THE PROGRAM ORTOCARTAN
FOR APPLICATIONS IN EINSTEIN'S RELATIVITY THEORY

Andrzej Krasiński, N. Copernicus Astronomical Center
Polish Academy of Sciences, Bartycka 18, 00 716 Warszawa, Poland

1. General remarks. ORTOCARTAN is written in LISP. For a non-expert user, the program can calculate the left-hand side of the Einstein's field equations from a given metric tensor. However, it contains fairly efficient sub-procedures for algebraic simplification, differentiation, substitutions and inverting matrices which can be used to write other agebraic programs in LISP with much less effort than the effort already invested. The sub-procedures are accessible to nonexpert users through a few "parasite" programs combining parts of ORTOCARTAN into simpler packages. This note describes some ideas used in constructing the algorithm.

2. Algebraic simplification. ORTOCARTAN can handle rational powers of rational numbers, simplifying e.g. $\sqrt{27}$ to $3\sqrt{3}$ or $8^{1/6}$ to $2^{1/2}$ automatically. With algebraic expressions, it avoids repeated simplifications by performing a thorough recursive simplification only on the user's input. Later, the level on which a simplification is needed is exactly controllable (e.g. when summing several expressions, only opportunities for cancellations or adding numerical factors must be looked for, re-analysing the terms of the sum is not necessary).

3. Differentiation. ORTOCARTAN has no difficulties differentiating functions whose arguments are other functions. The user-command (DERIV X <a function>) is always understood as the total derivative with respect to X. For instance, if x, y, z, t are independent variables and f = f (g, h, x), g = g (u, t), h = h (x, y, z), $u = x^2 + y^2 + z^2$, then (DERIV X F) will be printed as $2 X F_{,G} G_{,U} + F_{,H} H_{,X} + F_{,X}$, where commas denote partial derivatives. The arbitrary functions may be processed either with their arguments written out explicitly or with the arguments omitted, as in the example given. The arguments will be written out in subsequent calculations when the user writes them out on input. The dependencies of functions on their arguments are specified in a separate piece of input. It is also possible to represent definite algebraic expressions by atomic symbols so that the atom will be processed algebraically, but differentiation will enter the expression. The derivatives of the expression are then noted in the property list of the atom so that each derivative must be calculated only once; in later calls to differentiation they are simply retrieved from the property list.

4. Substitutions. ORTOCARTAN is not interactive, but the substitutions can be precisely directed to specified intermediate expressions which are printed out. Namely, each quantity is a component of a certain matrix. Each matrix has a unique name (appearing in print), and single components are identified by sets of indices. The substitution to be performed in a definite component is listed together with the appropriate set of indices, and the whole list is placed in the property list of the printname of the matrix with the indicator SUBS. In this way, the substitutions can be easily taken into account just before the quantity is printed and stored for further use. The program can replace parts of sums and products, e.g. the substitution B + D = U will

be performed in (A + B + C + D + E) producing (A + C + E + U). Such a substitution is equivalent to identifying a subset in a larger set and replacing it with another subset. The most recent version of ORTOCARTAN can also do some pattern-matching in the substitutions. The user can define certain variables (e.g. M and N) as MARKERS which can represent any expression. Then, (TAN M) = ((SIN M) / (COS M)) will mean that tan of any argument is to be replaced by sin/cos of the same argument. This is especially useful in processing truncated power series where by writing (E ** N) = 0 one ensures that only the terms linear in the parameter E will be kept while all higher powers will be discarded. The matching procedure recognizes the equality of functional forms of two expressions and assigns a value to each MARKER, the value being the sub-expression presently represented by the MARKER. The values are reassigned anew in each formula, so the same MARKER can represent different expressions in different formulae. The value is then inserted on the right-hand side of the equation in place of the MARKER, and the substitution is performed in the ordinary way. Each substitution is automatically followed by the appropriate algebraic simplification. The substitutions and the subsequent simplifications are performed in an economical way: the program begins to simplify a given expression only at the level where a replacement of a sub-expression actually occurred and proceeds from that level upwards. If no replacement occurred in an expression, then no simplification and no copying of list structures is performed.

5. Inverting matrices. A special "parasite" program is available which inverts matrices of arbitrary rank with algebraic elements. It calculates a determinant by making the matrix triangular, and so is applicable even to large matrices (rank $\leq$ 40) provided the determinant to be expected is not a huge sum.

6. Printing the results is performed by a LISP-program which is thus moderately portable. The program can print everything in the ordinary mathematical format, even if the exponents in exponential expressions have themselves exponents and subscripts. Exponents can be built upwards to an arbitrary height, subscripts to each line must be all placed in one level.

7. Missing features. Integration, factorizing polynomials and simplifying rational functions were left out deliberately because they would make the program too large as for our facilities and too complicated for our manpower (1 at present). Other features (e.g. complex numbers, Taylor expansions, writing programs in ORTOCARTAN itself without resorting to LISP) may be added in the future.

8. Availability. ORTOCARTAN is available in 4 implementations: 1. In Cambridge LISP on IBM 360/370 computers; 2. In University of Texas LISP 4.1 on CDC Cyber computers; 3. In SLISP/360 on IBM 360/370 and Siemens 4004 computers; 4. In LISP 1108 on UNIVAC 1100 computers. The minimum core required for medium size problems is 40000 words on CDC and 500 kbytes on IBM. The "parasite" programs will run with 30000 words and 300 kbytes. The IBM and UNIVAC versions were not updated since 1983, only the CDC version is constantly maintained. In particular, the pattern-matching is available only in the CDC version.

9. Acknowledgments. The program was written in cooperation with M. Perkowski and Z. Otwinowski. Parts of the work were supported by the Alexander von Humboldt Foundation (Bonn) and by the Deutsche Forschungsgemeinschaft.