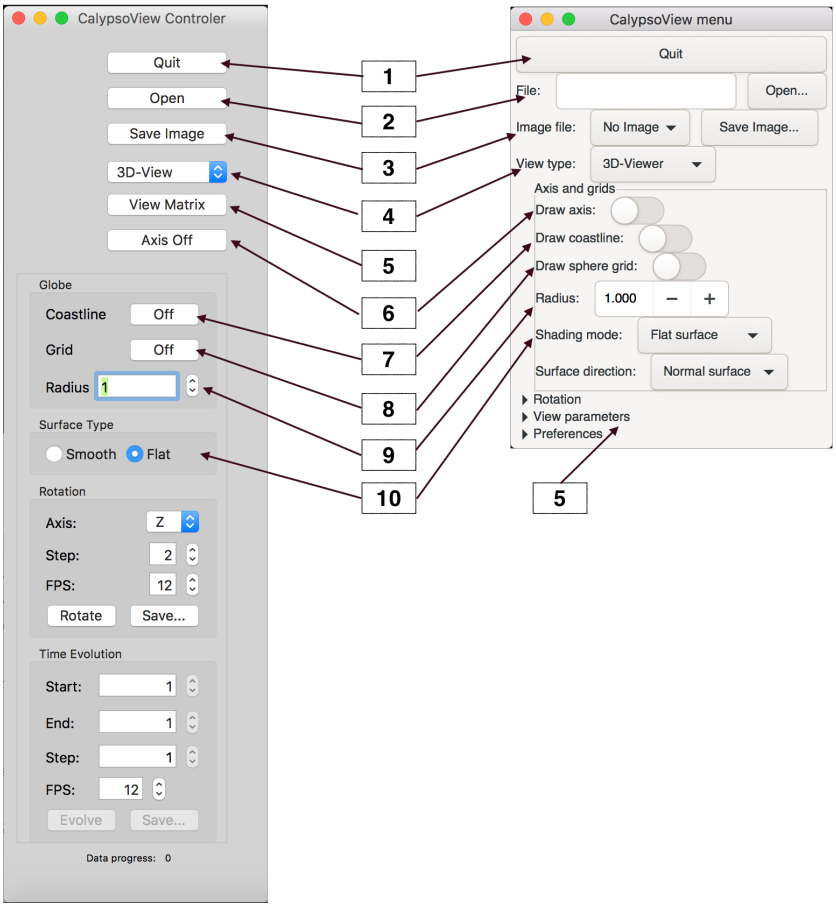


Figure 4: Menu windows of CalypsoView at starting.

Table 1: Data format and extensions for CalypsoView

File format	`extension`
VTK	`.vtk`
Compressed VTK	`.vtk.gz`
Sectioning binary	`.sdt`(*)
Compressed sectioning	`.sdt.gz`(*)
Isosurface Binary	`.sfm`
Compressed isosurface binary	`.sfm.gz`

(*) Grid data `[file_prefix].0.sgd` or `[file_prefix].0.sgd.gz` is required.

Note: At the first time using CalypsoView on MacOS, nothing could be displayed in the viewer window. In that case, lighting parameter may be missing. Please set light parameters in Preference menu.

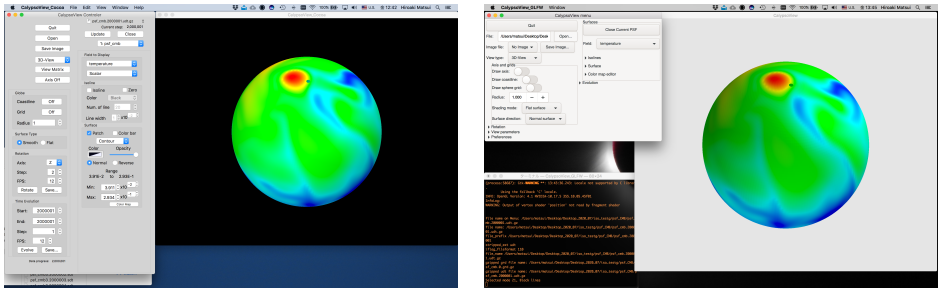


Figure 5: Desktop after loading data for MacOS (lest) and Linux (right).

7.2 Save image

To save image of the viewer window, click "Save image" bottun (3 in Figure 4) and set the image file name and direcotry to save in the file menu. Image data can save as PNG (`.png`) or bitmap (`.bmp`) format. PNG format is smaller size than bitmap format. If no file extension set in the file save menu, file format is chosen in the default faumat in the Preference menu (for Mac) or in the next of the "Save Image" bottun (Linux).

7.3 Interface of Viewer window and viewing mode

The visalised objects can move by mouse or trackpad. THe mouse interface depends on the viewing mode. The viewing mode is selected by View type menu (4 in Figure 4). The following viewing mode can choose:

3D-View: 3-dimentional view.

Streo-View: Streo view using anaglyph. Please use red and blue glass.

Map-View: Map projection using Aitof projection.
Map projection does not support griph (arrow) visualization.

XY-View: Display parallel with xy-plane.

XZ-View: Display parallel with xz-plane.

YZ-View: Display parallel with yz-plane.

The mouse interface is the follwing:

Push and drag:

3D-View and Streo-View Rotate object

Map-View, XY-View, XZ-View, and YZ-View Move object horizontally in screen.

Swipe by two finger: Zoom in and out.

Push two finger and drag: Move object horizontally in screen.

Push option and drag: Move object vertically in screen.

7.3.1 Axis and grids

The axis is drawn when "Axis" switch (6 in Figure 4) is turned on. The axis is not shown in the Map-View mode. The coastline is drawn when "Coastline" switch (7 in Figure 4) is turned on. A grid on a sphere with 30 degree increment is drawn when "grid" switch (8 in Figure 4). THe radius of the coastline and grid are set in the "Radius" box (9 in Figure 4). Example images for axis, coastline, and grid are shown in Figure **??**.

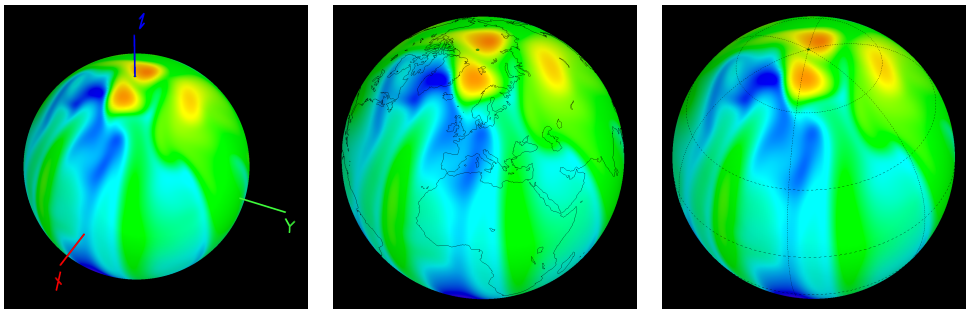


Figure 6: Images with axis(left), with coastline(middle), and sphere grid(right).

7.4 Preference menu

7.5 View transfer menu

7.6 Surfacing menu

References

[1] Bullard, E. C. and Gellman, H., Homogeneous dynamos and terrestrial magnetism, *Proc. of the Roy. Soc. of London*, **A247**, 213–278, 1954.

[2] Christensen, U.R., Aubert, J., Cardin, P., Dormy, E., Gibbons, S., Glatzmaier, G. A., Grote, E., Honkura, H., Jones, C., Kono, M., Matsushima, M., Sakuraba, A., Takahashi, F., Tilgner, A., Wicht, J. and Zhang, K., A numerical dynamo benchmark, *Physics of the Earth and Planetary Interiors*, **128**, 25–34, 2001.

Appendix A Definition of parameters for control files

A.1 Block `view_transform_ctl`

Parameters to construct view transform matrix and projection matrix are defined in this block.
(Back to Section 7.5)

A.1.1 Block `image_size_ctl`

Number of pixels of the image is defined in this block.

`x_pixel_ctl [NUM_PIXEL_X]`
Number of pixels in the horizontal direction `[NUM_PIXEL_X]` is defined by integer.

`y_pixel_ctl [NUM_PIXEL_Y]`
Number of pixels in the veritcal direction `[NUM_PIXEL_Y]` is defined by integer.

A.1.2 Array `viewpoint_in_viewer_ctl [DIRECTION] [POSITION]`

Position of eye in the viewer coordinate is defined in each `[DIRECTION]` and `[POSITION]`. `[DIRECTION]` is set by text (`x, y,` or `z`), and `[POSITION]` is set by real.

A.1.3 Array `look_at_point_ctl [DIRECTION] [POSITION]`

Position to look at is defined in each `[DIRECTION]` and `[POSITION]`. `[DIRECTION]` is set by text (`x, y,` or `z`), and `[POSITION]` is set by real.

A.1.4 `scale_factor_ctl [SCALE]`

Scale of the object is defined by real. In viewer coordinate, object is scaled by 1 / `[SCALE]` times.

A.1.5 Array `view_rotation_vec_ctl [DIRECTION] [POSITION]`

Direction of the rotation axis of the object is defined by `[DIRECTION]` and `POSITION]. \verb|DIREC`
is set by text (`x, y,` or `z`), and `[POSITION]` is set by real.

A.1.6 `view_rotation_deg_ctl [DEGREE]`

Angle of the rotation of the object is defined by `[DEGREE]`.

A.1.7 Block `projection_matrix_ctl`

Parameters of projection is defined in this block.

`perspective_angle_ctl [PERSPECTIVE]`
Angle of perspective `[PERSPECTIVE]` is defined by real.

`perspective_xy_ratio_ctl [XY_RATIO]`
Aspect ratio of horizontal and vertical screen `[XY_RATIO]` is defined by real.

`perspective_near_ctl [NEAR]`
Nearrest distance from eye `[NEAR]` is defined by real.

`perspective_far_ctl [FAR]`
Farest distance from eye `[FAR]` is defined by real.

A.1.8 Block `streo_view_parameter_ctl`

Parameters of streo image is defined in this block. This block is only used for the streo view.

`focal_point_ctl [FOCAL]`
Focal point `[FOCAL]` is defined by real.

`eye_separation_ctl [SEPARATION]`
Eye separation `[SEPARATION]` is defined by real.

A.2 Block `pvr_color_ctl`

Parameters to construct colormap and colorbar information in this block.
(Back to Section 7.5)

A.2.1 Block `colormap_ctl`

Parameters to construct colormap information in this block.

A.2.2 Block `colorbar_ctl`

Parameters to construct colorbar information in this block.

Appendix B GNU GENERAL PUBLIC LICENSE

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent li-

censes, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

(b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

(c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

(a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source

code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

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

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

   This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify

a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> one line to give the program's name and a brief idea of what it does.
> Copyright (C) yyyy name of author
>
> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright (C) yyyy name of author
> Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
> This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program
> 'Gnomovision' (which makes passes at compilers) written by James Hacker.
>
> signature of Ty Coon, 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.