SOFTWARE FOR A CONFORMALLY INVARIANT YANG-MILLS TYPE ENERGY AND EQUATION ON 6-MANIFOLDS

LAWRENCE J. PETERSON

Abstract. This document contains some guidelines for understanding and using the computer software files that accompany this document. The author used the these files, together with *Mathematica* and John M. Lee's *Ricci* software package, to compute many of the symbolic formulae appearing in Gover, Peterson, and Sleigh's article "A conformally invariant Yang-Mills type energy and equation on 6-Manifolds."

6 January 2022

1. Overview

Several computer software files should accompany this document. These software files relate to the article "A conformally invariant Yang-Mills type energy and equation on 6-Manifolds," by A. Rod Gover, Lawrence J. Peterson, and Callum Sleigh, [4]. This document contains some guidelines for reading, understanding, and using these software files. The files, along with this document, are available on the University of North Dakota Scholarly Commons. The Scholarly Commons is a Web-based repository of articles, white papers, data, and other types of scholarly materials. It is located at

`https://commons.und.edu`

The second author of [4] developed most of the software files that accompany this document, and he then used these files to derive and verify several of the symbolic formulae that appear in that article. In doing this, he used *Mathematica* together with John M. Lee's *Ricci* software package. See [5, 6]. *Mathematica* is a computer algebra system that one may purchase and install on most personal computers. *Ricci* is a *Mathematica* package that one may use together with *Mathematica* to perform tensor calculus operations in differential geometry. One may only use *Ricci* if one is also using *Mathematica*.

Twelve of the software files used in the writing of [4] are *Mathematica* "notebooks." These notebooks contain *Ricci* and *Mathematica* software commands which the author of this document used in his work with the symbolic formulae in [4]. These twelve notebooks accompany this document. The names of all twelve notebooks end with the suffix "`.nb`." All twelve notebooks contain commands which ask *Mathematica* to read in various other files. The names of these other files end in the suffix "`.m`." These files should all accompany this document. One of these files is the file `Ricci1.61.m`. This file contains the source code for the *Ricci* software package, [5]. Lee has granted everyone permission to copy, modify, and redistribute the *Ricci* package *subject to certain restrictions*. He states these restrictions at the top of the `Ricci1.61.m` file. You may view this file with many file reader programs. Each of the twelve notebooks reads in the `Ricci1.61.m` file.

Further information on *Ricci*, including a user's manual, may be available at Lee's Web site. Lee's Web site is currently located at

`https://sites.math.washington.edu/~lee`

One may also obtain information about *Ricci* from *Ricci* itself. To do this, one begins by opening a notebook and issuing the command to read in the `Ricci1.61.m` file. One then types in and runs (i.e. issues) a command beginning with the "`?`" symbol and followed by the name of a *Ricci* command. For example, the command "`?TensorSimplify`" returns a brief description of *Ricci*'s `TensorSimplify` command.

This document (the document you are currently reading) includes a list of the file names of the twelve notebooks that we discussed above and a brief description of the contents of each of these notebooks. It also contains brief descriptions of most of the other files that accompany this document. In addition, this document contains some technical notes which may help the reader to download the notebooks and accompanying files and actually run the notebooks with *Mathematica*. (To "run" a notebook means to have *Mathematica* execute the software commands in the notebook.) One section of this document provides an important warning which may help readers avoid problems in future work sessions with *Mathematica*. This document also contains a section that discusses the sign conventions and notation of [1, 2, 3]; this will help the reader understand some of the notebooks. The document contains a disclaimer statement and some notes on possible future versions of *Mathematica*.

The authors of [4] collaborated on the writing of their article, and they also discussed the software work related to the article. But except for the file `Ricci1.61.m`, the second author of [4] was the sole developer of the software code that accompanies this document. So for the remainder of the present document and throughout the twelve *Mathematica* notebooks that accompany it, the words "I," "my," and "me" will refer to the author of the present document. The words "we," "our," and "us" will refer to Gover, Sleigh, and me or to the reader and me.

As of the time of this writing, a slightly updated version of [4] is under review for publication in a journal.

2. Disclaimer

Rod Gover, Callum Sleigh, and I are not responsible for the consequences of any errors in this document or the accompanying files. I am nevertheless confident that the statements and formulae in [4] are correct.

3. Downloading, Reading, and Using the Files

Depending on the software available on one's computer, it may be possible for a person to read my twelve notebooks and the associated files without saving them to a computer hard drive or other storage device. If the name of a file ends in the

"`.m`" suffix, one should be able to read the file with many text-reading or text-editing programs. To read the notebooks easily, however, one will need a working copy of *Mathematica*. If *Mathematica* is not available, it may be possible to read the notebooks by using the "Wolfram Player." The Wolfram Player may allow one to view *Mathematica* notebooks, but it may not allow a person to actually run the commands in the notebooks. For further information on the Wolfram Player, visit

`http://www.wolfram.com/player`

To run my notebooks, one will need a computer and a working copy of *Mathematica* on that computer. I ran all twelve notebooks on a Linux system running *Mathematica* 11.0.1 and again on a Microsoft Windows system running *Mathematica* 12.1.1. The notebooks may not run correctly with older or newer versions of *Mathematica*. See the section on "Future versions of *Mathematica*" later on in this document.

To run the notebooks, one should download the notebooks and all of the other files that accompany this document. The main reason for downloading the notebooks and the accompanying files is the fact that all twelve notebooks contain commands which ask *Mathematica* to read in external files. These commands contain the locations, or directory paths, of the external files. If the commands are to work effectively, one will likely need to first download the notebooks and the external files.

My *Ricci* notebooks and files, as well as the file `Ricci1.61.m`, are all located in one combined file located on the University of North Dakota Scholarly Commons. I discussed the Scholarly Commons above. After arriving at the Scholarly Commons, one should go to "Advanced Search," open up two search boxes, specify "Author" for each box, type "Lawrence" and "Peterson" into the respective boxes, and then click on "Search."

If one wishes to download the combined file, *one needs to do this carefully*. One would download the combined file and "extract" the twelve notebooks and the other files from it. This extraction process would create a new directory (or folder) structure on one's computer or storage device. This new directory structure would contain all of the notebooks and other files. There is a slight danger, however, that the downloading and extraction process will overwrite previously existing files on the computer. The file extraction process will create a new directory with the name `yme`. This could cause a problem if one's computer has a previously existing file, directory, or folder with this name. To reduce the chances of problems, one should create a new directory on his or her computer. One should do this before downloading the combined file. One should then download the combined file by clicking on the appropriate button on the Scholarly Commons page. One should store the combined file in the new directory and then perform the extraction process.

The name of the combined file is `yme.zip`. After downloading `yme.zip`, a Linux user should be able to use the `unzip` command to extract the notebooks and other files from the `yme.zip` file. The `unzip` command takes the files in `yme.zip` and uses them to create a directory structure on the user's computer. The new directory structure contains the extracted files. I executed the `unzip` command on my computer as follows. First, I opened a window containing the command prompt. I then navigated to the directory on which I had saved the `yme.zip` file. Finally, I entered the following command at the command prompt:

```
unzip yme.zip
```

This command created a new directory structure within the directory containing the `yme.zip` file. This new directory structure contained my twelve notebooks and the other related files.

It may also be possible for a Linux user to extract the files from `yme.zip` by working through a directory navigator and file viewer program. In doing this, one still needs to be aware of my above warning concerning the overwriting of existing files on the computer. In any case, the user would navigate to the directory containing the `yme.zip` file and then click on the file name ("`yme.zip`").

Users of Microsoft Windows may also be able to extract the files by working with a file navigator program. If the Windows user clicks on the file `yme.zip`, a menu for "Extract" may appear. By choosing the correct options, it may be possible to have Windows extract the files and create a directory or folder structure containing the files. The same warning applies.

Similar procedures and warnings may apply to other operating systems.

As I noted above, all twelve of my notebooks contain commands to read in other files, namely one or more of the files one extracted from `yme.zip` above. These file-reading commands begin with the characters "`<<`". When one encounters such commands, it will be obvious that the commands are commands asking *Mathematica* to read in other files. One may need to edit these commands to ensure that they are consistent with the directory (or folder) structure on the computer. The characters "`<<`" indicate the command to read in a file. The characters following the `<<` characters are the directory path and file name of the file that *Mathematica* is to read in. The semicolon character ("`;`") is a delimiter which marks the end of a command. The user should *not* alter the characters "`<<`" or the semicolon. After the user edits the directory paths, the file-reading commands may still not work properly. If one encounters problems running these edited commands, it may help to delete the first few characters at the beginning of the commands and then type them in again. After one completes this process, the commands should work properly.

4. Warning Concerning the Default Output Format Type

If one chooses to run any of my twelve notebooks, one should set *Mathematica*'s default output format type to "`OutputForm`." This will allow *Mathematica* to format *Ricci* output properly. Each of my twelve notebooks contains a command which sets the default output format type to "`OutputForm`." When one uses this command, however, one needs to be careful. *Mathematica* may record the value of the default output format type in its configuration files. In the future, when a person starts a new work session with *Mathematica*, *Mathematica* may use the default output format type that it recorded during the previous *Mathematica* work session. **If this default output format type is "OutputForm," then, in the future work session, *Mathematica* may not display graphical output.** This could be a problem for people who use *Mathematica* for purposes other than work with *Ricci*.

To allow *Mathematica* to display graphical output in future *Mathematica* work sessions, it is very important to set the default output format type to either "`TraditionalForm`" or "`StandardForm`." (With *Mathematica* 11.0.1, one may also set the default output format type to "`InputForm`," but this may produce undesirable results.) One may change the default output format type at the conclusion of a work session with any of my twelve notebooks. Each of my twelve notebooks contains commands to set the default output format type to either "`TraditionalForm`" or "`StandardForm`." These commands appear at the end of each notebook. Alternatively, one may update the default output format type at the beginning of a future work session with *Mathematica*.

Further information appears near the beginning of the notebook `FGI.nb`. The reader should read the special warning statement at the beginning of `FGI.nb`.

5. Future Versions of *Mathematica*

I ran all twelve of my *Ricci* notebooks on an Ubuntu Linux system with *Mathematica* 11.0.1. After making very slight adjustments to the notebooks, I then ran them on a Microsoft Windows system with *Mathematica* 12.1.1. Future versions of *Mathematica* may not be compatible with *Ricci*. If a compatibility problem should occur, *Mathematica* will likely display an error message at the time that one runs the command to read in the file `Ricci1.61.m`. If one encounters compatibility problems involving newer versions of *Mathematica*, one should consult John M. Lee's Web site to see if he has developed a newer updated version of *Ricci* that is compatible with the latest version of *Mathematica*. See [5]. If a suitable updated version of *Ricci* is not available, it may be possible for users to solve compatibility problems by modifying the *Ricci* source code themselves. But before doing this, users should read the comments at the top of the file `Ricci1.61.m`. It may also be possible to solve the compatibility problems by running appropriate *Mathematica* commands before one runs the command to read in the file `Ricci1.61.m`.

Compatibility problems may occur if new versions of *Mathematica* contain "reserved" or "protected" words that match identifiers in the *Ricci* source code. To solve such compatibility problems, it may be helpful to begin by studying the comments and code near the top of the `Ricci1.61.m` file. If one modifies the *Ricci* source code, one should be sure to document the changes in the manner described at the top of the `Ricci1.61.m` file.

The software that accompanies this document (the notebooks and the files whose names end in "`.m`") includes many files other than `Ricci1.61.m`. These other files also contain definitions that may be incompatible with future versions of *Mathematica*. If these files cause compatibility problems in the future, it may be possible to solve these problems by using techniques similar to the ones I have described above.

6. Contents of Files

This section gives a brief description of the contents of each of my twelve *Mathematica* notebooks. It also discusses some of the other files that accompany this document. The names of all of these other files end in the suffix "`.m`." All of the notebooks are contained in the directory (or folder)

```
~/yme/nbook
```

Here and below, the reader may have to replace the tilde symbol with an appropriate directory path. This may depend on the user's operating system and the directory structure on the user's computer. The files whose names end with the suffix "`.m`" are in other directories.

I will begin by listing the name of each notebook along with a brief description of the contents of the notebook. In what follows, notation is as in our current article, namely [4].

FGI.nb: This notebook assembles the Fefferman-Graham invariant, i.e. the scalar conformal invariant of Proposition 3.4 of [2]. It does so by using the formula for this invariant given in [2]. It then assembles this invariant a second time by using the formula given in [1]. In each case, the notebook compares the result against an expression we previously computed and saved in the file

```
~/yme/dmfiles/FGI.m.
```

NabBach.nb: This notebook shows that $\nabla_a B^a{}_c + 2P^{ek} A_{ekc} = 0$ in dimension 6.

ObsDesc.nb: In this notebook, we establish the first displayed equation in the statement of Theorem 6.1 of the first arXiv.org version of our article. This equation reads as follows:

$$(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E = (X_E Z^{Dd} - X^D Z_E{}^d) 16\mathcal{O}^6_{cd}\,.$$

In deriving this equation, we assemble an expanded symbolic expression for $(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E$. We do this by combining previously computed expansions of the five summands that we discuss in our descriptions of the notebooks Summand1.nb, Summand2.nb, and Summand3.nb, and Sum4and5.nb below. We read these previously computed symbolic expressions in from the following files:

```
~/yme/dmfiles/DeldDelOm.m
~/yme/dmfiles/DelPHOm.m
~/yme/dmfiles/DelJOm.m
~/yme/dmfiles/BDelOmOm.m
```

In the course of establishing the above equation for $(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E$, we also assemble $(X_E Z^{Dd} - X^D Z_E{}^d)16\mathcal{O}^6_{cd}$; when we do this, we use our macro for $\mathcal{O}^6_{cd}$. The name of this macro is `ObsD6`. We test this macro in the notebook ObsTen.nb.

ObsTen.nb: Here we construct $\mathcal{O}^6_{ab}$, the Fefferman-Graham obstruction tensor in dimension 6. In doing this, we use the symbolic formula for this tensor given in display (61) of [3]. In the notebook ObsTen.nb, we also test our macro for the obstruction tensor in dimension 6. The name of this macro is `ObsD6`.

PHat.nb: This notebook verifies our equation for the transformation of the Schouten tensor under conformal change of metric. This equation reads as follows:

$$\widehat{P}_{ab} = P_{ab} - \nabla_a \Upsilon_b + \Upsilon_a \Upsilon_b - \frac{1}{2}\boldsymbol{g}_{ab}\Upsilon_c \Upsilon^c.$$

Sum4and5.nb: In the proof of Theorem 6.1 of the first arXiv.org version of our article, we express $(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E$ as the sum of five summands. In the notebook Sum4and5.nb, we compute the sum of the fourth and fifth of these summands. Specifically, we expand the following:

$$-(\delta\Omega)_b{}^D{}_G \Omega^b{}_c{}^G{}_E + \Omega^b{}_c{}^D{}_G (\delta\Omega)_b{}^G{}_E.$$

We compare the result against an expression we previously saved in the file

`~/yme/dmfiles/BDelOmOm.m.`

Here the "B" in the file name stands for "bracket."

Summand1.nb: In the proof of Theorem 6.1 of the first arXiv.org version of our article, we express $(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E$ as the sum of five summands. In the notebook Summand1.nb, we expand the first of these summands, namely $(\delta d \delta\Omega)_c{}^D{}_E$. We compare the result against an expression we previously saved in the file

`~/yme/dmfiles/DeldDelOm.m.`

Summand2.nb: In the proof of Theorem 6.1 of the first arXiv.org version of our article, we express $(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E$ as the sum of five summands. In the notebook Summand2.nb, we expand the second of these summands, namely $-4(\delta(P\#\Omega))_c{}^D{}_E$. We compare the result against an expression that we previously saved in the file

`~/yme/dmfiles/DelPHOm.m.`

Summand3.nb: In the proof of Theorem 6.1 of the first arXiv.org version of our article, we express $(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E$ as the sum of five summands. In the notebook Summand3.nb, we expand the third of these summands, namely $2(\delta(J\Omega))_c{}^D{}_E$. We compare the result against an expression that we previously saved in the file

`~/yme/dmfiles/DelJOm.m.`

WeylCInv.nb: In this notebook, we do the following:

(1) We establish equation (36) of the first arXiv.org version of our article. This equation reads as follows:

$$\left\langle F_A, Q_2^A F_A \right\rangle = \\ 4A_{abc}\nabla^b P^{ac} - JC_{abcd}C^{abcd} + 4C_{abcd}\nabla^d\nabla^b P^{ac} + 4P_{ab}C^a{}_{cde}C^{bcde}.$$

(2) We establish the first displayed equation that follows equation (36) in the first arXiv.org version of the article. This equation reads as

follows:

$$\mathcal{S}(A) = \int_M (8A_{abc}\nabla^c P^{ab} - JC_{abcd}C^{abcd} + 4P_{ab}C^a{}_{cde}C^{bcde})\, d\mu.$$

(3) We establish equation (53) of the first arXiv.org version of our article. This equation reads as follows:

$$FGI = \\ 2\,\langle F_A, Q_2^A F_A\rangle + 8C_{abcd}C^a{}_e{}^c{}_i C^{bedi} - 4C_{abcd}C^a{}_e{}^c{}_i C^{bide} \\ + (\textit{divergences}).$$

Here FGI denotes the Fefferman-Graham invariant of Proposition 3.4 of [2]. When we establish the above equation, we assemble FGI by simply reading it in from the file

```
~/yme/dmfiles/FGI.m
```

XZSlot.nb: We do two main things in this notebook. First, we assemble the XZ-terms of $(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E$. We do so by keying in rightmost member of the equation in Figure 4 of the first arXiv.org version of our article. We also assemble $(X_E Z^{Dd} - X^D Z_E{}^d)16\mathcal{O}^6_{cd}$; in doing so, we use our macro `ObsD6`. We then work in dimension 6 and show that the XZ-terms of $(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E$ are equal to $(X_E Z^{Dd} - X^D Z_E{}^d)16\mathcal{O}^6_{cd}$.

The other main thing we do in the notebook XZSlot.nb is similar to the first thing we do. Specifically, we again assemble the XZ-terms of $(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E$, but we do so in a different way. We assemble an expression for $(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E$ by combining previously computed expressions for the five summands we discussed above in our descriptions of the notebooks Summand1.nb, Summand2.nb, Summand3.nb, and Sum4and5.nb. We obtain these previously computed expressions by reading them in from the following files:

```
~/yme/dmfiles/DeldDelOm.m
~/yme/dmfiles/DelPHOm.m
~/yme/dmfiles/DelJOm.m
~/yme/dmfiles/BDelOmOm.m
```

After we assemble the expression for $(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E$, we then isolate the XZ-terms in this expression. We also assemble $(X_E Z^{Dd} - X^D Z_E{}^d)16\mathcal{O}^6_{cd}$; to do this, we use our macro `ObsD6` as before. We then work in dimension 6 and show that the sum of the XZ-terms is equal to $(X_E Z^{Dd} - X^D Z_E{}^d)16\mathcal{O}^6_{cd}$.

ZZSept20.nb: In this notebook, we show that the ZZ-terms of $(\mathfrak{D}\nabla^{\mathcal{T}})_c{}^D{}_E$ vanish. Specifically, we show that the expression on the right-hand side of the equation in Figure 3 of the first arXiv.org version of the article is zero.

In addition to the twelve *Mathematica* notebooks, many other files accompany this document. The names of all of these other files end in the suffix "`.m`." Each *Mathematica* notebook contains several commands which instruct *Mathematica* to read in the contents of some of the `.m` files. The `.m` files in turn contain commands for *Mathematica* and *Ricci*. The `.m` files are in various directories. We discuss the files in the various directories below.

The directory

`~/yme/dtm3`

contains three files, which we describe below.

- `Ricci1.61.m`: This file contains the code for the *Ricci* software package. A command for *Mathematica* to read this file appears near the beginning of each of the twelve notebooks. When the user has *Mathematica* read this file, he or she is effectively "starting" the *Ricci* software package.
- `June8tan.m`: This file contains a command to define the tangent bundle of the given manifold to *Ricci*.
- `March19.m`: Contains definitions to exploit the symmetry of the second-order covariant derivative of a density.

The files in the directory

`~/yme/dmfiles`

contain symbolic tensor formulae. When *Mathematica* notebooks read in these files, *Ricci* interprets them and constructs its own symbolic representations of the symbolic tensor formulae. Many notebooks contain their own commands for generating symbolic tensor formulae. After a notebook constructs a symbolic tensor formula, it may read in a previously computed version of the formula from an external `.m` file and compare it against the version that it has just constructed. The comparison serves as a partial verification of the correctness of both the new and the old versions of the symbolic tensor formula. Other notebooks may read in a symbolic tensor expression from an external file without actually constructing the formula itself. Thus some notebooks are able to use symbolic tensor formulae that we have constructed in other notebooks.

The directory

`~/yme/dtm`

contains files relating to *Mathematica*'s default output format type. When a notebook reads one of these files in, *Mathematica* sets its default output format type to the type indicated in the file. The above directory also contains the file `PrintTime.m`. This file defines a command that a notebook can later use to print the current time and date. This can be helpful when a notebook is performing computations that take a long time to run.

The directory

`~/yme/dtm2`

contains several files, which we describe below.

- `Maybund.m`: Contains a command defining the tractor bundle for computations with the *Ricci* package.
- `Maybase.m`: Contains basic definitions relating to the tractor splitting operators Y^B, Z^{Bc}, and X^B. We use these definitions directly in many of our notebooks.

- `MayCottBach.m`: Contains definitions relating to the Cotton tensor (the Cotton-York tensor) and the Bach tensor.
- `May2.m`: Contains the definition of a density f, as well as definitions relating to the Shouten and Weyl tensors and the tractor curvature. It also contains definitions of some tractor operators that we do not use in this project.
- `May3.m`: Contains the definition of a rule which our notebooks use to set the traces of the Weyl tensor equal to zero.
- `May6.m`: Contains the definitions of many rules that our notebooks use in their manipulations of tensor expressions. The purpose of some of these rules is to change the order in which *Mathematica* arranges certain factors in tensor expressions.
- `May7.m`: Contains the definitions of many rules and commands that notebooks can use to manipulate tensor expressions. One of the most important definitions in this file is the definition of the function `ToWeylRho`. (Here "rho" is the Schouten tensor.) The function `ToWeylRho` converts the Riemannian curvature tensor, the Ricci tensor, and the scalar curvature into expressions involving the Weyl tensor and the Schouten tensor. Many of the other definitions in this file are for rules and commands that we do not use in this project very often.

The directory

```
~/yme/dtm3b
```

Contains one file, namely `July19.m`. This file contains a definition of the tensor `ObsD6`. This is the macro we use in *Ricci* to denote $\mathcal{O}^6_{ab}$, i.e. the Fefferman-Graham obstruction tensor in dimension 6. The file also contains a rule to expand `ObsD6` and express it in terms of the Cotton, Bach, and Weyl tensors. The file `July19.m` also contains the definition of a command to expand the tractor curvature and express it in terms of the Schouten tensor, the Weyl tensor, and the tractor splitting operators.

The twelve notebooks also use several commands defined by the *Ricci* package itself. For example, the notebooks use the following commands: `CommuteCovD`, `DefineRule`, `FactorConstants`, `NewDummy`, `SuperSimplify`, `TensorExpand`, and `TensorSimplify`. These commands are all defined in the *Ricci* software package. Further information on these commands is available in the *Ricci* user's manual and from the *Ricci* package itself. I discuss these sources of information in Section 1, above.

7. Notational Conventions

Our article, namely the article [4] of Gover, Peterson, and Sleigh, discusses the Fefferman-Graham invariant of Proposition 3.4 of [2] as well as the Fefferman-Graham obstruction tensor of the same paper. In the notebook FGI.nb, I construct an explicit formula for the Fefferman-Graham invariant. I do so by using the explicit formula for this invariant given in [2]. To facilitate the understanding of my construction of this explicit formula, we will discuss some of the sign conventions and notational conventions of [2]. We will do so in this section. The notebook

- The Riemannian curvature tensor of FG is the negative of ours.
- The Ricci tensor and scalar curvature of FG are the same as ours.
- The tensor A_{jk} of FG is the same as our Schouten tensor P_{jk}.
- The tensor C_{jkl} of FG is the same as our tensor $-A_{jkl}$. Here A_{jkl} is the Cotton tensor.
- The Weyl tensor of FG is the negative of ours.

FIGURE 1. The notational conventions of [2].

FGI.nb also constructs the Fefferman-Graham invariant by using the explicit symbolic formula for this invariant given in [1]. For this reason I will comment on the sign conventions of [1] below. Finally, the notebook ObsTen.nb contains a construction the Fefferman-Graham obstruction tensor in dimension 6. The construction uses the symbolic formula for this tensor given in [3], so we will discuss the various sign conventions of [3] as well.

We begin with the conventions of [2]. A comparison of some of our conventions with those of [2] appears in Figure 1. In this figure, "FG" refers to [2] and "our" refers to [4].

Now consider the article [1] of Čap and Gover. The authors of this article define the Riemannian curvature tensor and the Ricci tensor on pages 248 and 247 of the article, respectively. Their definitions agree with ours. In my construction of the Fefferman-Graham invariant, I assume that the scalar curvature of [1] is the same as ours. Čap and Gover define the Schouten tensor on page 257 of [1]. They refer to the Schouten tensor as the rho-tensor. If we assume that their scalar curvature is the same as ours, then their Schouten tensor is the same as ours. Čap and Gover define the Cotton tensor (the Cotton-York tensor) on page 258 of their article. They let C_{abd} denote this tensor. Inspection of the definition of C_{abd} in [1] shows that $C_{abd} = A_{dab}$, where A_{dab} is our Cotton tensor. In my work in the notebook FGI.nb, I assume that the Weyl tensor of Čap and Gover is the same as ours. My work in this notebook shows that the formula for the Fefferman-Graham invariant in [1] agrees with the formula for this invariant given in [2].

The authors of [3] define the Laplacian, the Riemannian curvature tensor, Weyl tensor, the Schouten tensor, and the tensor J on page 313 of their article. Their definitions agree with ours. They define the Cotton tensor and the Bach tensor on pages 326 and 347 of their article. Again, their definitions agree with ours.

REFERENCES

[1] A. Čap, A.R. Gover: *Standard tractors and the conformal ambient metric construction.* Ann. Global Anal. Geom. **24** (2003), no. 3, 231-259.

[2] C. Fefferman, C.R. Graham: *Conformal invariants.* In: Elie Cartan et les mathématiques d'aujourd'hui, 95-116. Astérisque, Numeréro Hors Série, Société Mathématique de France, Paris, 1985.

[3] A.R. GOVER, L.J. PETERSON: *The ambient obstruction tensor and the conformal deformation complex.* Pacific J. Math. **226** (2006), no. 2, 309-351.

[4] A.R. Gover, L.J. Peterson, C. Sleigh: A conformally invariant Yang-Mills type energy and equation on 6-Manifolds. arXiv:2107.08515.

[5] J.M. Lee: *Ricci* software package.

`https://sites.math.washington.edu/~lee`

[6] Wolfram Research, Inc.: *"Mathematica" computer software.* Champaign, Illinois, 2016, Version 11.0.

Department of Mathematics, The University of North Dakota, 101 Cornell Street Stop 8376, Grand Forks, ND 58202-8376, USA

Email address: `lawrence.peterson@email.und.edu`