NASA CR-152020

M. Rubesin

A.R.A.P.

AERONAUTICAL RESEARCH ASSOCIATES *of* PRINCETON, INC.

NASA CR-152020

A.R.A.P. REPORT NO. 309

COMPUTER TRANSFORMATION OF PARTIAL DIFFERENTIAL EQUATIONS INTO ANY COORDINATE SYSTEM

ROGER D. SULLIVAN

JUNE 1977

PREPARED UNDER CONTRACT NO. NAS2-8014

BY

AERONAUTICAL RESEARCH ASSOCIATES OF PRINCETON, INC.
50 WASHINGTON ROAD, PRINCETON, NEW JERSEY 08540

FOR

AMES RESEARCH CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

ABSTRACT

Tensors provide a compact way of writing partial differential equations in a form valid in all coordinate systems. However, in order to find solutions of the equations with their boundary conditions they must be expressed in terms of the coordinate system under consideration. For complicated coordinates the expressions are very long and are tedious to calculate. The process of arriving at them from the tensor formulation has been automated by a software system, TENSR, described in this paper. An allied system that analyzes the resulting expressions term by term and drops those that are negligible is also described.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

The work reported here has been supported, for the most part, by several government contracts. These were awarded chiefly by the National Aeronautics and Space Administration, through both the Langley Research Center and the Ames Research Center, and by the Air Force Office of Scientific Research. The personnel of these agencies have been most helpful and cooperative.

Many of my colleagues at A.R.A.P. have made contributions; I particularly want to mention the following. Barry Gilligan wrote the original version of TENSR and it is still very much his. In particular, it was he who figured out how to break down parentheses and how to produce nested loops with variable depths of nesting. Peter Woodrow gave us courage by his certitude that it could be done and provided some tools that smoothed the way. John Yates helped TENSR become a reality through his contributions to its formulation. The enthusiasm of Guido Sandri has been a constant help. John Leonard made important contributions to TATTR, and Robert Beddini has served ably as the test pilot. Coleman Donaldson and Stokes Fishburne have shown great patience and have given unstinting support. And it was Coleman Donaldson who first showed me that tensors are useful, rather than just something to be admired.

To all of them and to many others who have braved blizzards of indices I give my thanks.

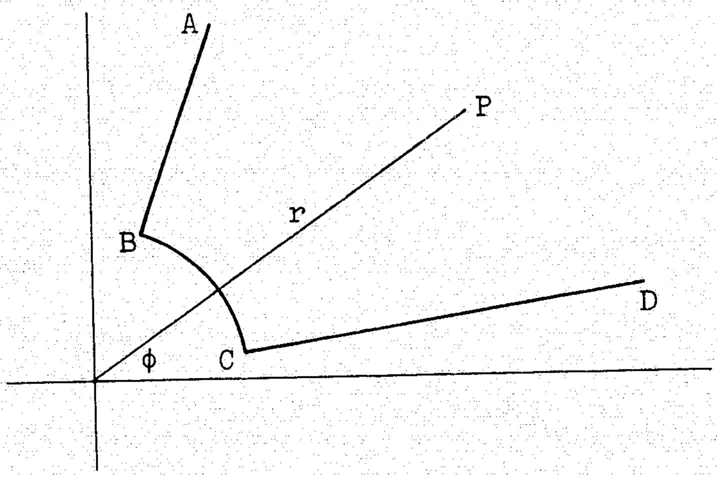
1. INTRODUCTION

It is natural when computing the flowfield about a cone to use a conical coordinate system. The description of the body surface in that system is simply the statement that one of the coordinates is equal to a constant, for example

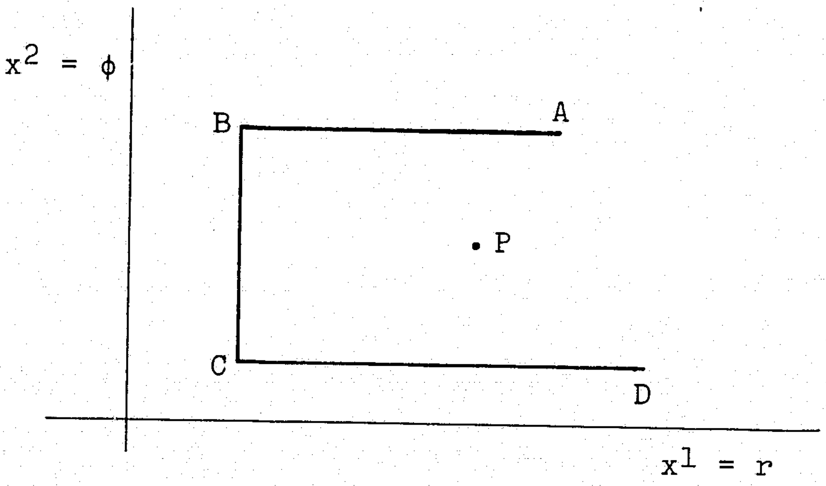
$$\theta = \theta_c$$

This makes the application of the boundary conditions to the equations of motion straightforward at the cost of complicating the equations themselves. The extra complication is worthwhile compared to the computational problems that arise when boundaries cut across coordinate surfaces.

Such a scheme can be regarded in two ways. In one view the conical coordinate system is an entity in Euclidean space which is more convenient than the Cartesian system for the problem at hand. In another view it is a mapping of the body and the flowfield around it into a rectangular grid. To illustrate with a two-dimensional example, a polar coordinate system can be viewed as a way of locating points in physical space, as in this sketch:



or as a mapping from that space to this:



The mathematics is the same in both cases, but the mapping interpretation makes it manifest that one gets the advantages of finite-difference methods for a rectangular grid along with simple boundary conditions.

Coordinate systems or mappings can also be devised so that points are unevenly spaced in physical space (so as to be concentrated near a body, for example) but mapped to evenly spaced points in the rectangular grid, again gaining an advantage for finite-difference computations.

The gains, however, are generally at the cost of more complicated equations to be solved. Nevertheless, the advantages of such a procedure for computation are so great that Thompson, Shanks, and Walker (Reference 1) use a time-dependent version that keeps a free surface, in the problem they consider, mapped into a fixed coordinate line. Thus at each time step of their calculation they recompute the mapping as an aid in computing the flow.

So far, only transformation of the independent variables of a problem has been mentioned. It is often desirable to transform the dependent variables, in the sense that the representation of vectors changes from components aligned with the original coordinates to components aligned with the new coordinates. For example, in transforming from Cartesian to polar coordinates in the plane, the representation of velocities would change from horizontal and vertical components to radial and tangential components. This also is ordinarily advantageous for the expression of boundary conditions. Its effect on the complexity of the equations is varied. There is a gain in unifying the representation of dependent and independent variables, but on the other hand new terms arise. The prototype of such terms is the centripetal acceleration term which turns up when radial and tangential velocity components are used.

At first glance it would seem that the transformation of the dependent variables as well as the independent variables would greatly complicate the task. However, the techniques of tensor analysis accomplish both transformations in one step with great economy. On the other hand, the compact tensor notation can disguise very long expressions which must be expanded before much can be accomplished numerically. (For tensor expressions which involve known quantities there is no problem in writing programs to calculate them. It is expressions involving unknowns that are principally considered here.)

As the body shapes considered grow more complicated, the coordinate systems adapted to them grow more complicated, the equations of motion expressed in terms of them grow longer, and the likelihood of getting them written down correctly decreases if the operation is done by hand. But the process at this stage is purely mechanical and so can be done by computer. The software system TENSR accomplishes just that.

The original version of TENSR, though, was produced to answer a different need. Since 1967 A.R.A.P. has been engaged in second-order closure techniques for calculating turbulent flows, and since 1968 the modeling has been done in the framework of tensor analysis in order to eliminate the bias that can easily be introduced when working in a fixed coordinate system. The principal unknown in our system of equations for turbulent flow, besides the variables describing the mean flow, is a tensor of second rank, namely the correlation of components of the velocity fluctuations, $\langle u_i' u_j' \rangle$. The governing equations are quite complex even in Cartesian coordinates. We achieved good results solving these equations for incompressible flows (Reference 2) but not without considerable problems in accurately developing computer programs for the purpose.

Therefore, when we approached the task of solving the equations of turbulent compressible flows — for which fluctuations of density, temperature, viscosity, and so forth, as well as of velocity, must be taken into account — we decided to automate the process even though only Cartesian coordinates were to be used, since the study involved the boundary layer on a flat plate (Reference 3). Thus, TENSR was born able to handle only Cartesian coordinates.

The output of the original TENSR was edited by hand to remove terms that are small in the context of boundary-layer theory. After TENSR was extended to be able to handle arbitrary systems, more complicated coordinates were processed and the output grew so much that it was desirable to make the dropping of terms automatic. For this purpose, a follow-on system, TATTR (for Term ATTRibutes), was written. The use of this program not only eased the burden of dropping terms but also forced the process to be more uniform and logical than had sometimes been the case in our work.

Another software system, DIFFR, uses the edited output of TENSR or of TATTR to create Fortran statements which become a main part of programs that solve the equations by finite-difference techniques.

This paper reports the operation and characteristics of TENSR and TATTR. Since the present version of DIFFR is tied to a specific finite-difference scheme, it is not further described here.

In the next section the notation used to handle tensors on the computer is introduced in the course of a brief lesson on tensor concepts. Sections 3 and 4 show some examples of what TENSR and TATTR do. These are followed by a section on possible future developments and on related efforts by others. Some characteristics of TENSR that may interest tensor experts are considered in Section 6. Finally there is a brief bibliography.

2. TENSORS: A QUICK LESSON

One view of tensors, sufficient for the understanding of TENSR, is given here. Readers already familiar with the subject are advised to skim the section to gain familiarity with the notation we have devised to communicate tensor expressions with computers. As is explaiend in Section 6, TENSR is not restricted in the type of space it deals with (except for the upper limit of ten dimensions) but applications such as those mentioned in Section 1 are met with in an ordinary three-dimensional Euclidean space so this lesson is confined to that case or a two-dimensional subset of it. The treatment here is expository; proofs may be found in the works cited in the bibliography.

In the powerful index notation used in tensor analysis the equation

$$C_i = A^n{}_i B_n \qquad (1)$$

stands (in three dimensions) for three equations each with three terms on the right:

$$\begin{aligned} C_1 &= A^1{}_1B_1 + A^2{}_1B_2 + A^3{}_1B_3 \\ C_2 &= A^1{}_2B_1 + A^2{}_2B_2 + A^3{}_2B_3 \qquad (2) \\ C_3 &= A^1{}_3B_1 + A^2{}_3B_2 + A^3{}_3B_3 \end{aligned}$$

The rules governing symbolic indices (that is, non-numeric indices) are these:

- A symbolic index appearing alone in a term, known as a free index, takes on the range of values in separate instances.

- A symbolic index appearing twice in a term, known as a dummy index, implies a summation over the range of values (the Einstein summation convention).

- A symbolic index doesn't appear three or more times in a single term without a special indication of the meaning to be assigned to the expression.

Thus, the usual apparatus that would accompany the condensation of equations (2), for example:

$$C_i = \sum_{n=1}^{3} A^n{}_i B_n \qquad (i = 1,2,3)$$

is eliminated, leaving just equation (1).

Some additional rules, important in the general case, but unnecessary and therefore often disregarded when only Cartesian coordinates are considered, are:

- Dummy indices in a term appear once up (as a superscript) and once down (as a subscript or in a derivative) for each pair.

- Free indices appear in the same position in each term in an equation.

The comparison of equations (1) and (2) gives only a hint of the economy of this notation. In three dimensions an equation with three free indices, e.g., equation (7) below, stands for 27 equations (not necessarily all independent) and a term with four pairs of dummy indices, e.g., in equation (8) below, stands for the sum of 81 terms.

In devising a notation to use on the computer, we decided to adopt conventions like those of Fortran in many respects, but the Fortran subscript representation was discarded since it lacked a means of representing superscripts and since it is unwieldly when expressions teem with indices. Instead, we decided that indices would consist of only one letter or digit and that superscripts would be preceded by " and subscripts by ' . Thus, equation (1) appears as

```
C'I = A"N'I*B'N
```

Since variables are not restricted to one character, this might appear as

```
CRS'I = AAA1"N'I*BS'N
```

Several superscripts or subscripts need not have separate " or ' . Thus, A^{ij} is represented by A"IJ .

Let a Cartesian coordinate system be denoted by y^i and some other system by x^i . Functions relating one to the other are assumed to exist and in most applications are known in advance. For an example in two dimensions, polar coordinates are related to Cartesian coordinates by

$$y^1 = x^1 \cos x^2$$

$$y^2 = x^1 \sin x^2$$

Actually, the x^i are seldom used in this context; it is more convenient to use r and ϕ :

$$y^1 = r \cos \phi$$

$$y^2 = r \sin \phi$$

But in speaking of the coordinates collectively, or especially in speaking of coordinate systems generally, the notation x^i is used.

Scalar fields, that is, scalars which are functions of position, such as density, are tensors of rank zero. The gradient of such a field is a tensor of rank one, that is, a vector. Vectors, like scalars, are invariant with respect to coordinate transformations. However, the representation of a vector in components does vary with the coordinate system. A representation in components which transform as do the partial derivatives of a scalar field is called a covariant representation of the vector and the indices appear as subscripts. For example, we might set

$$C_i = \frac{\partial \Psi}{\partial x^i} = \Psi_{:i}$$

(The colon is used to denote partial differentiation both in normal notation and in TENSR notation: C'I = PSI:I .) A representation in components which transform as do differential displacement components is called a contravariant representation of the vector and the indices appear as superscripts. For example

$$u^i = \frac{dx^i}{dt}$$

Covariant representations can be calculated from contravariant representations and vice versa by the formulas

$A_i = g_{in}A^n$ or A'I = @'IN*A"N

$A^j = g^{jm}A_m$ or @"J = @"JM*A'M

This is called lowering or raising the index. In place of the phrase covariant (contravariant) representation of a vector, one often speaks concisely of a covariant (contravariant) vector. Also, an index is often called covariant or contravariant according to its position as a subscript or superscript.

As a mnemonic, note that sub- and co- are short, whereas super- and contra- are long. It should be clear that covariant representations of vectors are not necessarily partial derivatives of scalars nor are contravariant representations of vectors necessarily differentials or velocities.

Higher rank tensors may have representations which are covariant (all subscripts), contravariant (all superscripts), and mixed (some of each) and the indices may be raised and lowered individually. For example:

$T^{i}{}_{j} = g^{in}T_{nj}$ or T"I'J = @"IN*T'NJ

$T_{i}{}^{j} = g^{j\ell}T_{i\ell}$ or T'I"J = @"JL*T'IL

$T^{ij} = g^{j\ell}g^{in}T_{n\ell}$ or T"IJ = @"JL*@"IN*T'NL

Equation (3) can be looked at as raising the first index of g_{ij} or lowering the second of g^{ij} ; in either case, we conclude that

$g^{i}{}_{j} = \delta^{i}{}_{j}$ or @"I'J = &"I'J

That is, the value of the mixed representation of the metric tensor is the Kronecker delta in any coordinate system. This contrasts with the fact that the value of g_{ij} is δ_{ij} and the value of g^{ij} is δ^{ij} in Cartesian coordinates only.

If a superscript and a subscript of a tensor are made the same, thus implying summation, a new tensor is formed of rank two less than the original — a process called contraction. Thus, the two vectors

$A_{i} = T^{n}{}_{ni}$ or A'I = T"N'NI

$B_{i} = T^{n}{}_{in}$ or B'I = T"N'IN

are different (in general) contractions of $T^i{}_{jk}$ or T"I'JK . Some tensors are the product of tensors. A tensor of rank p multiplied by one of rank q yields a tensor of rank $p + q$. The contraction of the product of two vectors, $A^n B_n$ or A"N*B'N is the scalar product of the vectors.

In addition to covariant and contravariant components of tensors there are also physical components, that is, components in the elementary sense. Velocity in polar coordinates provides an instructive example. The physical components are denoted by u for radial and v for tangential, the covariant and contravariant components by u_i and u^i . In this system the radial component is the same in all three versions, covariant, contravariant, or physical:

$$u_1 = u^1 = u \tag{4}$$

On the other hand, the three types of tangential component are quite different. The contravariant component can be described as an angular velocity, since

$$u^2 = \frac{v}{r} \tag{5}$$

whereas the covariant component is more a circulation, since

$$u_2 = rv \tag{6}$$

See References 4 and 5, in particular, for more on physical components.

The representation in polar coordinates of the velocity of a body moving in a circle at constant speed is unchanging. That is, the radial component is constant (at zero) and the tangential component is constant. Nevertheless, as is well known, the body is accelerating toward the center and the equations of motion in polar coordinates represent the situation by the centripetal acceleration term. The generalization of this term to situations involving arbitrary changes in tensors using arbitrary coordinate systems is embodied in Christoffel symbols. The Christoffel symbol of the first kind is given by

$$\$_{ijk} = \frac{1}{2}\left(g_{ij:k} + g_{ik:j} - g_{jk:i}\right)$$

or (7)

`$'IJK = (@'IJ:K + @'IK:J - @'JK:I)/2`

(The first choice for TENSR notation for the Christoffel symbols was ¢ , but A.R.A.P.'s line printer didn't have it available so $\$$ was chosen as the next best thing. Since the literature is not in agreement on notation for them and since the ordering of indices that I find convenient — that is, such that $\$_{ijk}$ is symmetric with respect to j and k — is unorthodox, I have adopted the practice of using $\$$ in ordinary as well as TENSR notation.) The Christoffel symbol of the second kind is

$$\$^{i}{}_{jk} = g^{in}\$_{njk} \quad \text{or} \quad \texttt{\$"I'JK = @"IN*\$'NJK}$$

For our purposes, the first kind has no function except to allow the calculation of the second kind. Neither kind of Christoffel symbol is a tensor.

Covariant differentiation, denoted by a comma preceding an index, is an operation that produces tensors from tensors. When applied to a contravariant vector (to slip into that terminology) its rule for calculation is

$$A^{i}{}_{,j} = A^{i}{}_{:j} + \$^{i}{}_{\alpha j}A^{\alpha}$$

Neither term on the right is a tensor, but the sum is. The covariant derivative of a covariant vector is

$$A_{i,j} = A_{i:j} - \$^{\alpha}{}_{ij}A_{\alpha}$$

The generalization to tensors of higher rank is straightforward. For example:

$$A^{ij}{}_{k,r} = A^{ij}{}_{k:r} + \$^{i}{}_{\alpha r}A^{\alpha j}{}_{k} + \$^{j}{}_{\alpha r}A^{i\alpha}{}_{k} - \$^{\alpha}{}_{kr}A^{ij}{}_{\alpha}$$

The rule is not difficult. Each index in the original tensor is replaced by a dummy in one of the extra terms. The rest of the index juggling is automatic by the rules for indices. The only aspect that is not automatic is the sign that accompanies each extra term. (The user of TENSR need not bother with any of this — TENSR does it all automatically.)

Since a scalar has no indices, its covariant derivative is simply its partial derivative:

$$\Psi_{,i} = \Psi_{:i}$$

This agrees with the previous assertion that the gradient of a scalar is a tensor. In Cartesian coordinates the Christoffel symbols are all zero; therefore, partial derivatives in Cartesian coordinates are also covariant derivatives.

It is convenient to have a notation for what might be called a contravariant derivative, that is, a covariant derivative with the index raised. Again, the literature is not consistent; TENSR uses an exclamation point:

```
PSI!K = @"KM*PSI,M
```

This brief lesson misses much of the splendor and importance of tensors. The reader is urged to consult some of the works cited in the bibliography for a better view. It should be noted that many of those books start by considering abstract spaces for which a metric does not exist. In that case there can be no raising or lowering of indices, and so covariant and contravariant vectors are truly different types of entities, unrelated to each other. Once spaces with metrics are considered, however, the viewpoint becomes closer to that presented here.

3. SOME TENSR SAMPLES

In ordinary applications, TENSR accepts information concerning the metric tensor and Christoffel symbols for a coordinate system, along with symbol substitutions to be made, and one or more equations in general tensor form. It proceeds to expand (internally) covariant derivatives into the sum of partial derivatives and Christoffel-symbol terms, to expand implied summations, to drop terms which are zero, to make substitutions specified, and to print the results and store them on disk.

For a short example, consider the transformation to polar coordinates of the Navier-Stokes equations for the steady flow of a fluid of constant density. Figure 1 shows the printout from such a run. The first seven lines are exact images of input cards. The symbol # is used to indicate comments; anything to the right of it is ignored by TENSR. The first card sets the range of dummy index summations as 1 and 2, and hence establishes that a two-dimensional space is to be considered. The next two cards define the metric; since the metric tensor is symmetric only three of the four elements need be supplied in each case. A blank followed by the separator ; suffices to indicate a zero element. Fortran conventions concerning the order of operations are observed; thus 1/R/R = (1/R)/R stands for r^{-2} . The fourth card supplies values for the Christoffel symbol of the second kind. Again because of symmetry, only six of the eight values are specified. Symbol substitutions are designated by the next card. In this case expressions involving physical components are to be substituted for covariant and contravariant components [compare equations (4)-(6)]. The second DEFINE INDEX card identifies K as a free index; in the absence of other indication the range of the free index is the same as the range of dummy indices. Finally, the equation to

```
     DEFINE INDEX  1, 2

     DEFINE  @"  1;  ; 1/R/R                                # POLAR

     DEFINE  @'  1;  ; R*R                                  # POLAR

     DEFINE  $"   ;  ; -R;   ; 1/R;  ;                      # POLAR

     DEFINE SYMBOL  U"1 = U; U"2 = V/R;  U'1 = U; U'2 = R*V  # POLAR

     DEFINE INDEX   K

     U"L*U'K,L = -P,K + NU*@"LN*U'K,LN              # NAVIER-STOKES

U"L*U'K,L = -P,K + NU*@"LN*U'K,LN

U*(U):1 + V/R*(U):2 - V/R*1/R*R*V = -(P):1 + NU*((U):1):1
     + NU*1/R/R*((U):2):2 - NU*1/R/R*(1/R*R*V):2 - NU*1/R/R*1/R*(R*V):2
     - NU*1/R/R*1/R*R*U + NU*1/R/R*R*(U):1

U*(R*V):1 + V/R*(R*V):2 - U*1/R*R*V + V/R*R*U = -(P):2
     + NU*((R*V):1):1 + NU*1/R/R*((R*V):2):2 - NU*(1/R*R*V):1
     + NU*1/R/R*(R*U):2 - NU*1/R*(R*V):1 + NU*1/R/R*R*(U):2
     + NU*1/R*1/R*R*V - NU*1/R/R*R*1/R*R*V + NU*1/R/R*R*(R*V):1
     - NU*1/R/R*R*1/R*R*V
```

Figure 1. The Navier-Stokes equations in polar coordinates.

be expanded appears. Here P stands for the pressure divided by the density and NU for the kinematic viscosity.

After TENSR checks for errors such as unbalanced parentheses, it confirms its verification by reprinting the equation. Then the results appear: the radial momentum equation and the tangential momentum equation, that is to say the Navier-Stokes equations in polar coordinates. There in a nutshell is the action of TENSR.

That TENSR is weak algebraically is evident. Some of these terms can be combined and many can be simplified. For example, the third term in the radial equation is hardly recognizable as the centripetal acceleration term since it appears as a transcription of the original form, -U"2*$"2'12*U'2 , without simplification to -V*V/R . In the original applications of TENSR, involving Cartesian coordinates, this shortcoming was very minor and so was allowed to stand. When it later became a problem, it was more convenient to deal with it in TATTR, the follow-on to TENSR, rather than in TENSR itself (see Section 4).

To illustrate the functions of the DEFINE cards, Figure 2 shows the output of a run with all but the first omitted. The equation to be expanded is the same but the result is quite different. Here, there are none of the replacements and dropping of terms that took place in the run illustrated in Figure 1. The free index, K , appears as such in each term. For each value of K (1 and 2), this represents one of the two-dimensional Navier-Stokes equations for arbitrary coordinates.

The explosion of terms that can occur begins to show here. Applying the rules for covariant derivatives given in Section 2, the viscosity term can be written

```
     DEFINE INDEX  1, 2


     U"L*U'K,L = -P,K + NU*@"LN*U'K,LN                  #  NAVIER-STOKES

U"L*U'K,L = -P,K + NU*@"LN*U'K,LN

U"1*(U'K):1 + U"2*(U'K):2 - U"1*$"1'K1*U'1 - U"1*$"2'K1*U'2

  - U"2*$"1'K2*U'1 - U"2*$"2'K2*U'2 = -(P):K + NU*@"11*((U'K):1):1

     + NU*@"12*((U'K):1):2 + NU*@"21*((U'K):2):1 + NU*@"22*((U'K):2):2

     - NU*@"11*($"1'K1*U'1):1 - NU*@"11*($"2'K1*U'2):1

     - NU*@"12*($"1'K1*U'1):2 - NU*@"12*($"2'K1*U'2):2

     - NU*@"21*($"1'K2*U'1):1 - NU*@"21*($"2'K2*U'2):1

     - NU*@"22*($"1'K2*U'1):2 - NU*@"22*($"2'K2*U'2):2

     - NU*@"11*$"1'K1*(U'1):1 - NU*@"11*$"2'K1*(U'2):1

     - NU*@"12*$"1'K2*(U'1):1 - NU*@"12*$"2'K2*(U'2):1

     - NU*@"21*$"1'K1*(U'1):2 - NU*@"21*$"2'K1*(U'2):2

     - NU*@"22*$"1'K2*(U'1):2 - NU*@"22*$"2'K2*(U'2):2

     + NU*@"11*$"1'K1*$"1'11*U'1 + NU*@"11*$"1'K1*$"2'11*U'2

     + NU*@"11*$"2'K1*$"1'21*U'1 + NU*@"11*$"2'K1*$"2'21*U'2

     + NU*@"12*$"1'K2*$"1'11*U'1 + NU*@"12*$"1'K2*$"2'11*U'2

     + NU*@"12*$"2'K2*$"1'21*U'1 + NU*@"12*$"2'K2*$"2'21*U'2

     + NU*@"21*$"1'K1*$"1'12*U'1 + NU*@"21*$"1'K1*$"2'12*U'2

     + NU*@"21*$"2'K1*$"1'22*U'1 + NU*@"21*$"2'K1*$"2'22*U'2

     + NU*@"22*$"1'K2*$"1'12*U'1 + NU*@"22*$"1'K2*$"2'12*U'2

     + NU*@"22*$"2'K2*$"1'22*U'1 + NU*@"22*$"2'K2*$"2'22*U'2
```

Figure 2. The Navier-Stokes equations for general coordinates in two dimensions. (Continued on next page)

```
- NU*@"11*$"1'11*(U'K):1 - NU*@"11*$"2'11*(U'K):2
- NU*@"12*$"1'12*(U'K):1 - NU*@"12*$"2'12*(U'K):2
- NU*@"21*$"1'21*(U'K):1 - NU*@"21*$"2'21*(U'K):2
- NU*@"22*$"1'22*(U'K):1 - NU*@"22*$"2'22*(U'K):2
+ NU*@"11*$"1'11*$"1'K1*U'1 + NU*@"11*$"1'11*$"2'K1*U'2
+ NU*@"11*$"2'11*$"1'K2*U'1 + NU*@"11*$"2'11*$"2'K2*U'2
+ NU*@"12*$"1'12*$"1'K1*U'1 + NU*@"12*$"1'12*$"2'K1*U'2
+ NU*@"12*$"2'12*$"1'K2*U'1 + NU*@"12*$"2'12*$"2'K2*U'2
+ NU*@"21*$"1'21*$"1'K1*U'1 + NU*@"21*$"1'21*$"2'K1*U'2
+ NU*@"21*$"2'21*$"1'K2*U'1 + NU*@"21*$"2'21*$"2'K2*U'2
+ NU*@"22*$"1'22*$"1'K1*U'1 + NU*@"22*$"1'22*$"2'K1*U'2
+ NU*@"22*$"2'22*$"1'K2*U'1 + NU*@"22*$"2'22*$"2'K2*U'2
```

Figure 2. The Navier-Stokes equations for general coordinates in two dimensions. (Continued from preceding page)

$$\begin{aligned} \nu g^{\ell n} u_{k,\ell n} = \nu g^{\ell n} u_{k:\ell n} &- \nu g^{\ell n}\left(\$^{\alpha}{}_{k\ell} u_{\alpha}\right)_{:n} \\ &- \nu g^{\ell n}\$^{\beta}{}_{kn} u_{\beta:\ell} + \nu g^{\ell n}\$^{\beta}{}_{kn}\$^{\alpha}{}_{\beta\ell} u_{\alpha} \\ &- \nu g^{\ell n}\$^{\beta}{}_{\ell n} u_{k:\beta} + \nu g^{\ell n}\$^{\beta}{}_{\ell n}\$^{\alpha}{}_{k\beta} u_{\alpha} \end{aligned} \tag{8}$$

When the implied summations are expanded in the two-dimensional case, two of these terms yield 16 terms each, three yield 8 terms each, and the first yields 4 terms for a total of 60 out of the 67 terms in the equation. (The equations of Figure 1 are very much shorter because all of the Christoffel symbols except $\$^{1}{}_{22} = -r$ and $\$^{2}{}_{12} = \$^{2}{}_{21} = 1/r$ are zero for the simple polar coordinate system.) If the first card in this example had been omitted too, the default range for dummies, 1, 2, and 3, would have applied. The result would have been an equation of 265 terms (252 of them viscosity terms): one of the Navier-Stokes equations in arbitrary coordinates for three dimensions. Since this is one of the simplest of the equations used to study fluid flows, the usefulness of TENSR can be appreciated.

Another example introduces some more features. Figure 3 shows a run deriving the axial momentum equation for an axisymmetric jet with no swirl. The cylindrical coordinate system given by X"1 = R , X"2 = Φ , X"3 = Z is used and the corresponding physical velocity components are denoted U , V , W . The differences between the input to this run and that of Figure 1 are: (1) The range of the dummy indices is set by default. (2) There are more elements of the metric tensor and of the Christoffel symbol to establish, and there are more substitutions to be specified. (3) The continuation of the preceding line is indicated by a character in column 6 as in Fortran. (4) The DEFINE NULL card establishes axisymmetry by indicating that derivatives with respect to Φ (:2) are zero

```
      DEFINE  @"  1;  ;  ; 1/R/R;  ; 1                 #  CYLINDRICAL

      DEFINE  @'  1;  ;  ; R*R;  ; 1                   #  CYLINDRICAL

      DEFINE  $"   ;  ;  ; -R;  ;  ;                   #  CYLINDRICAL
     C             ; 1/R;  ;  ;  ;  ;                  #  CYLINDRICAL
     C             ;  ;  ;  ;  ;  ;                    #  CYLINDRICAL

      DEFINE SYMBOL  U"1 = U ;  U"2 = V/R ;  U"3 = W ;  #  CYLINDRICAL
     C               U'1 = U ;  U'2 = R*V ;  U'3 = W    #  CYLINDRICAL

      DEFINE NULL  :2 ;  U"2 ;  U'2

      DEFINE INDEX  K = 2, 3

      U"L*U'K,L = -P,K + NU*U'K!L,L                   #  NAVIER-STOKES

   U"L*U'K,L = -P,K + NU*U'K!L,L

   0 = 0

   U*(W):1 + W*(W):3 = -(P):3 + NU*((W):1):1 + NU*((W):3):3

      + NU*1/R*(W):1
```

Figure 3. The axial momentum equation in cylindrical coordinates.

and specifies no swirl by indicating that U"2 and U'2 are zero. (This being the case the presence of U"2 and U'2 on the DEFINE SYMBOL cards is superfluous, but it is convenient to have them there for other applications.) (5) Since not all three values of K are desired, the DEFINE INDEX card specifies which are. For the axial equation only, K = 3 would appear, but the value 2 has been included to show that TENSR handles this extreme case, 0 = 0 , correctly. (6) The viscosity term has been written with the "contravariant derivative" symbol, ! , instead of explicitly using the metric tensor as in Figures 1 and 2.

These examples show the general features of TENSR and demonstrate that it produces correct results. They only hint at its necessity; the modeled equations for turbulence in use at A.R.A.P. generate thousands of terms for quite simple orthogonal coordinate systems (Reference 6).

4. SOME TATTR SAMPLES

The equations of Figure 1 were submitted to TATTR with the results shown in Figure 4. In addition to the equations, TATTR was given the following information regarding the dimensions (in the sense of dimensional analysis) of the quantities that appear as factors in those equations: U and V are velocities, P (being pressure divided by density) is a velocity squared; NU is a velocity times a length; R is a length; differentiation with respect to R (:1) is equivalent to dividing by a length; and differentiation with respect to Φ (:2) is dimensionless. These were supplied as the exponents to be attached to a length and a velocity: 1 and 1 for NU , 0 and 2 for P , and so forth. TATTR has simplified and combined terms, as can be seen by comparing Figures 1 and 4. However, the simplification is not complete since TATTR does not recognize that R can be taken out of a Φ derivative; for example, the seventh and eighth terms in the first equation could be combined. Nevertheless, the improvement is significant.

TATTR has also printed the exponents giving the dimensions of each term next to it. It is seen that both equations are dimensionally consistent. This is rather trivial in such a simple example, but the ability of TATTR to make this check has uncovered errors in long equations that might otherwise have gone undetected.

The dimensions of a term are examples of one type of attribute that TATTR can evaluate. Any concept that can be quantified by integers attached to the factors that appear, such that the integers add (or subtract) when the factors multiply (or divide) to form a term, is valid as an attribute for the purposes of TATTR. The type of attribute which motivated the design of TATTR is order-of-magnitude information. To illustrate, the axial-momentum equation of Figure 3 was used.

```
                                        LENGTH  VELOCITY

U*(U):1                                   -1       2
  + 1/R*V*(U):2                           -1       2
  - 1/R*V*V                               -1       2
   = -(P):1                               -1       2
    + NU*((U):1):1                        -1       2
    + NU/R/R*((U):2):2                    -1       2
    - NU/R/R*(V):2                        -1       2
    - NU/R/R/R*(R*V):2                    -1       2
    - NU/R/R*U                            -1       2
    + NU/R*(U):1                          -1       2
```

```
                                        LENGTH  VELOCITY

U*(R*V):1                                  0       2
  + 1/R*V*(R*V):2                          0       2
   = -(P):2                                0       2
    + NU*((R*V):1):1                       0       2
    + NU/R/R*((R*V):2):2                   0       2
    - NU*(V):1                             0       2
    + NU/R/R*(R*U):2                       0       2
    + NU/R*(U):2                           0       2
    - NU/R*V                               0       2
```

Figure 4. The Navier-Stokes equations in polar coordinates.

In addition to the equation, TATTR was supplied with attribute information similar to that for the run of Figure 4 plus data on an order of magnitude attribute, as follows:

	Length	Velocity	Delta
U	0	1	1
W	0	1	0
P	0	2	0
NU	1	1	2
R	1	0	1
:1	-1	0	-1
:3	-1	0	0

To understand the last column it is easiest to reinterpret the equation as nondimensional, NU standing for the reciprocal of the Reynolds number. (It seems strange at first to assign dimensions to the factors and at the same time to consider them nondimensional, but it is perfectly legal for both interpretations to coexist.) It is recognized that in a jet at high Reynolds number the radial component of velocity is small compared to the axial. Further, the radial extent of the flow due to the jet is small compared to its axial extent, so that changes (derivatives) in the radial direction are large compared to changes in the axial direction. Elementary boundary layer theory teaches us that if the reciprocal of the Reynolds number is of order δ^2 where δ is small, then the radial velocity and extent are of order δ compared to the corresponding axial quantities. The last column above, then, gives the exponents of δ that indicate the order of the various quantities.

The result of the TATTR run is shown in Figure 5; it is seen that the terms are dimensionally consistent but that one is small (of order δ^2) compared to the others. In a succeeding

```
                                        LENGTH VELOCITY    DELTA

U*(W):1                                   -1        2         0
   + W*(W):3                              -1        2         0
     = -(P):3                             -1        2         0
       + NU*((W):1):1                     -1        2         0
       + NU*((W):3):3                     -1        2         2
       + NU/R*(W):1                       -1        2         0




1 TERMS DROPPED DUE TO DELTA CRITERIA, LEAVING      5

                                        LENGTH VELOCITY    DELTA

U*(W):1                                   -1        2         0
   + W*(W):3                              -1        2         0
     = -(P):3                             -1        2         0
       + NU*((W):1):1                     -1        2         0
       + NU/R*(W):1                       -1        2         0
```

Figure 5. The axial momentum equation in cylindrical coordinates.

(optional) stage TATTR dropped the small term. If the relative orders of the factors were not known, the run could have been set up with separate "deltas" for various factors (TATTR can handle up to sixteen attributes at once). Various linear combinations could then have been tried as criteria for dropping terms. With a modicum of physics guiding the combinations to be used, this approach would have led to the same results.

Again, only a hint of the usefulness of TATTR is possible here. For the problem considered in Reference 6 the 9843 terms (in 18 equations) produced by TENSR were reduced to 529 terms by TATTR. These statistics are somewhat misleading — if the job had been attempted by hand, most of the negligible terms would have been easily spotted. Only the last few hundred would have been difficult, but that's plenty to justify the use of TATTR.

5. OUTLOOK

A.R.A.P. is in the midst of switching computers. Our old one, a DSC META-4, uses, basically, IBM 1130 software. Our new one, a PDP 11/70, uses DEC software. The IBM software is geared to the EBCDIC code for character handling, the DEC software to the ASCII code. For TENSR and TATTR, which mainly manipulate strings of characters, the difference is significant; the conversion will be somewhat more laborious than for ordinary A.R.A.P. programs such as the ones that have been developed with the help of TENSR and TATTR.

However, while converting, we will use the opportunity to make several improvements. The principal ones contemplated are:

(1) TENSR and TATTR will be combined. This will allow greater convenience for the user and greater efficiency of operation.

(2) The specification of different ranges on different dummies will be made possible.

(3) Conversion to physical coordinates, if desired, will be made automatic. In this case, specification of symbol substitutions will only be needed in terms of physical components instead of in terms of contravariant components, covariant components, and combinations thereof.

(4) The operation of the system will be speeded up and thus made more convenient. This will occur partly because of the switch to a faster machine, but also because of the implementation of ways of making it more efficient which have been recognized since the last major revision.

An additional improvement would be to allow substitutions before expansion. For example, given

```
C'IJK = A"L'IJ*B'LK + D'IK,J
```

and another expression where, say, C"JL'L,K occurs, the right-hand side of the former could be substituted in the latter with proper shuffling of the indices. This addition is quite feasible but is more involved than the other changes mentioned. For one thing, a means of handling conflicts in indices, as in the example just given, must be devised.

Another possibility is to provide output using true superscripts and subscripts. This would depend on the availability of suitable output devices and proper software.

Although TENSR and TATTR were developed largely with NASA and Air Force support, they were intended to help in developing programs. In some cases these programs were delivered to the Government; in other cases the programs were run by A.R.A.P. and the results delivered. In any case, TENSR and TATTR were not designed for export. The fact that they can be moved to a different machine is about to be proved, but whether portability will be improved in the process remains to be seen.

The usage of the combined system is not expected to become high in comparison with more general-purpose software, or in comparison with programs that churn out numbers. Therefore, it is doubtful that the effort to make it portable and, in particular, to prepare documentation sufficient to guide a remote user would be worthwhile. Instead, I would suggest that potential users communicate with A.R.A.P. concerning their needs.

As far as I know, no one else has attempted anything like TENSR and TATTR. Coordinate transformations to aid in flow-field computations, with varying degrees of automation, are plentiful. Reference 1, already mentioned, and Reference 7, which proposes conformal transformations in the cross-flow plane for computing supersonic flows, are just two recent examples of many papers describing many approaches.

Formac and similar systems do algebra and calculus automatically — and are much better at it than TENSR and TATTR — but have no provision for incorporating tensor notation. Howard (References 8 and 9, among other papers) has used such systems for doing tensor analysis, but he must write out covariant derivatives in terms of Christoffel symbols (for example, the right side of equation (8) instead of the left), and he uses explicit nested DO loops for each pair of dummy indices. But he recognizes the need for making such processes automatic, not only for conserving manpower, but also for drastically reducing the likelihood of making mistakes.

6. NOTE FOR TENSOR EXPERTS

This section covers some points that may be of interest to those with a relatively thorough knowledge of tensors.

First, TENSR is misnamed. It operates on the symbols supplied to it, as described in Section 3, whether they represent tensors or not. In fact, as soon as it introduces the Christoffel symbols it operates on the various terms separately, and the separate terms are not tensors. In other words, TENSR is not restricted to tensors.

Second, there is no need for the space involved to have a metric. The metric is relevant only if it appears in the input equations, either explicitly or implicitly through the use of ! . This is one reason the Christoffel symbols (more properly interpreted as affine connections when there is no metric) are separate inputs to TENSR. (The other reason is that to be able to determine them from a metric, TENSR would have to be taught to differentiate.)

Third, expressions can be expanded without anything being known about the space except its dimensions and that it has affine connections and, if used, a metric (but not the values of the affine connections or the metric), as is illustrated in Figure 2. Further, it is possible to suppress the introduction of the affine connections (Christoffel symbols), or to suppress the expansion of dummy indices, or both. Thus, there is great flexibility in determining what is to be done.

Fourth, the number of dimensions is limited to ten solely because that is the number of numerical digits available in computer output devices. The user may select any set out of the ten for dummy expansions and for free indices. In particular, for runs involving four-dimensional space-time, the user is free to let whim guide the choice rather than being forced to use 0,1,2,3 or 1,2,3,4 or even being forced to choose between them.

7. BIBLIOGRAPHY

This section contains a few suggestions for those who wish to pursue the subject of tensors. I have made no effort to inspect everything that is available; I report only on some works that I have chanced upon or that have been brought to my attention by others.

The first group considers tensors as tools useful in various branches of applied mathematics; if they consider general relativity, it is as one of several applications. The rest are all devoted to general relativity and develop the theory of tensors for that application only.

Michal, Reference 10, was a pioneer in making tensors accessible to engineers, but Sokolnikoff, Reference 4, is more thorough both in the basics and in the applications. A modern treatment by Aris, Reference 5, is very readable; the applications are restricted, as the title indicates, to fluid mechanics. Budiansky provides a remarkably thorough treatment in less than 50 pages (Reference 11). He includes just a taste of applications by considering mechanical deformations.

It is no wonder that much of what is written on tensors is in works on gravitation. Although they are extremely useful in many other fields, tensors are deemed indispensable in general relativity. Consider first three short works by men whose names have become bywords. Foremost among these is the monograph, Reference 12, by Einstein. Fear not! The parts dealing with mathematics are clear and straightforward. (I'll make no comment on the physics.) Schrödinger, Reference 13, is dry and elegant with barely a hint of what it's all about. By contrast, Dirac is discussing black holes by page 32 of Reference 14.

Two more thorough treatises worth looking into are Anderson, Reference 15, and Weinberg, Reference 16.

Finally, there is the magnificent *Gravitation* (Reference 17) by Misner, Thorne, and Wheeler. This book has exhaustive treatments of a multitude of subjects (the discussion of black holes doesn't start until page 872). It includes, of course, a thorough study of tensors, but also includes a parallel development of the mathematics from the viewpoint of differential geometry, along with discussions of the relationships between these two subjects. In spite of its bulk and comprehensiveness it is lively and accessible.

8. REFERENCES

1. Joe F. Thompson, Samuel P. Shanks, Ray L. Walker, "Numerical Solution of the Navier-Stokes Equations for 2D Hydrofoils," Engineering and Industrial Research Station, Department of Aerophysics and Aerospace Engineering, Mississippi State University, Report Number MMSU-EIRS-ASE-77-5, February 1977.

2. Coleman duP. Donaldson, Dryden Research Lecture: "Calculation of Turbulent Shear Flows for Atmospheric and Vortex Motions," AIAA Journal 10, 1, pp. 4-12, January 1972.

3. Coleman duP. Donaldson, Roger D. Sullivan, "An Invariant Second-Order Closure Model of the Compressible Turbulent Boundary Layer on a Flat Plate," A.R.A.P. Report No. 178, June 1972.

4. I.S. Sokolnikoff, *Tensor Analysis*, John Wiley and Sons, Inc., New York, 1951 (Fourth Printing 1960).

5. Rutherford Aris, *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1962.

6. Roger D. Sullivan, "GYC: A Program To Compute the Turbulent Boundary Layer on a Rotating Cone," A.R.A.P. Working Paper No. 76-2, August 1976.

7. Gino Moretti, "Conformal Mappings for Computations of Steady Three-Dimensional, Supersonic Flows," in *Numerical/Laboratory Computer Methods in Fluid Mechanics*, American Society of Mechanical Engineers, New York, 1976.

8 James C. Howard, "Automatic Problem Formulation Using the Metric Properties of Space," Journal of the Franklin Institute 292, 6, pp. 535-543, December 1971.

9. James C. Howard, "The Transformation of Aerodynamic Stability Derivatives by Symbolic Mathematical Computation," Aeronautical Quarterly, 26, pp. 109-116, May 1975.

10. Aristotle D. Michal, *Matrix and Tensor Calculus*, John Wiley and Sons, Inc., New York, 1947 (Third Printing 1948).

11. Bernard Budiansky, "Tensors," in *Handbook of Applied Mathematics*, edited by Carl E. Pearson, Van Nostrand Reinhold Company, New York, pp. 179-225, 1974.

12. Albert Einstein, *The Meaning of Relativity*, Fifth Edition, Princeton University Press, Princeton, New Jersey, 1956 (Third Princeton Paperback Printing 1972).

13. Erwin Schrödinger, *Space-Time Structure*, Cambridge University Press, Cambridge, 1950 (Fourth Printing 1963).

14. P.A.M. Dirac, *General Theory of Relativity*, John Wiley and Sons, Inc., New York, 1975.

15. James L. Anderson, *Principles of Relativity Physics*, Academic Press, New York, 1967.

16. Steven Weinberg, *Gravitation and Cosmology*, John Wiley and Sons, Inc., New York, 1972.

17. Charles W. Misner, Kip S. Thorne, John Archibald Wheeler, *Gravitation*, W.H. Freeman and Co., San Francisco, 1973.