GRTensorII

GRTensorII Release 1.50
For MapleV Releases 3 and 4

F. Installation and setup

Peter Musgrave
Denis Pollney
Kayll Lake

July 1996

Contents

Queen's University at Kingston, Ontario

This booklet outlines the installation and configurable options of the GRTensorII package.

1 Installation

The GRTensorII package is distributed as a set of MapleV library files, as follows:

`grii.m` – contains the GRTensorII program itself composed of the commands described in this series of booklets, along with the standard tensor object library.

`basislib.m` – contains definitions for objects defined for bases and NP tetrads.

`invar.m` – contains definitions for the scalar invariants of the Riemann tensor.

`dinvar.m` – contains definitions of some differential invariants (see `?dinvar`).

`grtools.m` – contains some extra commands (see `?difftool`, `?grlimit`).

With the MapleV.3 version of GRTensorII, there is also included a library `griihelp.m` which contains the online help information. The corresponding file for the MapleV.4 version is called `maple.hdb`. (See Section 3).

In addition to these libraries, a directory of input spacetimes (a collection of `*.mpl` files) is also distributed with GRTensorII.

The library files should be placed in a directory within the MapleV library path specified by the variable `libname`. To change the value of this variable so that it includes a particular directory, use the command

```
> libname := `c:/users/grtensor/lib/`, libname:
```

which in this case appends the directory `c:/users/grtensor/lib` to the `libname` sequence. On Unix machines, by adding this command to a file named `.mapleinit` in the user's home directory, it will be executed whenever MapleV is started, ensuring that the library directory is accessible (for MS-DOS/Windows based machines, the file `maple.ini` serves the same purpose).

The installation procedure for GRTensorII is as follows:

1. Copy the GRTensorII library files into their own directory or the standard MapleV library directory on the machine. (MapleV.4 users should be careful not to overwrite the standard `maple.hdb` with GRTensorII's `maple.hdb` help file.)

2. Copy the GRTensorII metric files into their own directory.

3. Ensure that the directory in which the library files have been stored (Step 1) is listed in the MapleV `libname` variable. If not, then modify the `.mapleinit` file (on Unix platforms, otherwise `maple.ini`), as described below.

4. Create a file called `grtensor.ini` in one of the directories specified by the `libname` variable. To this file, add a line which assigns the variable `grOptionMetricPath` to the metric directory specified in Step 2.[a]

[a] The `grOptionMetricPath` variable is described in Section 4. The `grtensor.ini` file is described in Section 2.

 GRTensorII development has been supported by the Natural Science and Engineering Research Council of Canada and the Advisory Research Committee of Queen's University.

Startup: The command

```
> readlib ( grii ):
```

reads the GRTensorII library into MapleV. At this point all of the GRTensorII commands described in these booklets are available to the user. To initialize the program, run the command

```
> grtensor():
```

If a file called `grtensor.ini` exists in one of the directories specified by the `libname` variable, it is read and executed by the `grtensor()` command (see Section 2).

To check that GRTensorII is installed properly, start a new MapleV session and execute the following commands:

```
> readlib ( grii ):
> grtensor():
> qload ( npschw ):
> grcalc ( RicciSc ):
> grdisplay ( _ ):
```

This set of commands checks that the `grii.m` library is readable, loads a spacetime from the directory specified by `grOptionMetricPath`, and carries out a simple calculation to ensure that the automatically loading libraries (`basislib.m` in this case) can be read properly.

Support libraries: For the MapleV.4 version, online help will be read as it is requested from the `maple.hdb` file included with GRTensorII, provided it is in the library path. For the MapleV.3 version, however, the online help pages are not read by default when the GRTensorII library is loaded, but must be loaded separately using the command

```
> readlib ( griihelp ):
```

This command can also be added to the `grtensor.ini` file so that it will be executed automatically when the GRTensorII package is initialized by the `grtensor()` command.

The only other library which needs to be loaded manually is the `grtools.m` library, which can be read at any time using `readlib`. The other libraries (`basislib.m`, `invar.m`, `dinvar.m`) contain GRTensorII object definitions which will be automatically loaded when they are needed (by commands such as `grcalc()`). They can also be loaded manually, however this should *not* be done using `readlib()`. Instead, the command `grlib()` exists for the purpose of loading GRTensorII object libraries:

`grlib ( `*libName*` )`

libName – The name of a GRTensorII object library which is to be loaded.

Example: `> grlib ( basislib ):`

2 Initialization

The GRTensorII environment can be controlled by the creation of an initialization file. In addition to displaying version information, the command `grtensor()` searches the MapleV library path[1] for the file `grtensor.ini`. This file can contain any number of MapleV commands, and in particular GRTensorII commands, to set up the environment in which GRTensorII is run. For instance, the `grtensor.ini` file commonly contains settings for the `grOption` variables (described in Section 4), such as directories in which to locate metric files.

For example, if metrics are to be loaded from the `c:/users/grtensor/metrics` directory, the command

```
grOptionMetricPath := `c:/users/grtensor/metrics/`:
```

could be added to the `grtensor.ini` file. The `grtensor()` command reads and executes the commands in the initialization file, in this case setting the `grOptionMetricPath` variable to its desired default value.

3 Online help

Online help is provided for all of the GRTensorII commands described in these booklets. The method of providing the help differs slightly from MapleV.3 to MapleV.4.

In the MapleV.3 version of GRTensorII, online help is available upon loading the library `griihelp.m`. (The help screens have been kept separate from the program library itself in order allow the user the option of not including the help screens in situations where memory usage is critical.) Online help can be loaded automatically when the `grtensor()` command is executed by adding the line

```
readlib ( griihelp ):
```

to the `grtensor.ini` file.

For the MapleV.4 version, the online help is contained in the file `maple.hdb` provided with the GRTensorII package. This file should be located in a directory specified by the `libname` variable. It will be read as help is requested.

[1] Specified by the `libname` variable (see `?libname`).

Help pages exist for each of the GRTensorII commands described in these booklets. In addition, the following pages provide some additional summarized information:

`?grt_commands` – a summary of GRTensorII commands.

`?grt_objects` – a summary of objects in the standard GRTensorII library. See Booklet *C: Calculating tensor components.*

`?grt_operators` – a list of GRTensorII operators contained in the standard library. See Booklet *C: Calculating tensor components.*

`?grt_basis` – a list of objects contained in the standard libarary useful for calculations in space-times described by a basis or null tetrad.

Additional information can be found at the GRTensorII world wide web site,

`http://astro.queensu.ca/~grtensor/`

Inquiries and bug reports should be directed to

`grtensor@astro.queensu.ca`

The authors welcome any and all feedback regarding the GRTensorII package and its use.

4 Global option variables

A number of global variables control the GRTensorII environment. Their values can be displayed using the command `groptions()`. The values of the `grOption` variables can be changed by re-assigning them. For example,

```
> grOptionCoordNames := false:
```

sets the value of the `grOptionCoordNames` variable. The following option variables can be set:

`grOptionAlterSize` (boolean, default=`false`) If this variable is set to `true`, the new size of each component modified by `gralter()` is displayed as the command executes. Sizes are given in MapleV words.

`grOptionCoordNames` (boolean, default=`true`) This variable controls whether tensor indices are displayed as names (eg. t, r, theta, phi) or numbers when their components are output using `grdisplay()`.

`grOptionDefaultSimp` (integer or name, default=`normal`) This variable controls the type of simplification called during the calculation of an object by the `grcalc()` command. The default is to use the MapleV routine `normal()`. The `grOptionDefaultSimp` variable can be set to any of:

`simplify, trig, power, hypergeom, radical, expand, factor, normal, sort, sqrt, trigsin, cons, consr`

The names correspond to the names used in the second argument of the `gralter()` command (see Booklet *C: Calculating tensor components*).

In general it is recommended that `gralter()` be used when simplification becomes necessary rather than changing the global simplification via this variable, as it is unlikely that any particular choice will work well in general situations, and an incorrect choice can greatly increase calculation times.

`grOptionDisplayLimit` (integer, default=5000) This integer variable controls the size threshold beyond which a tensor component will not be displayed. If the component exceeds this limit only its size is displayed. The value is measured in words as determined by the MapleV `length()` function.

`grOptionLLSC` (boolean, default=`true`) This variable indicates whether the Landau-Lifshitz spacelike signature convention is to be assumed on loading and defining new spacetimes. If set to `true`, then four dimensional spacetimes specified by a metric, g_{ab}, or a general basis for which the signature has not been explicitly given, are assumed to have signature +2 (where signature is defined as the number of positive metric components minus the number of negative metric components in a locally orthonormal basis). Bases satisfying an NP inner product are assumed to have signature -2. If the `grOptionLLSC` variable is set to `false`, then the `sig` object will not be assigned unless it is explicitly given by the spacetime input file or specified in `makeg()`.

`grOptionMetricPath` (string, default=homedir) This option variable is a MapleV string which specfies the default directory from which metric files are to be loaded (by the `qload()` command or to which they will be saved (by the `makeg()` command). The sub-directory names should be separated using a forward slash, '/', even on MS-DOS/Windows systems. On these systems directory names must also end with a '/', as in

`‘c:/users/grtensor/metrics/‘`

`grOptionqloadPath` (sequence, default=unassigned) This variable can be used to specify a set of directories which will be searched by `qload()` when loading metrics. Individual directory names are listed as MapleV strings with sub-directories separated by the forward slash character, '/'. Directores are searched in the order in which they are listed. If this variable is left unassigned, or if the metric file is not found in any of the listed directories, then the directory specified by the `grOptionMetricPath()` variable is searched.

`grOptionTermSize` (integer, default=100) This variable effects the display of 1- and 2-index objects using the `grdisplay()` command. If the size of individual components of such objects are smaller than this value, then the object will appear in the form of a matrix when displayed using `grdisplay()`. If the components are larger than the value of this variable, they are listed individually as they are for tensors with three or more indices. By setting this variable to some very high value the matrix form of display can be assured, while setting this value to zero will prevent matrix-type display of tensors. The size of object is measured in words as returned by the MapleV `length()` command.

`grOptionTrace` (boolean, default=`true`) If this variable is set to `true`, then the names of objects calculated during the execution of a `grcalc()` or `grcalcalter()` command will be displayed as they are computed. Note, this option may not work for some MapleV windowed interfaces for which the output for the `printf()` command is not flushed until the calling procedure is completed.

`grOptionWindows` (boolean, default=`true`) If this variable is set `true` the GRTensorII 'windows' display routines are used in the output of tensor components via `grdisplay()`. These routines provide support for superscript and subscript of indices in windowed versions of MapleV. However, in text-based versions, they display a number of extraneous characters than confuse the output. This option should be set to `false` if text-based MapleV is being used to run GRTensorII.

`grOptionVerbose` (boolean, default=`false`) When this variable is set `true` a notification message is printed out as each component is calculated by `grcalc()` or `grcalcalter()`.

Commands described in this booklet:

The information contained in this booklet is also available from the following online help pages:

`?grtensor`, `?groptions`, `?grlib`, `?libname`, `?grt_commands`, `?grt_objects`, `?grt_basis`, `?grt_operators`, `?grt_invars`.