Symbolic software for Y-energy extremal Finsler submanifolds

Vladimir Balan and Ariana Pitea

Abstract. The paper describes a novel software library which provides the basic relativistic geometric objects for (pseudo-) Finsler manifolds. This extends the MAPLE symbolic computation framework initiated by Rutz de Solange ([4]). In particular, the Y-mean curvature ([42]) is studied for pseudo-Finsler locally Minkowski manifolds, and the PDEs of the Y-extremal minimal hypersurfaces are derived for the conformally deformed Berwald-Moor relativistic model.

M.S.C. 2000: 49Q05, 53B40, 53C60, 53A10, 53C42.
Key words: Finsler space, Y-minimal surfaces, mean curvature, m-root metric.

1 Introduction. Y-extremal hypersurfaces

Let (F, M) be a real n-dimensional Finsler space, endowed with a Finsler connection $\nabla = (N^i_j, F^i_{jk}, C^i_{jk})$, where the metricity, the $vv-v$ symmetry and $h-$vanishing deflection are assumed (i.e., cf. [43], it is a generalized Cartan connection). Then its coefficients are given by:

$$\left\{ \begin{array}{l} N^i_j = \gamma^i_{0j} - C^i_{js}(\gamma^s_{00} + \tau^s_{00}) + g^{is}\tau_{0js} \\ F^i_{jk} = \frac{1}{2}g^{is}(\delta_j g_{sk} + \delta_k g_{sj} - \delta_s g_{jk}) - \frac{1}{2}g^{is}(-g_{kt}T^t_{si} - g_{it}T^t_{sk} + g_{st}T^s_{jk}) \\ C^i_{jk} = \frac{1}{2}g^{is}\dot{\partial}_k g_{sj}, \end{array} \right. \tag{1.1}$$

where

$$\delta_j = \partial_j - N^s_j\dot{\partial}_s, \quad \partial_j = \partial/\partial x^j, \ \ \dot{\partial}_s = \partial/\partial y^s,$$
$$g_{ij} = 1/2\partial^2 F^2/\partial y^i \partial y^j, \quad g^{is}g_{sj} = \delta^i_j.$$

The $hh-h$ torsion $\{T^i_{jk} = F^i_{[jk]}\}$ is assumed to be known. We denote by 0 index the transvection with y^k, $\tau_{[ij]} = \tau_{ij} - \tau_{ji}$ and the Einstein summation rule is assumed.

Remarks. a) For $T^i_{jk} = 0$, (N, F, C) becomes *the Cartan connection.*

b) For $T^i_{jk} = \delta^i_{[j}\tau_{k]}$ the connection is *semi-symmetric* ([43, 42]).

c) For $T^i_{jk} = y_{[j}C^i_{|k]}$, with $y_i = g_{i0}$ and $C^j = g^{rs}C^i_{rs}$, we get *the Barthel connection.*

Differential Geometry - Dynamical Systems, Vol.11, 2009, pp. 41-53.

We may consider as well the geometric object

$$\Gamma^i_{jk} = F^i_{jk} + C^i_{js}N^s_{k.} \tag{1.2}$$

Let be a vector field $X \in \chi(M)$. This produces on M a connection $\bar\nabla = \{\bar\Gamma^i_{jk}\}$, called *the linear Y-connection associated to* ∇, with coefficients

$$\bar\Gamma^i_{jk} = \tilde\Gamma^i_{jk} + C^i_{jr}\partial_k Y^r = \bar F^i_{jr} + \tilde C^i_{jr} Y^r_k, \tag{1.3}$$

where "dash" means the replacement of y by $Y(x)$, and

$$Y^j_j = \partial_j Y^i + \bar N^i_j. \tag{1.4}$$

We remark that $N^i_j = F^i_{0j}$ ([42]) and

$$Y^i_k = \partial_k Y^i + \tilde N^i_k = \partial_k Y^i + \tilde F^i_{sk} Y^s = Y^i_{|k}. \tag{1.5}$$

Proposition. ([42]) *a) The following property holds true:*

$$\bar T^i_{jk} \equiv \bar\Gamma^i_{[jk]} = \tilde T^i_{jk} + \tilde C^i_{[js}Y^s_{k]} \tag{1.6}$$

b) $\bar\nabla$ is metrical with respect to $\bar g_{ij} = \tilde g_{ij}$ (the Riemannian Y-metric); c) a transversal unit vector field Y to a hypersurface $\Sigma:\ \ x = x(u^1, ..., u^{n-1}, c)$ is provided by $\{Y^i\}$ such that

$$\tilde g_{ij} Y^i B^j_\alpha = 0, \alpha = \overline{1, n-1}, \quad \tilde g_{ij} Y^i Y^j = 1, \tag{1.7}$$

where $B^i_\alpha = \partial x^i/\partial u^\alpha$. As well, such a field in the auto-parallelism case ($Y^i_s Y^s = 0$) by

$$\tilde g_{ij} Y^i Y^j = 1,\ \ \bar g_{[is} Y^s_{j]} + Y_{[i} T_{j]rs} Y^r Y^s + Y_s \tilde T^s_{ij} = 0, \tag{1.8}$$

For the Cartan connection case one gets $(1.8)_2$ rewritten as $\bar g_{is} Y^s_j = \bar g_{js} Y^s_i$. The construction above yields a frame $\{B^i_\alpha; Y^i\}$ along Σ. Then it is known the following result ([42]):

Proposition. *a) $g_{ab} = \tilde g_{ij} B^i_\alpha B^j_\beta$ defines an induced by $\tilde g$ metric on Σ.*

b) The connection induced by $\tilde\Gamma$ on Σ has the coefficients

$$\Gamma^\alpha_{\beta\gamma} = B^\alpha_i (B^i_{\beta\gamma} + B^j_\beta B^k_\gamma \bar\Gamma^i_{jk}),$$

where we have denoted $B^i_{\beta\gamma} = \partial^2 x^i/\partial u^\beta \partial u^\gamma$ and $\{B^\alpha_i, Y_i\}$ is the dual coframe *with respect to $\{B^i_\alpha, Y^i\}$.*

c) One has the relations ([43, 42])

$$\begin{cases} B^i_\alpha B^\alpha_j + Y^i Y_j = \delta_{ij}, \quad B^i_\alpha B^\beta_j = 0, \\ B^i_\alpha Y_i = 0, B^\alpha_i Y^i = 0, \quad Y^i Y_j = 1, \\ g_{\alpha\beta} B^\beta_i = g_{ij} B^j_\alpha, Y_i = g_{ij} Y^j. \end{cases}$$

As well, we have the following

Corollary. *a) The Y-induced connection coefficients are*

$$\Gamma^{\alpha}_{\beta\gamma} = B^{\alpha}_{i}[B^{i}_{\beta\gamma} + B^{j}_{\beta}B^{k}_{\gamma}(\tilde{F}^{i}_{jk} + \tilde{C}^{i}_{js}Y^{s}_{k})].$$

b) Denoting

$$B^{i}_{\alpha;\beta} = B^{i}_{\alpha\beta} + \bar{\Gamma}^{i}_{jk}B^{j}_{\alpha}B^{k}_{\beta} - B^{i}_{\gamma}\Gamma^{\gamma}_{\alpha\beta},$$

we get the second fundamental form *of coefficients*

$$H_{\alpha\beta} = \tilde{F}^{-1}B^{i}_{\alpha;\beta}g^{\alpha\beta},$$

where $g^{\alpha\beta}g_{\beta\gamma} = \delta^{\alpha}_{\gamma}$, *and* the shape operator

$$H^{\alpha}_{\beta} = g^{\alpha\gamma}H_{\gamma\beta}.$$

Remarks. a) In general, $H_{[\alpha\beta]} = Y_i\bar{T}^{i}_{jk}B^{j}_{\alpha}B^{k}_{\beta} \neq 0$; still for the Cartan connection, this vanishes, i.e., $H_{\alpha\beta}$ is symmetric.

b) *the Y-mean curvature* is given by

$$H = g^{\alpha\beta}H_{\alpha\beta} = Tr(H^{\alpha}_{\beta}).$$

Theorem. *A hypersurface $\Sigma : x = x(u)$ is Y-extremal (the Y-energy integral of the immersion is minimized) iff one of the following conditions occurs ([42, (3.2)]):*

a) $[\frac{\partial\tau}{\partial x^i} - \frac{\partial}{\partial u^\alpha}(\frac{\partial\tau}{\partial B^i_\alpha})]Y^i = 0$;

b) $H = (\tilde{T}_j{}^i{}_k + \tilde{C}_jY^j_k)Y^k$, *where* $\sigma = \sqrt{\det(g_{\alpha\beta})}$.

Remark. In the Cartan $Y-$connection case, Y-extremality is equivalent to Wegener's equation

$$H = \tilde{C}_jY^j_kY^k,$$

and the right hand side is null iff Y is autoparallel (in the geodesic case). In this case, $T^i_{jk} = C^i_{[js}Y^s_{k]} \neq 0$.

Theorem. The Cartan Y-connection*, which is $h-$ and $v-$metrical, with null $h-$deflection, $vv-v$ symmetrical and with $T^i_{jk} = F(Y^s_{[j}C^i_{sk]})$, induces a metric linear connection (with respect to $\bar{g}_{ij}$, the Riemannian Y-metric) and has the coefficients*

$$\begin{cases} C^i_{jk} = \frac{1}{2}\frac{\partial g_{jk}}{\partial x^s}g^{si} \\ F^m_{ik} = g^{mj}[\gamma_{ijk} - C_{ijs}N^s_k + C_{[jks}(FY^s_{i]} - N^s_{i]})], \\ N^i_k = \gamma^j_{0k} + \tilde{C}^j_{kr}(\partial_iY^r)Y^i \end{cases}$$

b) For the induced connection, one has $H_{[\alpha\beta]} = 0$, and the submanifold is (pseudo-)Riemannian with respect to $H_{\alpha\beta}$ too (!) and the Y-extremal surfaces satisfy $H = 0$.

2 N-extremal surfaces

We describe further the algorithm for finding the $Y-$normal $N \perp \Sigma$ orthogonal to the hypersurface Σ, according to [42].

Task. Given $\{g_{ij}(x,y)\}$ and $\{B^i_\alpha(u)\}$, find $\{N^i\}$, such that

$$g_{ij}N^iN^j = 1, \quad g_{ij}N^iB^j_\alpha = 0, \alpha = \overline{1, n-1}.$$

We note that the first relation rewrites $F(x,N) = 1$.

1. Build $D = [B^i_1, ..., B^i_{n-1}, d^i]$ nonsingular, by adding an extra column.
2. Compute the algebraic complements q_i of d^i. These satisfy the relations
$$q_sB^s_\alpha = 0, \quad q_sd^s = \det D.$$
3. Solve the system $g_{is}(x,p)p^s = q_i, i = \overline{1,n}$. This is *non-linear*, but compatible, since its Jacobian with respect to p is ([42])
$$\det[\partial(g_{is}(x,p)p^s - q_i)/\partial p^k] = \det[2C_{isk}(x,p)p^s + g_{ik}(x,p)] = \det(g_{ik}(x,p)) \neq 0.$$
4. Normalize the solution: $N^i = \frac{p^i}{F(x,p)}$.

Remarks. a) $\{N^i\}$ is *contravariant* ([42, (5.12)]), and $\frac{\partial N^i(x,B)}{\partial B^j_\alpha} = -B^{\alpha i}N_j$, where

$$B^{\alpha i} = g^{ij}(x,N)B^\alpha_j = g^{\alpha\beta}B^i_\beta. \tag{2.1}$$

Proposition. *a) The induced contact connection on Σ of the generalized Cartan connection is given by*

$$F^i_{j\gamma}(u) = F^i_{jk}(x,N)B^k_\gamma + C^i_{j\gamma}(x,N)N^k_\gamma,$$

where $N^k_\gamma = \frac{\partial N^k}{\partial u^\gamma} + N^k_h(x,N)B^h_\gamma$ and where, according to (1.6),

$$N^k = N^k(x,B), \quad \frac{\partial N^k}{\partial u^\gamma} = \partial_i N^k B^i_\gamma + \frac{\partial N^k}{\partial B^i_\alpha}B^i_{\alpha\gamma}.$$

b) The second fundamental form $H_{\alpha\beta}$ is given by

$$H_{\beta\gamma} = \frac{N_iB^i_{\beta;\gamma}}{F(x,N)},$$

where $B^i_{\beta;\gamma} = B^i_{\beta\gamma} + B^j_\beta F^i_{j\gamma} - B^i_\alpha\Gamma^\alpha_{\beta\gamma}$ with the tangent connection coefficients

$$\Gamma^\alpha_{\beta\gamma} = B^\alpha_i(B^i_{\beta\gamma} + B^j_\beta F^i_{j\gamma}).$$

c) In general, the second fundamental form is non-symmetric, since $H_{[\beta\gamma]} = N_i(B^j_{[\beta}F^i_{j\gamma]})$.

Theorem. The minimal hypersurface equation with respect to the generalized Cartan connection is given by

$$(T^i_{i0} + T^i_{ij}C^j + C^i_{|i})|_{y=N} = H_{\alpha\beta}B^\alpha_i B^\beta_j (g^{ij} + C^i C^j + g^{ik}\dot{\partial}_k C^j)|_{y=N}, \tag{2.2}$$

where $C^i_{|k} = \partial_k C^i + C^s F^i_{sk}, C^i = C^i_{jk}g^{jk}$.

Corollary. *a) In the case of Cartan connection, a hyperplane is minimal iff* $(C^i_{|i})_{y=N} = 0$.

b) In the case of Barthel connection, a hyperplane is minimal iff $(C^i_{|i})_{y=N} = 0$.

Remarks. a) Denoting $l_k = \frac{y_k}{F}$, we have

$$T^i_{jk} = \delta^i_{[j}\omega_{k]} \Rightarrow T^i_{jk} = -\delta^i_{[j}l_{k]}, C = FC^i_{|i}/(n-1);$$

b) If a connection is $h-$ and $v-$metrical, with $h-$deflection zero, $vv - v$ symmetrical and Γ^i_{jk} is given by the consequences in a), i.e., if it is a "Cartan C-connection" ([42, p. 663]), then it is determined by F iff

$$C^* \equiv (n-1) + F^2 C^i|_i \neq 0, \quad C^i|j = \dot{\partial}_j C^i + C^s C^i_{sj}.$$

For such a connection, the left side in (2.2) vanishes and $H_{[\alpha\beta]} = 0$.

3 Conformally deformed Berwald-Moor case. Dedicated software

The algorithm of determining the mean curvature for $Y-$surfaces which is described in the previous sections is further implemented for a conformally deformed Berwald-Moor Finsler structures (M, F), $(\dim M = n \in \overline{3,4})$. The Maple 12 symbolic package develops the PDEs of minimal surfaces and derives the mean curvature for the case of dimension $n = 4$. The basic locally Minkowski Finsler fundamental function is of the form ([45, 46])

$$F(y) = \sqrt[m]{y^1 \dots y^m},$$

and the conformally deformed Finsler metric is $\tilde{F}(x, y) = e^{2\sigma(x)}F(y)$. The software code containing the Maple dedicated procedures is included in the Appendix.

4 Conclusions

Based on the theory of $Y-$minimal hypersurfaces in (pseudo-)Finsler spaces initiated by M. Matsumoto ([42]) and developed by A. Bejancu ([18]), we extend the symbolic Maple software developed by Rutz de Solange ([4], 2000). The Appendix contains the code applied to the quartic conformally deformed Berwald-Moor Finsler models.

Acknowledgement. The present work was developed as part of the Romanian Academy Grant No. 5 / 5.02.2008.

References

[1] J.C.Álvarez Paiva, G.Breck, *What is wrong with the Hausdorff measure in Finsler spaces*, arXiv.math.DG/0408413, 2004; Adv. Math. 204 (2006), 647-663.

[2] J.C.Álvarez Paiva, A.C.Thompson, *Volumes on normed and Finsler spaces*, In: Riemann-Finsler Geometry, MSRI Publications, 49 (2004), 1-46.

[3] H. Akbar-Zadeh, Sur les sous-varietes des varietes Finsleriennes, C.R. Acad. Sci. Paris Ser. A-B 266 (1968), A146-A148.

[4] P.L. Antonelli, *Equivalence problem for systems of second order ordinary differential equations*, Encyclopedia of Math., Kluwer Acad. Publ, 2000.

[5] G.S.Asanov, *Finsler Geometry, Relativity and Gauge Geometry*, D.Reidel, Dordrecht, 1985.

[6] V.Balan, *CMC and minimal surfaces in Berwald-Moor spaces*, Hypercomplex Numbers in Geometry and Physics, 2(6), 3 (2006), 113-122.

[7] V. Balan, *CMC and minimal surfaces in Finsler spaces with convex indicatrix*, Conference on Integral Geometry, Geometric Inequalities, Finsler Geometry and Convexity, Related Topics, 22-29 Dec. 2006, Tianjin, China, http://www.nim.nankai.edu/activities/conferences/hy061226/lectures/Presentation_Balan.pdf

[8] V. Balan, *BH-mean curvature in Randers spaces with anisotropically scaled metric*, Proceedings of The International Conference "Differential Geometry and Dynamical Systems" (DGDS-2007), October 5-7, 2007, Bucharest, Romania, BSG Proceedings 15, Geometry Balkan Press 2008, 34-39.

[9] V.Balan, *DPW method for CMC surfaces in Randers spaces*, in: "Modern Trends in Geometry and Topology" (Eds: D. Andrica, P.A. Blaga, S. Moroianu), Proc. of The Seventh International Workshop on Differential Geometry and Its Applications, 5-11 September 2005, Deva, Romania, Cluj University Press 2006, 79-86.

[10] V.Balan, *Moving frames of Riemannian surfaces in GL^3 spaces*, Algebras, Groups and Geometries, Hadronic Press, 17 (2000), 3, 273-282.

[11] V.Balan, *On the DPW Method for the Tangent Group*, Proceedings of the Conference of Differential Geometry, Lie Algebras and Global Analysis, Thessaloniki 1998, Geometry Balkan Press, Bucharest 2000, 1-9.

[12] D.Bao, S.-S. Chern, Z. Shen, *An Introduction to Riemann-Finsler Geometry*, Springer-Verlag, 2000.

[13] W. Barthel, *A Uber die Minimal Flaachen in gefaserten Finsler-Raumen* , Ann. Mat. Pura Appl. 36 (1954), 159-190.

[14] W. Barthel, *Variationsprobleme der Oberfachenfunktion in der Finslerschen Geometrie*, Math. Z. 62 (1955), 23-36.

[15] W. Barthel, *Extremalprobleme in der Finslerschen In haltsgeometrie* , Ann. Univ. Sarav 4 (1956), 171-183.

[16] A.Bejancu, *Finsler Geometry and Applications*, E.Horwood, 1990.

[17] A.Bejancu, *Special immersions of Finsler spaces*, Stud.Cerc.Mat. 39 (1987), 463-487.

[18] A. Bejancu, *Geometry of Finsler Submanifolds*, Kluwer Acad. Publishers 2000.

[19] A. Bejancu, *Geometry of Finsler subspaces. I*, An. Stiint. Univ. "Al.I. Cuza" Iasi Sect. I a Mat. 32 (1986), 69-83.

[20] G.Belletini, M.Paolini, *Anisotropic motion by mean curvature in the context of Finsler geometry*, Hokkaido Math. Jour. 25 (1995), 537-566.

[21] G.Belletini, M.Paolini, *Some aspects of mean curvature flow in presence of nonsmooth anisotropies*, In: 3-rd European Congress of Mathematics (ECM), Barcelona, Spain, July 10-14, 2000, (Ed: Casacuberta, Charles & al), Volume II, Basel, Birkhäuser, Progress in Mathematics 202 (2001), 245-253.

[22] G.Belletini, M.Paolini, S.Venturini, *Some results on surface measures in claculus of variations*, Ann. di Mat. Pura ed Appl., IV vol. CLXX (1996), 329-259.

[23] G.Breck, *Minimalité des sous-variétés totalement géodésiques en géométrie Finslerienne*, arXiv:math.DG/0409320, v1, 18 Sept. 2004.

[24] V. M. Chernov, *On defining equations for the elements of associative and commutative algebras and on associated metric forms*, Space-Time Structure (Ed: D.Pavlov), Tetru Eds., 2006, 227-249.

[25] I. Comic, *The induced curvature tensors of a subspace in a Finsler space*, Tensor, N. S. 23 (1971), 21-34.

[26] I. Comic, *The intrinsic curvature tensors of a subspace in a Finsler space*, Tensor, N. S. 24 (1972), 19-28.

[27] I. Comic, *Induced and intrinsic curvature tensors of a subspace in the Finsler space*, Publications de L'Institut Mathematiane 23 (1978), No. 37, 67-74.

[28] E.T. Davies, *Subspaces of a Finsler space*, Proc. London Math. Soc. 49 (1947), 19-39.

[29] S. Deng and Z. Hou, *A simple proof of a Theorem of H.C. Wang*, School of Mathematical Sciences and LPMC, Nankai University, Tianjin, China, 2004, scientific report.

[30] S.Deng and Z. Hou, Some applications of Lie theory to Finsler geometry, preprint.

[31] S.Deng and Z. Hou, The group of isometries of a Finsler space, Pacific Journal of Mathematics 207, 1 (2002), 149-155.

[32] J.Dorfmeister, V.Balan, *On the DPW method for general Lie groups*, The 932-nd A.M.S. Conference, 27-28 March 1998, Kansas State University, Manhattan, Kansas, USA, lecture.

[33] Josef Dorfmeister, Franz Pedit, Hongyong Wu, *Weierstrass type representations of harmonic maps into symmetric spaces*, Comm. Analysis and Geometry, 6(1998), 633-668.

[34] S. Dragomir, Submanifolds of Finsler spaces, Confer. Semi. Mat. Univ. Bari 217 (1986), 15 pp.

[35] Shoichi Fujimori, Shimpei Kobayashi, Wayne Rossman, *Loop Group Methods for Constant Mean Curvature Surfaces*, arXiv:math/0602570v1 [math.DG] 25 Feb 2006.

[36] Q. He, Y.-B. Shen, *On Bernstein type theorems in Finsler spaces with the volume form induced from the projective sphere bundle*, Proc. of the AMS, 134 (3) (2006), 871-880.

[37] M.K. Gupta and P.N. Pandey, *Relations between the main scalars of a four-dimensional Finsler space and its hypersurface*, Differential Geometry - Dynamical Systems (DGDS), 10 (2008), 132-138.

[38] H. Hombu, *Die KrAummungstheorie im Finslerschen Raum*, J. Fac. Sci. Hokkaido Univ., Ser. I. Math. 5 (1936), 67-94.

[39] J.Inoguchi, M.Toda, *Timelike minimal surfaces via loop groups*, Acta Appl. Math. 83 (2004), 313-355.

[40] S. Kobayashi, K. Nomizu, *Foundations of Differential Geometry* I, II, Interscience Publishers, New York, (1963), (1969).

[41] M. Matsumoto, *The induced and intrinsic Finsler connections of a hypersurface and Finslerian projective geometry*, J. Math. Kyoto Univ. 25 (1985), 107-144.

[42] M. Matsumoto, *Theory of Y-extremal and minimal hypersurfaces in a Finsler space*, J. Math. Kyoto Univ. 26 (1986), 647-665.

[43] R.Miron, M.Anastasiei, *The Geometry of Vector Bundles. Theory and Applications*, Kluwer Acad. Publishers, FTPH, no.59, 1994.

[44] Gh. Munteanu, *Minimal hypersurfaces of a complex Finsler space*, Proc. of Conf. "Riemannian Geometry and Applications" Brasov, June 21-25, 2007, preprint.

[45] D.G. Pavlov, *Four-dimensional time*, Hypercomplex Numbers in Geometry and Physics, Ed. "Mozet", Russia, 1, 1 (2004), 31-39.

[46] D.G. Pavlov, *Generalization of scalar product axioms*, Hypercomplex Numbers in Geometry and Physics, Ed. "Mozet", Russia, 1, 1 (2004), 5-18.

[47] H.Qun, Y.Shen, *On the mean curvature of Finsler submanifolds*, Chinese Jour. of Contemp. Math., 27 (2006), No. 4, 1-12.

[48] H.Qun, Y.Shen, *On Bernstein type theorems in Finsler spaces with the volume form induced from the projective sphere bundle*, Proc. of the A.M.S. 134 (2005), No. 3, 871-880.

[49] H. Rund, *The Differential Geometry of Finsler Spaces*, Springer-Verlag 1959.

[50] H. Rund, *A divergence theorem for Finsler metrics*, Monatsh. Math. 79 (1975), 233-252.

[51] H. Rund, *A Uber die parallelverschiebung in Finslerschen Raumen*, Math. Z. 54 (1951), 115-128.

[52] H. Rund, *The theory of subspaces of a Finsler space. I*, Math. Z. 56 (1952), 363-375

[53] H. Rund, *The theory of subspaces of a Finsler space. II*, Math. Z. 57 (1953), 193-210

[54] T. Sakaguchi, *Subspaces in Finsler space*, Memoirs of the National Defense Academy, Japan 28 (1988), 1-38.

[55] Z.Shen, *Differental Geometry of Sprays and Finsler Spaces*, Kluwer Acad. Publishers, 2001.

[56] Z.Shen, *On Finsler geometry of submanifolds*, Mathematische Annalen, Springer-Verlag 311 (1998), 549-576.

[57] M. Souza, *Superficies Minimas en Espacos de Finsler com Uma Metrica de Randers*, Ph.D. Thesis, University of Brasil, 2001.

[58] M. Souza, K. Tenenblat, *Minimal surface of rotation in Finsler space with a Randers metric*, Math. Ann. 325 (2003), 625-642.

[59] M.Toda, *Weierstrass-type representation of weakly regular pseudospherical surfaces in Euclidean space*, BJGA 7,2 (2002), 87-136.

[60] J. M. Wegener *Hyperflaachen in Finslerschen Raumen als Transversalfachen einer Schar von Extremalen*, Mh. Math. Phys. 44 (1936), 115-130.

[61] J. M. Wegener *Untersuchungen Auber Finslerschen Raume*, Lotos 84 (1936), 4-7.

[62] T. Weinstein, *An introduction to Lorentz surfaces*, de Gruyter Expsitions in Math. 22, Walter de Gruyter, Berlin, 1996.

Authors' address:

Vladimir Balan and Ariana Pitea
University Politehnica of Bucharest, Faculty of Applied Sciences,
Dept. Mathematics-Informatics I, Splaiul Indpendentei 313,
RO-060042 Bucharest, Romania.
E-mail: vbalan@mathem.pub.ro, apitea@mathem.pub.ro

Appendix.

```
    Symbolic software for the mean curvature of Y-hypersurfaces
                          = case: 4D =

Input: Y,f
Output:  H

> restart; with(tensor): with(PDEtools, casesplit, declare):
> tens:=proc(name,how) ‘name‘||‘_‘:=how: name:=get_compts(‘name‘||‘_‘):
      end proc:

x_(n) = position vector;  x(n) =effective components
y_(n) = direction vector;  y(n) =effective components
n = the dimension of the manifold

> n:=4:
> tens(x,create([1],eval(array(sparse,1..n,[x1,x2,x3,x4])))):
> tens(Y,create([1],eval(array(sparse,1..n,[Y1(x[1],x[2],x[3],x[4]),
    Y2(x[1],x[2],x[3],x[4]),Y3(x[1],x[2],x[3],x[4]),
    Y4(x[1],x[2],x[3],x[4])])))):
> tens(y,create([1],eval(array(sparse,1..n,[y1,y2,y3,y4])))):
> get_compts(x_):

I. Function f is given; matrix g_; g =effective components (nxn)

> e:=exp(1): f:=1/2*(e^(x[1]+x[2]+x[3]+x[4])*root[4](y[1]*y[2]*y[3]*y[4]))^2:
> sx:=seq(x[i],i=1..n):
> sxy:=(seq(x[i],i=1..n),seq(y[i],i=1..n)):
> sxY:=(seq(x[i],i=1..n),seq(Y[i],i=1..n)):
> ff:=unapply(f,sxy): man:=array(symmetric,sparse,1..n,1..n):
   for i from 1 to n do for j from 1 to n do
        man[i,j]:=simplify(D[i+n,j+n](ff)):
   end do;end do:
> g__ij_:=create([-1,-1],eval(apply(copy(man),sxy))):
> g__ij:=get_compts(g__ij_):
> gY__ij_:=create([-1,-1],eval(apply(copy(man),sxY))):
> gY__ij:=get_compts(gY__ij_):
> gY_ij_:= invert(gY__ij_, 'detgY__ij_'): gY_ij:=get_compts(gY_ij_):

II.The inverse of the matrix g is computed (matrix g_inv)

> g_ij_:= invert(g__ij_, 'detg__ij_'):

III.The partial derivatives of the metric g, D1g, are computed

> coord:=[sx]: D1g__ij_:= d1metric(g__ij_,coord):

IV. Christoffel's symbols of the first and second kind are computed,
    gamma_ijk=effective components of the Chtistoffel's simbols of second kind

> Cf1:= Christoffel1 (D1g__ij_):
> Cf2:= Christoffel2 (g_ij_, Cf1): gamma_ijk:=get_compts(Cf2):

V. The  product  yP of  the  column  y_  and the line  y_ is computed

>     man:=array(sparse,1..n,1..n):
      for i from 1 to n do for j from 1 to n do man[i,j]:=y[i]*y[j];
      end do;end do: y__ij:=create([1,1],eval(man)):

ggam is gamma^i_jk yP^{jk}

> gam:=get_compts(prod(Cf2,y__ij,[2,1],[3,2])): ggam:=array(1..n):
   for i from 1 to n do ggam[i]:=unapply(gam[i],sxy): end do:
```

```
VI.The components N_ of the  nonlinear connection  are computed ,
   N=effective components
>    man:=array(sparse,1..n,1..n):
     for i from 1 to n do for j from 1 to n do
       man[i,j]:=simplify(D[j](gamma[i]));end do;end do:
> N_i_j_:=create([1,-1],eval(apply(copy(man),sxy))): N_i_j:=get_compts(N_i_j_):
> NY_i_j_:=create([1,-1],eval(apply(copy(man),sxY))):
> NY_i_j:=get_compts(NY_i_j_):
VII.The components of Chern connection , gamma_sjk_, gamma_i_jk_, are computed,
   gamma_sjk, gamma_i_jk=effective components of gamma_sjk_, gamma_i_jk_
>    man:=array(sparse,1..n,1..n,1..n):
       for i from 1 to n do for j  from 1 to n do for k from 1 to n do
     man[i,j,k]:=simplify(D[k+n](unapply(g__ij[i,j],sxy))):end do;end do;end do;
> der:=create([-1,-1,-1],eval(apply(copy(man),sxy))): deriv:=get_compts(der):
> derY:=create([-1,-1,-1],eval(apply(copy(man),sxY))):
> derivY:=get_compts(derY):
> produs:=prod(N_i_j_,der,[1,3]): produs1:=get_compts(produs):
> produsY:=prod(NY_i_j_,der,[1,3]): produs1Y:=get_compts(produsY):
>    man:=array(sparse,1..n,1..n,1..n):
     for i from 1 to n do for j from 1 to n do for k from 1 to n do
       man[i,k,j]:=simplify(D[j](unapply(g__ij[i,k],sxy))-
                   unapply(produs1[i,j,k],sxy)):
     end do;end do;end do:
> delta:=create([-1,-1,-1],eval(apply(copy(man),sxy))):
> Delta:=get_compts(delta):
> man:=array(sparse,1..n,1..n,1..n):
   for i from 1 to n do for j from 1 to n do for k from 1 to n do
       man[i,j,k]:=simplify(unapply(Delta[i,k,j],sxy)+unapply(Delta[i,j,k],sxy)-
                   unapply(Delta[j,i,k],sxy)/2):
   end do;end do;end do:
> gamma__sjk_:=create([-1,-1,-1],eval(apply(copy(man),sxy))):
> gamma__sjk:=get_compts(gamma__sjk_):
> gammaY__sjk_:=create([-1,-1,-1],eval(apply(copy(man),sxY))):
> gammaY__sjk:=get_compts(gammaY__sjk_):
> gamma_i_jk_:=simplify(prod(g_ij_,gamma__sjk_,[2,1])):
> gamma_i_jk:=get_compts(gamma_i_jk_):
> gammaY_i_jk_:=simplify(prod(g_ij_,gammaY__sjk_,[2,1])):
> gammaY_i_jk:=get_compts(gammaY_i_jk_):
VIII.The components of Cartan's connection are computed, Cartan__ijk_,
   Cartan_i_jk_, Cartan__ijk. Cartan_i_jk are the effective components.
>  man:=array(sparse,1..n,1..n,1..n):
   for i from 1 to n do for j from 1 to n do for k from 1 to n do
      man[i,j,k]:=simplify(1/2*D[i,j,k](ff)):
   end do;end do;end do:
> cartan__ijk_:=create([-1,-1,-1],eval(apply(copy(man),sxy))):
> cartan__ijk:=get_compts(cartan__ijk_):
> cartanY__ijk_:=create([-1,-1,-1],eval(apply(copy(man),sxY))):
> cartanY__ijk:=get_compts(cartanY__ijk_):
> cartan_i_jk_:=simplify(prod(g_ij_,cartan__ijk_,[2,1])):
> cartan_i_jk:=get_compts(cartan_i_jk_):
> cartanY_i_jk_:=simplify(prod(gY_ij_,cartanY__ijk_,[2,1])):
> cartanY_i_jk:=get_compts(cartanY_i_jk_):
> prodcn_:=simplify(prod(cartan_i_jk_,N_i_j_,[2,1])):
```

```
> prodcn:=get_compts(prodcn_):
> prodcnY_:=simplify(prod(cartanY_i_jk_,NY_i_j_,[2,1])):
> prodcnY:=get_compts(prodcnY_):
> man:=array(sparse,1..n,1..n,1..n):
   for i from 1 to n do for j from 1 to n do for k from 1 to n do
      man[i,j,k]:=simplify(unapply(gammas[i,j,k],sxy)-
                  unapply(prodcn[i,j,k],sxy));
   end do;end do;end do:
> F_:=create([1,-1,-1],eval(apply(copy(man),sxy))):F:=get_compts(F_):
> FY_:=create([1,-1,-1],eval(apply(copy(man),sxY))):F:=get_compts(FY_):
```

Y-nonlinear connection

```
> Y_i_j__compts:=array(sparse,1..n,1..n):
   for i from 1 to n do for k from 1 to n do
      Y_i_j__compts[i,k]:=simplify(diff(Y[i],x[k])+NY[i,k]):
   end do;end do:
> Y_i_j_:=create([1,-1],eval(Y_i_j__compts)):Y_i_j:=get_compts(Y_i_j_):
> prodcY_:=simplify(prod(cartanY_i_jk_,Y_i_j_,[2,1])):
> prodcY:=get_compts(prodcY_):
```

Computing the components of Gammabar^i_jk connection

```
> gammabar__compts:=array(sparse,1..n,1..n,1..n):
   for i from 1 to n do for j from 1 to n do for k from 1 to n do
      gammabar__compts[i,j,k]:=simplify(FY[i,j,k]+prodcnY[i,j,k]);
   end do;end do;end do;
> gammabar_:=create([1,-1,-1],eval(gammabar__compts)):
> gammabar:=get_compts(gammabar_):
```

The surface X is declared

```
> tens(u,create([1],eval(array(sparse,1..n-1,[u1,u2,u3])))):get_compts(u_):
> X__compts:=array(1..n):
> X__compts[1]:=X1(u[1],u[2],u[3]):X__compts[2]:=X2(u[1],u[2],u[3]):
> X__compts[3]:=X3(u[1],u[2],u[3]):
> X_:=create([1],eval(X__compts)):X:=get_compts(X_):
```

Computing n-1 tangent vectors to the family M^(n-1)(c),B_a

```
> B_j_a:=array(1..n,1..n-1):
   for j from 1 to n do for a from  1 to n-1 do
      B_j_a[j,a]:=simplify(diff(X[j],u[a])):
   end do;end do: evalm(B_j_a):
```

Solving the system obtained from the transversality conditions

```
> meq:=array(1..n,1..n):
   for i from 1 to n do for j from 2 to n do s:=0: for r from 1 to n do
      s:=s+gY__ij[i,r]*Ysj[r,j]-gY__ij[j,r]*Y_i_j[r,i];
   end do;meq[i,j]:=s: end do;end do; eval(meq): meqq:=array(1..n):
   for i from 1 to n do s:=0: for r from 1 to n do
      s:=s+Y[r]*Y_i_j[i,r]:
   end do;meqq[i]:=s:end do: eval(meqq):
> meq12:=meq[1,2]:meq13:=meq[1,3]:meq23:=meq[2,3]:
> meqq1:=meqq[1]:meqq2:=meqq[2]:meqq3:=meqq[3]:
> declare(Y1(x[1],x[2],x[3]),Y2(x[1],x[2],x[3]),Y3(x[1],x[2],x[3])):
> sys:=simplify([numer(meq12)=0,numer(meq13)=0,numer(meq23)=0,numer(meqq1)=0,
     numer(meqq2)=0,numer(meqq3)=0]):
> #sol:=pdsolve(sys):
```

```
The induced Riemannian Y-metric , g__ab
> XY:=(seq(X[i],i=1..n),seq(Y[i],i=1..n)):
> gX__ij__compts:=array(symmetric,sparse,1..n,1..n):
   for i from 1 to n do for j from 1 to n do
      gX__ij__compts[i,j]:=simplify(D[i+n,j+n](ff)): end do;end do:
> gX__ij_:=create([-1,-1],eval(apply(gX__ij__compts,XY))):
> gX__ij:=get_compts(gX__ij_): g__ja:=array(1..n,1..n-1):
> for j from 1 to n do for a from 1 to n-1 do s:=0:for i from 1 to n do
    s:=simplify(s+gX__ij[j,i]*B_j_a[i,a]);g__ja[j,a]:=s;
  end do;end do;end do; evalm(g__ja):
> mg__ab:=array(1..n-1,1..n-1):
> for a from 1 to n-1 do for b from 1 to n-1 do s:=0:for j from 1 to n do
    s:=s+g__ja[j,a]*B_j_a[j,b];end do:
  mg__ab[a,b]:=s; end do;end do; evalm(mg__ab):
> g__ab__compts:=array(sparse,1..n-1,1..n-1):
> for a from 1 to n-1 do for b from 1 to n-1 do
     g__ab__compts[a,b]:=simplify(mg__ab[a,b]):
  end do;end do:
> g__ab_:=create([-1,-1],eval(g__ab__compts)): g__ab:=get_compts(g__ab_):
The inverse of the induced Riemannian Y-metric , g_ab, are computed
> g_ab_:= invert(g__ab_, 'detg__ab_'):g_ab:=get_compts(g_ab_):
Find the dual coframe, (B_a_i,Y__i)
> B_a_i:=array(1..n-1,1..n):
> for a from 1 to n-1 do for i from 1 to n do s:=0:
    for j from 1 to n do for b from 1 to n-1 do
        s:=simplify(s+g__ab[a,b]*B_j_a[j,b]*gX__ij[i,j]): B_a_i[a,i]:=s:
    end do;end do;
  end do;end do: eval(B_a_i):
> B_i_bg:=array(1..n,1..n-1,1..n-1):
> for i from 1 to n do for b from 1 to n-1 do for g from 1 to n-1 do
    B_i_bg[i,b,g]:=simplify(diff(B_j_a[i,b],u[g]));
  end do;end do;end do: eval(B_i_bg):
Find the coefficients of the connection gamma^a_bg(u)
> bgabar_i_bk:=array(1..n,1..n-1,1..n):
> for j from 1 to n do for b from 1 to n-1 do for k from 1 to n do
    for i from 1 to n do
        s:=0:s:=B_j_a[j,be]*gammabar[i,j,k]+s:bgabar_i_bk[i,b,k]:=s:
    end do; end do;end do;end do;eval(bgabar_i_bk):
> bgab_i_bg:=array(1..n,1..n-1,1..n-1):
> for i from 1 to n do for b from 1 to n-1 do for g from 1 to n-1 do
    for k from 1 to n do
        s:=0:s:=s+bgabar_i_bk[i,b,g]:bgab_i_bg[i,b,g]:=s:
    end do;
  end do;end do;end do;eval(bgab_i_bg):
> Bsuma_i_bg:=array(1..n,1..n-1,1..n-1):
> for i from 1 to n do  for b from 1 to n-1 do for g from 1 to n-1 do
    Bsuma_i_bg[i,b,g]:=B_i_bg[i,b,g]+bgab_i_bg[i,b,g]:
  end do;end do;end do:eval(Bsuma_i_bg):
> gamma_a_bg:=array(1..n-1,1..n-1,1..n-1):
> for a from 1 to n-1 do for b from 1 to n-1 do for g from 1 to n-1 do
    s:=0:for i from 1 to n do
        s:=s+B_a_i[a,i]*Bsuma_i_bg[i,b,g];gamma_a_bg[a,b,g]:=s;
    end do;end do;
  end do;end do;eval(gamma_a_bg):
```

The components B_i_axb of the covariant derivative along the surface X

```
> bg_i_ab:=array(1..n,1..n-1,1..n-1):
> for i from 1 to n do for a from 1 to n-1 do for b from 1 to n-1 do
    s:=0:for g from 1 to n-1 do
        s:=s+B_j_a[i,g]*gamma_a_bg[g,a,b];end do:
  bg_i_ab[i,a,b]:=s; end do;end do;end do: eval(bg_i_ab):
> B_i_axb:=array(1..n,1..n-1,1..n-1):
> for i from 1 to n do for a from 1 to n-1 do for b from 1 to n-1 do
    B_i_axb:=B_i_bg[i,a,b]+bgabar_i_bk[i,a,b]-bg_i_ab[i,a,b];
  end do;end do;end do; eval(B_i_axb):
```

The norm of Y, normaY, is determined

```
> prodgY_:=simplify(prod(gX__ij_,Y_,[2,1])):prodgY:=get_compts(prodgY_):
> pnormaY_:=simplify(prod(prodgY_,Y_,[1,1])):pnormaY:=get_compts(pnormaY_):
> normaY:=sqrt(pnormaY):
```

The mean curvature M of the Y-surface

```
> if normaY=0 then print("The norm of Y is 0");
            else print("the mean  curvature of a  (hyper)surface X is"):fi:
> Bg__abj:=array(1..n-1,1..n-1,1..n):
> for a from 1 to n-1 do for b from 1 to n-1 do for j from 1 to n do
    s:=0:for i from 1 to n do
        s:=s+B_i_axb[i,a,b]*gX__ij[i,j]:
    end do; Bg__abj[a,b,j]:=s;
  end do;end do;end do: eval(Bg__abj):
> H__ab:=array(1..n-1,1..n-1):
> for a from 1 to n-1 do for b from 1 to n-1 do
    s:=0:for j from 1 to n do
        s:=s+Bg__abj[a,b,j]*Y[j];
    end do; H__ab[a,b]:=s/normaY;
  end do;end do; eval(H__ab):
> s:=0:
> for a from 1 to n-1 do for b from 1 to n-1 do
    s:=g_ab[a,b]*H__ab[a,b]+s;
  end do;end do;
> M:=s;
#==========================================================================================================================================
```