ISSN 0029 - 3865

CBPF-NF-058/89

COMPUTER ALGEBRA IN SPACETIME EMBEDDING*

by

Waldir L. ROQUE[†] and Renato P. dos SANTOS[1]

[1]Centro Brasileiro de Pesquisas Físicas - CBPF/CNPq
Rua Dr. Xavier Sigaud, 150
22290 - Rio de Janeiro, RJ - Brasil
E-Mail: RPSOLNCC.BITNET

[†]Departameneto de Matemática, Universidade de Brasília
70910 - Brasília, DF - Brasil
E-Mail: ROQOLNCC.BITNET

*To appear in Journal of Symbolic Computation.

ABSTRACT

In this paper we describe an algorithm to determine the vectors normal to a space-time V_4 embedded in a pseudo-Euclidean manifold M_{4+n}. An application of this algorithm is given considering the Schwarzchild spacetime geometry embedded in a 6 dimensional pseudo-Euclidean manifold, using the algebraic computing system REDUCE.

Key-words: Computer algebra; Embedding; Gravitation.

1 Introduction

The General Relativity Theory defines the physical spacetime as a differentiable pseudo-Riemmanian 4-dimensional manifold (Hawking & Ellis, 1973) . The spacetime is seen through the *intrinsic* geometry. However, many interesting results have shown that the *extrinsic* geometry gives sometimes a better understanding of the physical structure of the spacetime. As an example of its growing interest we can cite the recent studies of Gravity Theories in more than four dimensions and the study of the geometry of extended objects as in String and Membrane Theory.

In contrast to General Relativity, where the metric uniquely specifies the geometry, to describe a spacetime locally and isometrically embedded in a pseudo-Euclidean manifold with dimension $4+n$, two new quantities have to be considered: the *second fundamental form* and the *torsion* vector (or the third fundamental form). These two quantities are well known in the study of differentiable manifolds (Eisenhart, 1946) and come out from the Gauss-Codazzi-Ricci equations, which are the integrability conditions for the existence of the embedding (Maia, 1986).

Rigorously it is not necessary to know the embedding to obtain the second fundamental form and the torsion vector . Both can be obtained directly through the field equations (Maia & Roque, 1989) which are highly non-linear parcial differential equations in the 4 spacetime coordinates. However, if we know the embedding *a priori*, to determine these two quantities we need to get first the set of vectors normal to the spacetime V_4.

In this paper we will be concerned with the discussion of an algorithm which helps to determine these vectors from the embedding coordinates and then to find out the second fundamental form and the torsion vector. The following section sets up the main equations that rule the embedding theory. In section 3 the algorithm for determining the vectors normal to the spacetime V_4 is described. This algorithm has been implemented in the algebraic computing system REDUCE and its application is done in section 4 for the case of Schwarzchild embedding. Some comments and remarks on an extension of this algorithm to select and determine the rank of any $m \times n$ matrix are left to section 5 of the paper.

2 Embedding Equations

A local embedding of a spacetime V_4 in a pseudo-Euclidean M_{4+n} manifold is done when a set of Cartesian coordinates X^μ is specified as functions of the spacetime coordinates x^i (greek indices run from 1 to $4+n$, lowercase latin letters run from 1 to 4 and capital latin letters run from 5 to $4+n$). At any point of the manifold we can find a set of n vector fields N_A orthogonal to V_4 and to themselves. Thus if $\eta_{\mu\nu}$ denotes the Cartesian components of the metric of M_{4+n}, then the following set of equations are valid

$$X^\mu_{,i} X^\nu_{,j} \eta_{\mu\nu} = g_{ij}; \qquad (1.a)$$

$$X^\mu_{,i} N^\nu_A \eta_{\mu\nu} = 0\ ; \qquad (1.b)$$

$$N^\mu_A N^\nu_B \eta_{\mu\nu} = g_{AB}; \qquad (1.c)$$

where g_{ij} is the spacetime metric, $g_{AB} = k^2\epsilon^A\delta_{AB}$, $\epsilon^A = \pm 1$, depending on the signature of M_{4+n}, $X^\mu_{,i} = \frac{\partial X^\mu}{\partial x^i}$ are the components of the tangent vectors to V_4 (partial derivatives with respect to spacetime coordinates are indicated by a comma, as usual) and k is a constant.

The second fundamental form and the torsion vector are given, respectively, by

$$b_{ijA} = N^\mu_A X^\nu_{,ij}\eta_{\mu\nu}, \quad b_{ijA} = b_{jiA}, \quad \text{and} \tag{2.a}$$

$$A_{iAB} = N^\mu_{A,i} N^\nu_B \eta_{\mu\nu}, \quad A_{iAB} = -A_{iBA}. \tag{2.b}$$

In a matricial form the set of equations (1.b) can be written as

$$\mathbf{S}\cdot\mathbf{Y}_A = 0, \qquad A = 5, ..., 4+n; \tag{3}$$

where S is the $4\times(4+n)$ matrix formed by the components of the tangent vectors to V_4 multiplied by the metric components of M_{4+n} and $\mathbf{Y}_A$ is the column matrix $(4+n)\times 1$ formed by the components of the vectors N_A.

The homogeneous system described by equation (3) can be solved (for the non-trivial solution) by taking into account pure algebraic considerations: we need to find a square submatrix of S of order 4×4 that is invertible. That is always possible as the rows of the matrix S are exactly the components of the vectors that generates the tangent space of the spacetime. Thus, they are linearly independent. Therefore from linear algebra we know that there exist a submatrix 4×4 of S that is non-singular.

Let P be a 4×4 submatrix of S that is invertible and Q the matrix formed from S taking out the elements of P. Q is a $4\times n$ matrix. The system (3) can be written in the equivalent form

$$\mathbf{P}\cdot\hat{\mathbf{Y}}_A + \mathbf{Q}\cdot\bar{\hat{\mathbf{Y}}}_A = 0, \tag{4}$$

where $\hat{\mathbf{Y}}_A$ are the components of $\mathbf{Y}_A$ associated to the invertible submatrix and $\bar{\hat{\mathbf{Y}}}_A$ the components of $\mathbf{Y}_A$ associated to the remaining columns. Thus, from (4) we have that

$$\mathbf{P}\cdot\hat{\mathbf{Y}}_A = -\mathbf{Q}\cdot\bar{\hat{\mathbf{Y}}}_A, \tag{5}$$

which allows us to write,

$$\hat{\mathbf{Y}}_A = -\mathbf{P}^{-1}\cdot\mathbf{Q}\cdot\bar{\hat{\mathbf{Y}}}_A. \tag{6}$$

Taking into account the above definitions we write in the following section an algorithm to determine these quantities explicitly.

3 The Algorithm

ALGORITHM A

A1: Given the set of $4+n$ Cartesian coordinates X^μ as a $(4+n)\times 1$ column matrix and the metric tensor $\eta_{\mu\nu}$ as a $(4+n)\times(4+n)$ square matrix, compute the S matrix as $\mathbf{S}_{i\nu} = X^\mu_{,i}\eta_{\mu\nu}$.

A2: Using the algorithm B, decompose S and $\mathbf{Y}_A$ matrices in submatrices P, Q, $\hat{\mathbf{Y}}_A$ and $\bar{\hat{\mathbf{Y}}}_A$ such that $\det(\mathbf{P}) \neq 0$ and such that $\hat{\mathbf{Y}}_A$ contains the components of N_A corresponding to P and $\bar{\hat{\mathbf{Y}}}_A$ those corresponding to Q.

A3: Substitute $\mathbf{P}$, $\mathbf{Q}$, $\bar{\mathbf{Y}}_A$ and $\bar{\bar{\mathbf{Y}}}_A$ in $\bar{\mathbf{Y}}_A - \mathbf{P}^{-1}\cdot\mathbf{Q}\cdot\bar{\bar{\mathbf{Y}}}_A = 0$, obtaining a system of four linear equations in those components of N_A corresponding to $\mathbf{P}$.

A4: Solve that system of equations for the four components of each N_A in $\bar{\mathbf{Y}}_A$ in terms of the n others.

A5: **for** $A \leftarrow 5,\ldots,4+n$ **do for** $B \leftarrow 5,\ldots,A$ **do**

A5a: Substitute the expressions for N_A in $\mathbf{Y}_A\eta\mathbf{Y}_B = g_{AB}$ obtaining a non-linear equation in the components of N_A corresponding to $\mathbf{Q}$.

A5b: Solve this equation for one of the remaining components of N_A in terms of the others.

A5c: Return this solution to the next equation generated in step A5a. At the end of loop $n(n-1)/2$ components of N_A will remaing arbitrary. **endfor**

A6: Compute the second fundamental form and the torsion vector from eqns. (2.a) and (2.b). **stop**

ALGORITHM B

B1: (Inicialization) $p_1 \leftarrow 1$ (p_1 points to a candidate to be the first column of $\mathbf{P}$ (or $\bar{\mathbf{Y}}$)).

B2: **while** $p_1 \leq n+1$ **and** $\det(\mathbf{P}) = 0$[1] **do**

begin $p_2 \leftarrow p_1 + 1$ (p_2 points to a candidate to be the second column of $\mathbf{P}$ (or $\bar{\mathbf{Y}}$).

while $p_2 \leq n+2$ **and** $\det(\mathbf{P}) = 0$ **do**

begin $p_3 \leftarrow p_2 + 1$ (p_3 points to a candidate to be the third column of $\mathbf{P}$ (or $\bar{\mathbf{Y}}$).

while $p_3 \leq n+3$ **and** $\det(\mathbf{P}) = 0$ **do**

begin $p_4 \leftarrow p_3 + 1$ (p_4 points to a candidate to be the fourth column of $\mathbf{P}$ (or $\bar{\mathbf{Y}}$).

while $p_4 \leq n+4$ **and** $\det(\mathbf{P}) = 0$ **do**

begin $j \leftarrow 1$, $k \leftarrow 1$ (j points to a column of $\mathbf{S}$ (or $\mathbf{Y}$), k to a column of $\mathbf{Q}$ (or $\bar{\bar{\mathbf{Y}}}$)).

repeat

begin if $j = p_1$ **then** store the column of $\mathbf{S}$ pointed by j as the first column of $\mathbf{P}$, the one of $\mathbf{Y}$ as the first of $\bar{\mathbf{Y}}$ **else**

if $j = p_2$ **then** store the column of $\mathbf{S}$ pointed by j as the second column of $\mathbf{P}$, the one of $\mathbf{Y}$ as the second of $\bar{\mathbf{Y}}$ **else**

if $j = p_3$ **then** store the column of $\mathbf{S}$ pointed by j as the third column of $\mathbf{P}$, the one of $\mathbf{Y}$ as the third of $\bar{\mathbf{Y}}$ **else**

if $j = p_4$ **then** store the column of $\mathbf{S}$ pointed by j as the fourth column of $\mathbf{P}$, the one of $\mathbf{Y}$ as the fourth of $\bar{\mathbf{Y}}$ **else**

---

[1] Note that det(P) can result an expression that can be zero or not depending of physical informations unavaliable to REDUCE. In this case, if it is not immediatly zero, the actual program could ask the user if it should be taken as zero by use of the internal (symbolic) procedure YESP.

Store the column of $\bar{S}$ pointed by j as the k-th column of $\mathbf{Q}$, the one of $\mathbf{Y}$ as the k-th column of $\bar{\mathbf{Y}}$ and $k \leftarrow k+1$ **endif**.

$j \leftarrow j+1$ **end**

until $j > n+4$ **endrepeat**.

$p_4 \leftarrow p_4 + 1$ **endwhile**

$p_3 \leftarrow p_3 + 1$ **endwhile**

$p_2 \leftarrow p_2 + 1$ **endwhile**

$p_1 \leftarrow p_1 + 1$ **endwhile**.

B3: **return** $\mathbf{P}$, $\mathbf{Q}$, $\bar{\mathbf{Y}}$, and $\bar{\bar{\mathbf{Y}}}$.

The termination of the algorithm at step B3 is guaranteed by the existence of the non-singular submatrix $\mathbf{P}$.

The algorithm above has been implemented in the algebraic computing system REDUCE (Hearn, 1986; Rayna, 1987; Stauffer et. al., 1988) making use of its MATRIX facilities (see Davenport et. al., (1988), for a good introduction to matrix representation in Computer Algebra). However, for shortage of space, we left the program out of the paper.

4 The Schwarzchild Embedding

In the specific case of Schwarzchild spacetime the embedding (Rosen, 1965) is done in a 6-dimensional pseudo-Euclidean manifold ($n = 2$) with metric $\eta_{\mu\nu} = diag(-1,-1,+1,+1,+1,+1)$. The Schwarzchild embedding is given by the coordinates,

$$\begin{aligned}
X^1 &= \sqrt{\beta}\cos t, \\
X^2 &= \sqrt{\beta}\sin t, \\
X^3 &= f(r), \\
X^4 &= r\sin\theta\cos\phi, \\
X^5 &= r\sin\theta\sin\phi, \\
X^6 &= r\cos\theta,
\end{aligned}$$

where $\beta = \beta(r)$ $(\beta(r) = 1 - \frac{2m}{r})$, $f(r)$ is a well defined function of r, and r, θ, ϕ, and t denote the spacetime coordinates. The four vectors tangent to V_4 are determined taking the derivative of the coordinates X^μ with respect to each one of the spacetime coordinates. We denote by $N_A{}^\mu$, $\mu = 1,...,6$ the components of the normal vectors with $A = 5, 6$, respectively. To determine these vectors the following conditions have to be considered: i) the orthogonality of the normal vectors with respect to the tangent vectors (eq. 1.b) (from this we obtain a set of 8 equations) and ii) orthonormality of the normal vectors (eq. 1.c)(from this we get 3 equations).

Out of a total of 11 equations we have now to determine the 6 components of the two vectors N_5 e N_6. We have a set of 11 equations for 12 unknowns. Notice that our unknowns are functions of the spacetime coordinates.

According to the algorithm (and program) developed in the previous section, we just need to set $n = 2$ and ask REDUCE to calculate the matrix S (step A1). After some algebraic manipulation we obtain for the Schwarzchild spacetime embedding the normal vectors,

$$\begin{aligned} N_5^\mu &= h(r)\left(\frac{\cos t}{\sqrt{\beta}}, \frac{\sin t}{\sqrt{\beta}}, -\beta, 0, 0, 0\right) \\ N_6^\mu &= l(r)\left(-\frac{\cos t}{\sqrt{\beta}}, -\frac{\sin t}{\sqrt{\beta}}, \frac{4mf'}{\beta^2}, -\left(\frac{4mf'^2}{\beta^2} - \frac{\beta}{4m}\right)\cos\phi\sin\theta, \right. \\ &\quad \left. -\left(\frac{4mf'^2}{\beta^2} - \frac{\beta}{4m}\right)\sin\phi\sin\theta, -\left(\frac{4mf'^2}{\beta^2} - \frac{\beta}{4m}\right)\cos\theta\right) \end{aligned}$$

where $h(r) = \frac{4mf'\sqrt{\beta}}{\sqrt{\beta^3 - 16m^2f'^2}}$, $l(r) = \frac{4m\beta^2}{\sqrt{16m^2f'^2 - \beta^2}\sqrt{16m^2(1+f'^2) - \beta^3}}$ and $f' = \frac{df}{dr}$.

It is easy now to calculate with REDUCE (but tedious by hand) the *second fundamental form* and the *torsion* vector, from the eqs.(2.a and 2.b - step A6):

$b_{115} = -h(r)$, $b_{116} = -l(r)$, $b_{225} = -\frac{\beta}{4m}\left(\frac{f''}{f'} + \frac{3\beta}{4m}\right)h(r)$,
$b_{226} = \left(\frac{4m^2}{\beta^3}f'f'' + \frac{3\beta^2}{16m}\right)l(r)$, $b_{336} = \frac{r}{4m\sqrt{\beta f'}}h(r)$,
$b_{446} = \frac{r}{4m\sqrt{\beta f'}}\sin^2\theta\, h(r)$,
$A_{256} = \frac{4mf'' + 3\beta f'}{\sqrt{\beta(\beta^3 - 16m^2f'^2)}}l(r)$.

5 Final Remarks

The geometrical and physical analysis of these quantities are not the main concern of this paper. However it is important to point out that geometrically the second fundamental form and the torsion vector are fundamental quantities as they determine, together with the metric, the structure of the embedding manifold and physically if General Relativity has to be considered as part of a more general theory of embedded manifolds then, besides the metric which represents the classical gravitation, these two quantities have also to be considered: the second fundamental form may be interpreted as the source of the matter fields and the torsion vector may represent a Yang-Mills gauge field (see Maia, 1986, for details).

The calculations were initially done in interactive form with the version 3.2 of REDUCE running in an IBM PC-XT and later on (by demand) in a microVAX running VMS. Finally we coded a fairly general program[2] for the 3.3 version of REDUCE requiring only as input the number of extra dimensions n, the set of Cartesian coordinates X^μ, and the metric $\eta_{\mu\nu}$ of the embedding manifold M_{4+n}.

The available physical memory of the PC (640 Kb, but less when the system is loaded) is a great limiting factor for the execution of calculus with more general functions and/or higher dimensions (this would involve matrices with order greater then 4×6). To circumvent the very often problem of free storage cell explosion in the PC, we had to make the trick of using the output of the results as input for the following steps.

[2] Complete program and output listings may be obtained from the authors.

Though this initial limitation at the PC, the problem above would have been far more difficult to solve with paper and pencil than with the interactive use of REDUCE.

The algorithm developed here can be extended to determine all non-singular submatrices of a given matrix determining, in addition, its rank[8]. Thus it can also be used to establish the existence and type of solution of a system of linear equations by the simple analysis of its coefficients' matrix and extended matrix ranks, for either symbolic (functions) or numeric matrix entries, as the manipulation is purely algebraic.

Acknowledgments

W.L.Roque is grateful to the CNPq for financial support through a research grant.

[8] If $\mathbf{A}$ is a $m \times n$ matrix, the number of submatrices of order $k \times k$ of $\mathbf{A}$ is given by $N = \binom{m}{k}\binom{n}{k}$, where $k \leq min(m,n)$ and $\binom{a}{b} = \frac{a!}{(a-b)!b!}$.

References

Davenport, J. H., Siret, Y., Tournier, E. (1988). *Computer Algebra: Systems and Algorithms for Algebraic Computation.* Academic Press.

Eisenhart, L. P. (1946). *Riemannian Geometry.* Princeton University Press.

Hawking, S. W. & Ellis, G. F. R. (1973). *The Large Scale Structure of Space-Time.* Cambridge University Press.

Hearn, A. C. (1986). *REDUCE Manual.* The Rand Corporation, Santa Monica.

Maia, M. D. (1986). The Physics of the Gauss-Codazzi-Ricci Equations. *Mat. Aplic. Comp.* **5**, *283-292*; On Kaluza-Klein Relativity. *Gen. Rel. Grav.* **18**, *695-699.*

Maia, M. D. & Roque, W. L. (1988). Classical Membrane Cosmology. *Phys. Lett. A* **139**, *121-124.*

Rayna, G. (1987). *REDUCE: A Software for Algebraic Computation*, Springer-Verlag.

Rosen, J. (1965). Embedding of Various Relativistic Riemannian Spaces in Pseudo-Euclidean Spaces. *Rev. Mod. Phys.* **37**, *204-214.*

Stauffer, D., Hehl, F.W., Winkelmann, V. and Zabolitzky, J.G. (1988). *Computer Simulation and Computer Algebra*, Springer-Verlag.