A Newman-Penrose Calculator for Instanton Metrics

T. Birkandan*
Istanbul Technical University,
Faculty of Science and Letters,
Dept. of Physics, Maslak 34469, Istanbul, Turkey

Abstract

We present a Maple11+GRTensorII based symbolic calculator for instanton metrics using Newman-Penrose formalism. Gravitational instantons are exact solutions of Einstein's vacuum field equations with Euclidean signature. The Newman-Penrose formalism, which supplies a toolbox for studying the exact solutions of Einstein's field equations, was adopted to the instanton case and our code translates it for the computational use.

Keywords : Gravitational Instantons; Newman-Penrose formalism; Symbolic computation

PACS: 04., 04.62.+v

1 Introduction

The interest in symbolic computational study of general relativity is growing rapidly as the capacity of the the computer systems increase. The computer is no longer an apparatus for the numerical relativist, but as the symbolic manipulators get easier to use, the more researchers get into the subject by using these systems. Thanks to the symbolic calculators, lengthy calculations needing time and care are no longer a problem for scientists. The preliminary

*E-mail : `birkandant@itu.edu.tr`

works on symbolic calculation goes back to 1965, Fletcher's program GRAD ASSISTANT could calculate Ricci tensor for simple metrics [1]. Two years later, in 1967, M. Veltman developed SCHOONSCHIP, a symbolic manipulation program to study renormalizability of gauge theories [2]. At 1968, Macsyma, the first comprehensive symbolic mathematics system, began to be developed by MIT scientists under the leadership of B. Martin [3]. Following the path of SCHOONSCHIP and Macsyma, a group leaded by S. Wolfram began designing SMP which is the ancestor of today's Mathematica [4]. A historical review can be found in reference [5]. Recent platforms like Maple [6], Mathematica [7], Matlab [8], Maxima [9] and Reduce [10] offer developed and optimized solutions with their easy to learn interfaces. There are more alternatives and all these platforms have their advantages and disadvantages. Therefore, the user should make a decision before getting deeper into programming. Another important point to consider is the specialized packages which run under these platforms. Using a profession based package reduces the work of the end-user. A package having a nice designed interface can be used even by the people having no idea about the platform. There are such packages for the general relativistic purposes. Some of the best known examples are: GRTensorII [11], Riemann [12], Riegeom [13], Riccir [14] and MathTensor [15]. GRTensorII, being a widespread package led significant progress in the area [16][17][18][19][20][21].

The Newman-Penrose (NP) formalism supplies a toolbox for investigating the exact solutions of the Einstein's field equations [22][23]. Goldblatt has developed NP formalism for gravitational instantons based on the $SU(2) \times SU(2)$ spin structure of positive definite metrics [24][25]. In the gravitational instanton case, the gravitational field decomposes into its self-dual and anti-self-dual parts and this decomposition is natural in the spinor approach which necessitates two independent spin frames for the spinor structure of 4-dimensional Rieman manifolds with Euclidean signature [26].

Gravitational instantons are exact solutions of Einstein's vacuum field equations with Euclidean signature. They are analogous to the Yang-Mills instantons, the finite action solutions of the classical Yang-Mills equations. They admit hyper-Kähler structure [26][27]. For a detailed overview one can see Eguchi et al's review and the articles cited in this paper [28].

Aliev and Nutku applied differential forms to the NP formalism for gravitational instantons which made the formalism more suitable for the symbolic computation [26]. The main source of our work will be the Aliev and Nutku's paper which will be referred as **I**.

2 The program

Our program, NPInstanton, consists of procedures calculating physical and mathematical quantities for instanton metrics using Newman-Penrose formalism. It is coded under Maple 11 and GRTensorII package. The procedures calculate these quantities:

- massless scalar equation,
- massless Dirac equation,
- source-free Maxwell equations,
- covariant and contravariant Dirac γ matrices,
- coframe $l(= l_\mu dx^\mu)$ and $m(= m_\mu dx^\mu)$,
- spin rotation coefficients,
- Weyl scalars,
- trace-free Ricci scalars,
- spinor equivalent of the connection 1-forms,
- basis 2-forms,
- curvature 2-forms,
- integrands of the Euler number and the Hirzebruch signature curvature part integrals,
- Petrov class of the spacetime

As one can see, some of the objects could be calculated by standard means and without using a signature-dependent package. But for the sake of completeness, we added these features to the program. By these, the program becomes a complete symbolic calculator for an instanton metric. The NP calculator of GRTensorII is not designed for the Euclidean signature and could give unexpected results. Therefore, a complete NP based calculator, combining the power of GRTensorII and NP formalism for these special metrics is useful.

Throughout the program, only the output variables of the procedures are the global ones, so, naturally, the user should be aware of the output names of the procedures that are being used. A few commands for simplification are added to the code which can be evaluated easily by an average personal computer (i.e. having a 512 Mb of memory), but it is always more convenient for the user to choose the right simplification technique for the problem after the calculation. The output must be regarded as a "raw material" for a simplification routine that is to be chosen by the user. The user having a computer with insufficient memory can extract the simplification routines from the program, simply by modifying it in an editor but it is not recom-

mended as it may lead to miscalculation of some properties such as Petrov type or (anti-)self-duality.
One can reach the program files using the web site given in the "Final Remarks" section.

2.1 Requirements

Any system that can run Maple 11 and GRTensorII is able to run NPInstanton.
The program uses,

1. GRTensorII Package: The famous and widely used free general relativity package. It has two versions, the one which NPInstanton uses works under Maple and a limited version for Mathematica is available. The developers of GRTensorII are Peter Musgrave, Denis Pollney and Kayll Lake [11]. Several objects (Ricci scalar, covariant Weyl tensor, etc.) are calculated by this package in our program. The package also has a powerful NP calculator but it is not designed for the metrics with Euclidean signature.

2. DifferentialGeometry Package of Maple 11: One of the most important new features of Maple 11 is the new package: "DifferentialGeometry". It is more convenient to use this package with linear algebraic quantities. The rather old "difforms" package would be another solution but it has some conflicts with linear algebraic quantities which are the major elements of our program. The DifferentialGeometry package is based upon the Vessiot package developed by I. M. Anderson, Florin Catrina, Cinnamon Hillyard, Jeff Humphries, Jamie Jorgersen, and Charles Miller at Utah State University. The redesign and expansion of Vessiot to DifferentialGeometry for Maple 11 was done by I. M. Anderson and E. S. Cheb-Terrab [29]. The definition of the wedge product is supplied by this package in our program.

3. linalg package of Maple: This rather old internal package is used for an eigenvalue calculation in Petrov classification section.

The program is set for a computer having a 512 Mb of RAM (Average value for today's personal computers). If the system has less memory, the user must change the line

```
kernelopts(gcfreq=10^7):
```

of the npinstanton.mws file to a lower value (10ˆ6 is the standard value of Maple). For a computer having larger than 1 Gb of memory, the user may change the gcfreq value as 10ˆ8. "gc" is the abbreviation of "garbage collection" and it is the Maple's internal routine which cleans the memory after an amount of memory is allocated. For a computer having a large amount of memory, one can increase the frequency of this process. The larger the gc frequency value results in more memory to be wasted but for a system having a large amount of memory it increases the performance for some calculations.

2.2 Running the program

The program consists of several procedures for calculations. The user runs the procedures by calling some specific commands and the output will be available for further calculations after the procedure is finished.

The program comes with four files:
i. NPInstanton program (npinstanton.mws)
ii. Sample input file (eguchihanson)
iii. Sample metric file in the GRTensorII format (eguchihanson.mpl) (This file should be copied into the GRTensor's metric directory.)
iv. README file containing short definitions of the commands (README.txt)

Sample input file should be in "C:/npinstanton" directory. If the user wishes to change this location permanently, in the NPInstanton program,

```
currentdir("C:/npinstanton"):
```

line should be changed in the appropriate way. Or the user may choose to give the path of the file after running the program. Changes to the npinstanton.mws file should be made using an editor such as notepad. The changes should not be made within Maple as, on saving the file, the Maple editor adds a "signature" which causes errors when executing the worksheet.
To run NPInstanton:
1. Open a Classic worksheet in Maple. Standard worksheet can also be used technically but GRTensorII suggests Classic worksheet.
2. Run NPInstanton from its location by read command.
For example,

```
read"C:/npinstanton/npinstanton.mws";
```

runs the program from "C:/npinstanton/" directory.

3. Enter the input file as requested. No ";" or ":" is required after the file name. Now, the program will load GRTensorII package, DifferentialGeometry package, information from the input file and check the Newman-Penrose legs. After the program checks the Newman-Penrose legs, the user will be prompted as

```
npinstanton>
```

and the session is ready for calculations. Most of the calculations are finished in seconds (even less than a second) but some spacetimes may have properties that may need more time to finish the calculation.

2.3 The input file

A sample input file containing the information about the Eguchi-Hanson instanton [28] is included in the package. The definitions should be given using Maple's syntax. The lines to be used will be explained below using the sample input file:

```
metric:=eguchihanson:
```

Enter the name of the metric file here. The metric file of the Eguchi-Hanson spacetime is included in the package. This metric file should be placed in the GRTensorII's metric directory.

```
# Enter the general definitions here:
assume(a::real):
A:=sqrt(1-(a/r)^4):
```

This place is reserved for the user's personal definitions. The constants, assumptions, etc. should be set here.

```
l_covar[1]:=1/(sqrt(2)*A):
l_bar_covar[1]:=1/(sqrt(2)*A):
l_covar[2]:=0:
l_bar_covar[2]:=0:
```

```
l_covar[3]:=-I*r*A*cos(theta)/(2*sqrt(2)):
l_bar_covar[3]:=I*r*A*cos(theta)/(2*sqrt(2)):
l_covar[4]:=-I*r*A/(2*sqrt(2)):
l_bar_covar[4]:=I*r*A/(2*sqrt(2)):
m_covar[1]:=0:
m_bar_covar[1]:=0:
m_covar[2]:=r*exp(-I*xi)/(sqrt(2)*2):
m_bar_covar[2]:=r*exp(I*xi)/(sqrt(2)*2):
m_covar[3]:=I*r*exp(-I*xi)*sin(theta)/(sqrt(2)*2):
m_bar_covar[3]:=-I*r*exp(I*xi)*sin(theta)/(sqrt(2)*2):
m_covar[4]:=0:
m_bar_covar[4]:=0:
```

The components of the covariant NP legs are entered here. l_covar[1] being the first component of the covariant l_μ leg and l_bar_covar[1] being the complex conjugate of l_covar[1], etc.. Maple does not do the complex simplifications because they need assumptions that may cause wrong calculations. Therefore, the most appropriate way to define the legs is to give the both by hand. This definitions are the Maple forms of the Eguchi-Hanson NP legs [28]:

$$l = \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{1-\frac{a^4}{r^4}}}dr - \frac{ir}{2}\sqrt{1-\frac{a^4}{r^4}}(d\xi + \cos\theta d\phi)\right], \tag{1}$$

$$m = \frac{re^{-i\xi}}{2\sqrt{2}}\left(d\theta + i\sin\theta d\phi\right). \tag{2}$$

The A value being $\sqrt{1-\frac{a^4}{r^4}}$ was given in the general definitions section.

```
spinorcomponent1:=exp(I*(m+(1/2))*xi)*psi1[r,theta,phi]:
spinorcomponent2:=exp(I*(m+(1/2))*xi)*psi2[r,theta,phi]:
spinorcomponent3:=exp(I*(m-(1/2))*xi)*psi3[r,theta,phi]:
spinorcomponent4:=exp(I*(m-(1/2))*xi)*psi4[r,theta,phi]:
```

Spinor components can be given here if the user will be calculating the Dirac equation. They can be choosen as,

$$\psi_1 = e^{i(m+\frac{1}{2})\xi}\Psi_1(r,\theta,\phi), \tag{3}$$

$$\psi_2 = e^{i(m+\frac{1}{2})\xi}\Psi_2(r,\theta,\phi), \tag{4}$$

$$\psi_3 = e^{i(m-\frac{1}{2})\xi}\Psi_3(r,\theta,\phi), \tag{5}$$

$$\psi_4 = e^{i(m-\frac{1}{2})\xi}\Psi_4(r,\theta,\phi) \tag{6}$$

in the traditional form for the Eguchi-Hanson space [30].

```
scalarfunction:=exp(I*m*xi)*Phi(r,theta,phi):
```

The scalar function can be given here for the calculation of the scalar equation. This definition can be skipped if the used does not need to calculate this object. It can be choosen as

$$\varphi = e^{im\xi}\Phi(r,\theta,\phi) \tag{7}$$

for the Eguchi-Hanson space, φ being the scalar function.

2.4 Command definitions

The list of commands can be given as,

```
scalaroperator()
dirac()
maxwell()
gammamatrices()
coframe()
spinrotcoeff()
weylscalar()
tfricciscalar()
conn1form()
basis2form()
curv2form()
topologicalnumbers()
```

and the definitions of these commands are the following,

- scalaroperator():

This command calculates the massless scalar equation, finding the scalar operator. The scalar operator is given as

$$H\varphi \equiv \frac{1}{\sqrt{|g|}}\partial_\nu\sqrt{|g|}g^{\mu\nu}\partial_\mu\varphi. \tag{8}$$

Here, g is the determinant of the metric.
The procedure takes the scalar function (name: scalarfunction) from the input file.
The output to be used thereafter:

```
> scalarop;
```

For the massive case, one can equate this object to $M^2\varphi^2$, M being the mass of the scalar particle. As an additional property, this procedure is not dependent on the metric signature and it does not use the NP objects so it can be used for any spacetime in four dimensions by extracting it from the program.

- dirac():

This command calculates the massless Dirac equations,

$$\gamma^\mu\nabla_\mu\psi = 0 \tag{9}$$

where

$$\nabla_\mu = \partial_\mu - \Gamma_\mu \tag{10}$$

and Γ_μ are the spin connections. The procedure takes the spinor vector components from the input file under these names: spinorcomponent1 (ψ_1), spinorcomponent2 (ψ_2), spinorcomponent3 (ψ_3) and spinorcomponent4 (ψ_4). The output to be used thereafter is the components of the "dirac" vector as (i=1, 2, 3, 4):

```
> dirac[i];
```

For the massive case, one can equate these objects to $\frac{M}{i}\psi$ vector, M being the spinor mass and $i \equiv \sqrt{-1}$.

- maxwell():

This command calculates the source-free Maxwell equations using the equations (**I**.102-109). F_{ij}'s in the output are the usual Maxwell field matrix components. The output is set to be "maxwell" vector whose components are the source-free Maxwell equations as (i=1, 2, 3, 4):

```
> maxwell[i];
```

- gammamatrices():

This command finds the covariant and contravariant Dirac γ matrices and tests them with anticommutation relations

$$\{\gamma^{\mu}, \gamma^{\nu}\} = 2g^{\mu\nu}. \tag{11}$$

The output is "gamma_contr" (contravariant γ matrices: γ^{μ}) and "gamma_covar" (covariant γ matrices: γ_{μ}) vectors as (i=1, 2, 3, 4):

```
> gamma_contr[i];
> gamma_covar[i];
```

The calculation of the Dirac γ matrices could be put into the dirac() procedure but for further calculations involving higher dimensions may need just these matrices. To find the γ matrix for the extra dimension, one can use these results and look for the γ matrix of the extra dimension using the anticommutation relations as in reference [31]. Therefore the procedure is a separate one.

- coframe():

This command calculates the coframe $l \equiv$L ($= l_{\mu}dx^{\mu}$) and $m \equiv$M ($= m_{\mu}dx^{\mu}$). The output variables are the following, "_bar" denoting the complex conjugate:

```
> L;
> M;
> L_bar;
> M_bar;
```

- spinrotcoeff():

This procedure calculates the spin rotation coefficients using equations in (**I**.20). The output can be used later by calling:

```
> npkappa; > nptau; > npsigma;
> nprho; > nppi; > npnu;
> npmu; > nplambda; > npgamma;
> npepsilon; > npalpha; > npbeta;
```

in the meaning of the spin rotation coefficients κ, τ, σ, ρ, π, ν, μ, λ, γ, ε, α, β respectively.

- weylscalar():

This command calculates the Weyl scalars using equation (**I**.68). The output variables are:

```
> weylscalar0; > weylscalar1; > weylscalar2, > weylscalar3;
> weylscalar4; > weylscalartilde0; >weylscalartilde1;
> weylscalartilde2; > weylscalartilde3; > weylscalartilde4;
```

for Ψ_0, Ψ_1, Ψ_2, Ψ_3, Ψ_4, $\tilde{\Psi}_0$, $\tilde{\Psi}_1$, $\tilde{\Psi}_2$, $\tilde{\Psi}_3$, $\tilde{\Psi}_4$ respectively. "tilde" means the variable has a tilde in the meaning of a different spin frame.
Another property of this procedure is that it finds out the Petrov class of the spacetime according to equations (**I**.120-122). Petrov classification of Euclidean spacetimes were first studied by Hacyan [33] and then by Karlhede [34].

- tfricciscalar():

This command calculates the trace-free Ricci scalars using equation (**I**.69) and sets them to variables for a later use:

```
> tfricciscalar00; > tfricciscalar01; > tfricciscalar02;
> tfricciscalar10; > tfricciscalar11; > tfricciscalar12;
> tfricciscalar20; > tfricciscalar21; > tfricciscalar22;
> scalofcurv;
```

for Φ_{00}, Φ_{01}, Φ_{02}, Φ_{10}, Φ_{11}, Φ_{12}, Φ_{20}, Φ_{21}, Φ_{22}, scalar of curvature Λ respectively.

- conn1form():

This procedure calculates the spinor equivalent of the connection 1-forms given by equations (**I**.36-37). The output can be reached by calling

```
> GAMMA00; > GAMMA01; > GAMMA10; > GAMMA11;
> GAMMAtilde0pr0pr; > GAMMAtilde0pr1pr;
> GAMMAtilde1pr0pr; > GAMMAtilde1pr1pr;
```

for Γ^0_0, Γ^1_0, Γ^0_1, Γ^1_1, $\tilde{\Gamma}^{0'}_{0'}$, $\tilde{\Gamma}^{1'}_{0'}$, $\tilde{\Gamma}^{0'}_{1'}$, $\tilde{\Gamma}^{1'}_{1'}$ respectively where "tilde" means the variable has a tilde and "pr" means "prime" in the meaning of a different spin frame.
Another property of the procedure is that, it finds out the gauge by checking the necessary and sufficient conditions for (anti-)self-duality namely, $\Gamma_{ab} \equiv 0$ implies self duality and $\tilde{\Gamma}_{x'y'} \equiv 0$ implies anti-self-duality.

- basis2form():

This command finds the basis 2-forms using the definitions in equation (**I**.49). The output can be reached by

```
> L00; > L01; > L10;
> Ltilde0pr0pr; > Ltilde0pr1pr; > Ltilde1pr0pr;
```

for L^0_0, L^1_0, L^0_1, $\tilde{L}^{0'}_{0'}$, $\tilde{L}^{1'}_{0'}$, $\tilde{L}^{0'}_{1'}$ respectively. Here, "tilde" means the variable has a tilde and "pr" means "prime" in the meaning of a different spin frame.

- curv2form():

This command calculates the curvature 2-forms using equations (**I**.90-91) and sets them to these variables:

```
> Theta00; > Theta01; > Theta10;
> Thetatilde0pr0pr; > Thetatilde0pr1pr; > Thetatilde1pr0pr;
```

for Θ^0_0, Θ^1_0, Θ^0_1, $\tilde{\Theta}^{0'}_{0'}$, $\tilde{\Theta}^{1'}_{0'}$, $\tilde{\Theta}^{0'}_{1'}$ respectively where "tilde" means the variable has a tilde and "pr" means "prime" in the meaning of a different spin frame.

- topologicalnumbers():

This command calculates the integrands of the Euler number and the Hirzebruch signature curvature part integrals using the relations (**I**.115-117) namely,

$$\chi = \frac{1}{4\pi^2}\int_{\mathcal{M}} \Big[\, |\,\Psi_0\,|^2 + 4\,|\,\Psi_1\,|^2 + 3\,\Psi_2^2 + |\,\tilde{\Psi}_0\,|^2 + 4\,|\,\tilde{\Psi}_1\,|^2 + 3\,\tilde{\Psi}_2^2 - 2\,(\,|\,\Phi_{00}\,|^2 + |\,\Phi_{02}\,|^2\,) - 4\,(\,|\,\Phi_{01}\,|^2 + |\,\Phi_{11}\,|^2 + |\,\Phi_{12}\,|^2 - 3\Lambda^2\,)\Big]\, l \wedge \bar{l} \wedge m \wedge \bar{m} \qquad (12)$$

$$\tau = -\frac{1}{6\pi^2}\int_{\mathcal{M}} \Big[\, |\,\Psi_0\,|^2 + 4\,|\,\Psi_1\,|^2 + 3\,\Psi_2^2 - |\,\tilde{\Psi}_0\,|^2 - 4\,|\,\tilde{\Psi}_1\,|^2 - 3\,\tilde{\Psi}_2^2\Big]\, l \wedge \bar{l} \wedge m \wedge \bar{m} \,-\eta_s(\partial\mathcal{M}) \qquad (13)$$

$\eta_s(\partial\mathcal{M})$ being the eta-invariant and this value will not be taken into consideration in the program. The output can be reached by calling

```
> eulernumber_integrand;
> hirzebruch_signature_integrand;
```

These numbers have a special importance for the Atiyah-Patodi-Singer index theorem of operators on manifolds with boundary [35][36][37][38].

3 Examples

In this section, we will apply our program to two instanton metrics and calculate some objects using the special commands. Lengthy output values are not written to avoid distracting the reader's attention.

3.1 Example 1: Calculations for the Eguchi-Hanson metric

Eguchi-Hanson instanton [28] is the most similar to the Yang-Mills instanton of Belavin et al. [39] and the metric is given as

$$ds^2 = \frac{1}{1-\frac{a^4}{r^4}}dr^2 + r^2(\sigma_x^2 + \sigma_y^2) + r^2(1 - \frac{a^4}{r^4})\sigma_z^2. \qquad (14)$$

Here,

$$\begin{aligned}
\sigma_x &= \frac{1}{2}(-\cos\xi d\theta - \sin\theta\sin\xi d\phi), \\
\sigma_y &= \frac{1}{2}(\sin\xi d\theta - \sin\theta\cos\xi d\phi), \\
\sigma_z &= \frac{1}{2}(-d\xi - \cos\theta d\phi).
\end{aligned} \tag{15}$$

and the dyad was given in eqn. 1 and eqn. 2. We run our program in Maple and type the name of the input file:

```
> read("C:/npinstanton/npinstanton.mws");
Enter the name of the definition file (include the
path if other than C:/npinstanton): eguchihanson
```

The program now checks the definitions of the NP legs and shows the primary definitions (metric, coordinates, command list). When

```
npinstanton>
```

is prompted the session is ready for calculations. Now, let us calculate the connection 1-forms:

```
npinstanton> conn1form();
```

The program calculates and shows the calculated values. Then the gauge is found by the program as:

```
SELF DUAL GAUGE because all connection 1-forms without tilde are zero
```

before prompting for another calculation, the procedure shows the output variables which can be used for a later calculation. for example, let us call Γ_0^1 of equation (**I**.36):

```
npinstanton> GAMMA01;
```

The scalar operator can be calculated by using the command,

```
npinstanton> scalaroperator();
```

and the output will be the scalar operator. This result can be used by calling the name "scalarop" afterwards.
To find the Weyl scalars and the Petrov class, one can run,

```
npinstanton> weylscalar();
```

After the Weyl scalars are shown, the Petrov type is found to be:

```
Petrov-type D according to anti-self-dual part
```

The whole results aggree with the literature [34].

3.2 Example 2: Calculations for the Nutku helicoid metric

The Nutku helicoid metric is given as

$$\begin{aligned} ds^2 &= \frac{1}{\sqrt{1+\frac{a^2}{r^2}}}[dr^2 + (r^2+a^2)d\theta^2 + \left(1+\frac{a^2}{r^2}\sin^2\theta\right)dy^2 \\ &\quad -\frac{a^2}{r^2}\sin 2\theta dy dz + \left(1+\frac{a^2}{r^2}\cos^2\theta\right)dz^2] \end{aligned} \tag{16}$$

where $0 < r < \infty$, $0 \leq \theta \leq 2\pi$, y and z are along the Killing directions and are taken to be periodic coordinates on a 2-torus [40]. This is an example of a multi-center metric. This metric reduces to the flat metric if we take $a = 0$. Since this solution has curvature singularities, it has not been studied extensively aside from three articles [30][31][32][41]. The NP legs can be chosen as,

$$l^\mu = \frac{a\sqrt{\sinh 2x}}{2}(1, i, 0, 0), \tag{17}$$

$$m^\mu = \frac{1}{\sqrt{\sinh 2x}}(0, 0, \cosh(x - i\theta), i\sinh(x - i\theta)). \tag{18}$$

Let us calculate the basis 2-forms:

```
npinstanton> basis2form();
```

After the calculation, to call L_0^0 of equation (**I**.49),

```
npinstanton> L00;
```

and the output will be

-$\frac{1}{2}Ia^2\cosh(x)\sinh(x)dx1 \wedge dx2$-$\frac{1}{2}Idx3 \wedge dx4$

dx's were defined previously, at the very beginning of the session in terms of the real coordinates. We can calculate the Dirac equation by,

```
npinstanton> dirac();
```

and

```
npinstanton> dirac[3];
```

calls the third component of the Dirac equation vector which can be simplified and used for calculations.
We can calculate the connection 1-forms as,

```
npinstanton> conn1form();
```

For this definition the gauge turns out to be:

```
ANTI-SELF DUAL GAUGE because all connection 1-forms with tilde are zero
```

and the connection 1-forms are shown.
For the Weyl scalars and the Petrov class, one can run,

```
npinstanton> weylscalar();
```

After the Weyl scalars are shown on the screen, the Petrov type is found to be:

```
Petrov-type I according to self-dual part
```

The whole results aggree with the literature [40].

4 Final remarks

We introduced a Maple11+GRTensorII based symbolic calculator consisting of procedures for instanton metrics using a Euclidean Newman-Penrose formalism. The program and sample files are available via the author's webpage:
http://atlas.cc.itu.edu.tr∼birkandant/npinstanton.htm

5 Acknowledgement

The author would like to express his sincere to thanks to Prof. Mahmut Hortaçsu for help and support and Profs. Ayşe H. Bilge and Neşe Özdemir for scientific assistance and Fuat İ. Vardarlı for technical assistance throughout this work. This work is supported by TUBITAK, the Scientific and Technological Council of Turkey.

References

[1] J. G. Fletcher, "GRAD ASSISTANT" *UCRL-14624-T report* (1965)

[2] M. Veltman, "Schoonschip, A CDC 6000 program for symbolic evaluation of algebraic expressions" *CERN preprint* (1967)

[3] Macsyma distribution http://www.symbolics.com/

[4] S. Wolfram, *Commun. of the ACM* **28,** 4 (1985)

[5] I. Cohen, I. Frick, J. E. Åman, "10th International Conference on General Relativity and Gravitation", B. Bertotti et al. (eds.), 139 (1984)

[6] Maple distribution http://www.maplesoft.com/

[7] Mathematica distribution http://www.wolfram.com/

[8] Matlab distribution http://mathworks.com/

[9] Maxima distribution http://www.ma.utexas.edu/maxima.html/

[10] Reduce distribution http://www.uni-koeln.de/REDUCE/

[11] P. Musgrave, D. Pollney and K. Lake, *Fields Institute Commun.* **15**,313 (1996). See also http://www.grtensor.org/

[12] R. Portugal, S. L. Sautú, *Comp. Phys. Commun.* **105**, 233 (1997)

[13] R. Portugal, *Comp. Phys. Commun.* **126**, 260 (2000)

[14] J. Kadlecsik, *Comp. Phys. Commun.* **93**, 265 (1996)

[15] MathTensor distribution http://smc.vnet.net/MathTensor.html

[16] N. Pelavas, N. Neary, K. Lake, *Class.Quant.Grav.* **18**, 1319 (2001)

[17] M. Ishak, L. Chamandy, N. Neary, K. Lake, *Phys. Rev. D* **64,** 024005 (2001)

[18] K. Santosuosso, D. Pollney, N. Pelavas, P. Musgrave, K. Lake, *Comp. Phys. Commun.* **115**, 381 (1998)

[19] M.S.R. Delgaty, K. Lake, *Comp. Phys. Commun.* **115**, 395 (1998)

[20] D. N. Vulcanov, *Comp. Phys. Commun*. **154**, 205 (2003)

[21] P. Musgrave, K. Lake, *Class. Quantum Grav.* **14**, 1285 (1997) and see http://www.grtensor.org/papers.html for more.

[22] E. T. Newman, R. Penrose, *Jour. Math. Phys.* **3**, 566 (1962)

[23] E. T. Newman, R. Penrose, *Jour. Math. Phys.* **4**, 998 (1962)

[24] E. Goldblatt, *Gen. Rel. and Grav.* **26**, 979 (1994)

[25] E. Goldblatt, *Jour. Math. Phys.* **35**, 3029 (1994)

[26] A. N. Aliev, Y. Nutku, *Class. Quant. Grav.* **16**, 189 (1999)

[27] G. W. Gibbons, P. J. Ruback, *Commun. Math. Phys.* **115**, 267 (1988)

[28] T. Eguchi, P. B. Gilkey, A. Hanson, *Phys. Rep.* **66**, 213 (1980)

[29] Maple 11 Help Documentation: "Overview of the DifferentialGeometry Package"

[30] Y. Sucu, N. Ünal, *Class. Quant. Grav.* **21**, 1443 (2004)

[31] T. Birkandan, M. Hortaçsu, *J. Phys. A: Math. Theor.* **40**, 1 (2007)

[32] T. Birkandan, M. Hortaçsu, *J. Math. Phys.* **48**, 092301 (2007)

[33] S. Hacyan, *Phys. Lett. A* **75**, 23 (1979)

[34] A. Karlhede, *Class. Quant. Grav.* **3**, L1 (1986)

[35] M. F. Atiyah, V. K. Patodi and I. M. Singer, *Math. Proc.Camb.Phil. Soc.* **77**, 43 (1975)

[36] M. F. Atiyah, V. K. Patodi and I. M. Singer, *Math. Proc. Camb.Phil. Soc.***77**, 405 (1975)

[37] M. Hortaçsu, K.D. Rothe,B. Schroer, 1*Nucl. Phys. B***17**, 530 (1980)

[38] M. Hortaçsu, *Lettere al Nuovo Cim.* **36**, 109 (1983)

[39] A. A. Belavin, A. M. Polyakov, A. S. Schwarz, Yu. S. Tyupkin, *Phys. Lett. B***59**, 85 (1975)

[40] A. N. Aliev, M. Hortaçsu, J. Kalaycı, Y. Nutku, *Class. and Quant. Grav.* **16**, 631 (1999)

[41] V. M. Villalba, *Jour. of Phys.:Conference Series* **24**, 136 (2005)