Introduction to Tensor Calculus

A.V.Smirnov

Contents

Preface

This material offers a short introduction to tensor calculus. It is directed toward students of continuum mechanics and engineers. The emphasis is made on tensor notation and invariant forms. A knowledge of calculus is assumed. A more complete coverage of tensor calculus can be found in [1, 2].

Nomenclature

$A \equiv B$	A is defined as B, or A is equivalent to B
$A_i B_i$	$\equiv \sum_i^3 A_i B_i$. Note: $A_i B_i = A_j B_j$
$\dot{A}$	partial derivative over time: $\frac{\partial A}{\partial t}$
$A_{,i}$	partial derivative over x_i: $\frac{\partial A}{\partial x_i}$
V	control volume
t	time
x_i	i-th component of a coordinate (i=0,1,2), or $x_i \equiv \{x, u, z\}$
RHS	Right-hand-side
LHS	Left-hand-side
PDE	Partial differential equation
..	Continued list of items

There are two aspects of tensors that are of practical and fundamental importance: *tensor notation* and *tensor invariance*. Tensor notation is of great practical importance, since it simplifies handling of complex equation systems. The idea of tensor invariance is of both practical and fundamental importance, since it provides a powerful apparatus to describe non-Euclidean spaces in general and curvilinear coordinate systems in particular.

A definition of a tensor is given in Section 1. Section 2 deals with an important class of Cartesian tensors, and describes the rules of tensor notation. Section 3 provides a brief introduction to general curvilinear coordinates, invariant forms and the rules of covariant differentiation.

1 Coordinates and Tensors

Consider a space of real numbers of dimension n, R^n, and a single real time, t. Continuum properties in this space can be described by arrays of different dimensions, m, such as scalars ($m = 0$), vectors ($m = 1$), matrices ($m = 2$), and general multi-dimensional arrays. In this space we shall introduce a *coordinate system*, $\{x^i\}_{i=1..n}$, as a way of assigning n real numbers[1] for every point of space There can be a variety of possible coordinate systems. A general *transformation rule* between the coordinate systems is

$$\tilde{x}^i = \tilde{x}^i(x^1..x^n) \tag{1}$$

Consider a small displacement dx^i. Then it can be transformed from coordinate system x^i to a new coordinate system $\tilde{x}^i$ using the partial differentiation rules applied to (1):

$$d\tilde{x}^i = \frac{\partial \tilde{x}^i}{\partial x^j} dx^j \tag{2}$$

This transformation rule[2] can be generalized to a set of vectors that we shall call *contravariant vectors:*

$$\tilde{A}^i = \frac{\partial \tilde{x}^i}{\partial x^j} A^j \tag{3}$$

[1] Super-indexes denote components of a vector ($i = 1..n$) and not the power exponent, for the reason explained later (Definition 1.1)

[2] The repeated indexes imply summation (See. Proposition 21)

That is, a contravariant vector is defined as a vector which transforms to a new coordinate system according to (3). We can also introduce the *transformation matrix* as:

$$a^i_j \equiv \frac{\partial \tilde{x}^i}{\partial x^j} \tag{4}$$

With which (3) can be rewritten as:

$$A^i = a^i_j A^j \tag{5}$$

Transformation rule (3) will not apply to all the vectors in our space. For example, a partial derivative $\partial/\partial x_i$ will transform as:

$$\frac{\partial}{\partial \tilde{x}^i} = \frac{\partial}{\partial \tilde{x}^i}\frac{\partial x^j}{\partial x^j} = \frac{\partial x^j}{\partial \tilde{x}^i}\frac{\partial}{\partial x^j} \tag{6}$$

that is, the transformation coefficients are the other way up compared to (2). Now we can generalize this transformation rule, so that each vector that transforms according to (6) will be called a *Covariant vector:*

$$\tilde{A}_i = \frac{\partial x^j}{\partial \tilde{x}^i} A_j \tag{7}$$

This provides the reason for using lower and upper indexes in a general tensor notation.

Definition 1.1 Tensor

Tensor of order m is a set of n^m numbers identified by m integer indexes. For example, a 3rd order tensor A can be denoted as A_{ijk} and an m-order tensor can be denoted as $A_{i_1..i_m}$. Each index of a tensor changes between 1 and n. For example, in a 3-dimensional space (n=3) a second order tensor will be represented by $3^2 = 9$ components.

Each index of a tensor should comply to one of the two transformation rules: (3) or (7). An index that complies to the rule (7) is called a **covariant index** *and is denoted as a sub-index, and an index complying to the transformation rule (3) is called a* **contravariant index** *and is denoted as a super-index.*

Each index of a tensor can be covariant or a contravariant, thus tensor A^k_{ij} is a 2-covariant, 1-contravariant tensor of third order.

From this relation and the independence of coordinates (9) it follows that $a^i_j b^j_k = b^i_j a^j_k = \delta_{ik}$, namely:

$$a^i_j b^j_k = \frac{\partial \tilde{x}^i}{\partial x^j}\frac{\partial x^j}{\partial \tilde{x}^k}$$
$$= \frac{\partial x^j}{\partial x^j}\frac{\partial \tilde{x}^i}{\partial \tilde{x}^k} = \frac{\partial \tilde{x}^i}{\partial \tilde{x}^k} = \delta_{ik} \tag{13}$$

2 Cartesian Tensors

Cartesian tensors are a sub-set of general tensors for which the transformation matrix (4) satisfies the following relation:

$$a^k_i a^k_j = \frac{\partial \tilde{x}^k}{\partial x^i}\frac{\partial \tilde{x}^k}{\partial x^j} = \delta_{ij} \tag{14}$$

For Cartesian tensors we have

$$\frac{\partial \tilde{x}^i}{\partial x^k} = \frac{\partial x^k}{\partial \tilde{x}^i} \tag{15}$$

(see Problem 4.3), which means that both (5) and (6) are transformed with the same matrix a^i_k. This in turn means that the difference between the covariant and contravariant indexes vanishes for the Cartesian tensors. Considering this we shall only use the sub-indexes whenever we deal with Cartesian tensors.

2.1 Tensor Notation

Tensor notation simplifies writing complex equations involving multi-dimensional objects. This notation is based on a set of *tensor rules.* The rules introduced in this section represent a complete set of rules for Cartesian tensors and will be extended in the case of general tensors (Sec.3). The importance of tensor rules is given by the following general remark:

Remark 2.1 Tensor rules *Tensor rules guarantee that if an expression follows these rules it represents a tensor according to Definition 1.1.*

Tensors are usually functions of space and time:

$$A_{i_1..i_m} = A_{i_1..i_m}(x^1..x^n, t)$$

which defines a tensor field, i.e. for every point x^i and time t there are a set of m^n nubers $A_{i_1..i_m}$.

Remark 1.2 Tensor character of coordinate vectors

Note, that the coordinates x^i are not tensors, since generally, they are not transformed as (5). Transformation law for the coordinates is actually given by (1). Nevertheless, we shall use the upper (contravariant) indexes for the coordinates.

Definition 1.3 Kronecker delta tensor

Second order delta tensor, δ_{ij} is defined as

$$\begin{aligned} i = j &\Rightarrow \delta_{ij} = 1 \\ i \neq j &\Rightarrow \delta_{ij} = 0 \end{aligned} \tag{8}$$

From this definition and since coordinates x^i are independent of each other it follows that:

$$\frac{\partial x^i}{\partial x^j} = \delta_{ij} \tag{9}$$

Corollary 1.4 Delta product

From the definition (1.3) and the summation convention (21), follows that

$$\delta_{ij} A_j = A_i \tag{10}$$

Assume that there exists the transformation inverse to (5), which we call b^i_j:

$$dx^i = b^i_j d\tilde{x}^j \tag{11}$$

Then by analogy to (4) b^i_j can be defined as:

$$b^i_j = \frac{\partial x^i}{\partial \tilde{x}^j} \tag{12}$$

Thus, following tensor rules, one can build tensor expressions that will preserve tensor properties of coordinate transformations (Definition 1.1) and coordinate invariance (Section 3).

Tensor rules are based on the following definitions and propositions.

Definition 2.2 Tensor terms

A tensor term is a product of tensors.

For example:

$$A_{ijk}B_{jk}C_{pq}E_qF_p \tag{16}$$

Definition 2.3 Tensor expression

Tensor expression is a sum of tensor terms. For example:

$$A_{ijk}B_{jk} + C_iD_{pq}E_qF_p \tag{17}$$

Generally the terms in the expression may come with plus or minus sign.

Proposition 2.4 Allowed operations

The only allowed algebraic operations in tensor expressions are the addition, subtraction and multiplication. Divisions are only allowed for constants, like $1/C$*. If a tensor index appears in a denominator, such term should be redefined, so as not to have tensor indexes in a denominator. For example,* $1/A_i$ *should be redefined as:* $B_i \equiv 1/A_i$*.*

Definition 2.5 Tensor equality

Tensor equality is an equality of two tensor expressions.

For example:

$$A_{ij}B_j = C_{ikp}D_kE_p + E_jC_{jki}B_k \tag{18}$$

Definition 2.6 Free indexes

A free index is any index that occurs only once in a tensor term. For example, index i is a free index in the term (16).

Proposition 2.7 Free index restriction

Every term in a tensor equality should have the same set of free indexes.

For example, if index i is a free index in any term of tensor equality, such as (18), it should be the free index in all other terms. For example

$$A_{ij}B_j = C_jD_j$$

is not a valid tensor equality since index i is a free index in the term on the RHS but not in the LHS.

Definition 2.8 Rank of a term

A rank of a tensor term is equal to the number of its free indexes.

For example, the rank of the term $A_{ijk}B_jC_k$ is equal to 1.

It follows from (2.7) that ranks of all the terms in a valid tensor expression should be the same. Note, that the difference between the order and the rank is that the order is equal to the number of indexes of a tensor, and the rank is equal to the number of free indexes in a tensor term.

Proposition 2.9 Renaming of free indexes

Any free index in a tensor expression can be named by any symbol as long as this symbol does not already occur in the tensor expression.

For example, the equality

$$A_{ij}B_j = C_iD_jE_j \tag{19}$$

is equivalent to

$$A_{kj}B_j = C_kD_jE_j \tag{20}$$

Here we replaced the free index i with k.

Definition 2.10 Dummy indexes

A dummy index is any index that occurs twice in a tensor term.

For example, indexes j, k, p, q in (16) are dummy indexes.

Proposition 2.11 Summation rule

Any dummy index implies summation, i.e.

$$A_i B_i = \sum_i^n A_i B_i \tag{21}$$

Proposition 2.12 Summation rule exception *If there should be no summation over the repeated indices, it can be indicated by enclosing such indices in parentheses.*

For example, expression:

$$C_{(i)} A_{(i)} B_j = D_{ij}$$

does not imply summation over i.

Corollary 2.13 Scalar product

A scalar product notation from vector algebra: $(A \cdot B)$ is expressed in tensor notation as $A_i B_i$.

The scalar product operation is also called a *contraction of indexes*.

Proposition 2.14 Dummy index restriction

No index can occur more than twice in any tensor term.

Remark 2.15 Repeated indexes

In case if an index occurs more than twice in a term this term should be redefined so as not to contain more than two occurrences of the same index. For example, term $A_{ik} B_{jk} C_k$ should be rewritten as $A_{ik} D_{jk}$, where D_{jk} is defined as $D_{jk} \equiv B_{j(k)} C_{(k)}$ with no summation over k in the last term.

Proposition 2.16 **Renaming of dummy indexes**

Any dummy index in a tensor term can be renamed to any symbol as long as this symbol does not already occur in this term.

For example, term A_iB_i is equivalent to A_jB_j, and so are terms $A_{ijk}B_jC_k$ and $A_{ipq}B_pC_q$.

Remark 2.17 **Renaming rules**

Note that while the dummy index renaming rule (2.16) is applied to each tensor term separately, the free index naming rule (2.9) should apply to the whole tensor expression. For example, the equality (19) above

$$A_{ij}B_j = C_iD_jE_j$$

can also be rewritten as

$$A_{kp}B_p = C_kD_jE_j \tag{22}$$

without changing its meaning.

(See Problem 4.1).

Definition 2.18 **Permutation tensor**

The components of a third order permutation tensor ε_{ijk} are defined to be equal to 0 when any index is equal to any other index; equal to 1 when the set of indexes can be obtained by cyclic permutation of 123; and -1 when the indexes can be obtained by cyclic permutation from 132. In a mathematical language it can be expressed as:

$$\begin{aligned} i = j \cup i = k \cup j = k \;\Rightarrow\; \varepsilon_{ijk} = 0 \\ ijk \in PG(123) \;\Rightarrow\; \varepsilon_{ijk} = 1 \\ ijk \in PG(132) \;\Rightarrow\; \varepsilon_{ijk} = -1 \end{aligned} \tag{23}$$

where $PG(abc)$ is a permutation group of a triple of indexes abc, i.e. $PG(abc) = \{abc, bca, cab\}$. For example, the permutation group of 123 will consist of three combinations: 123, 231 and 312, and the permutation group of 123 consists of 132, 321 and 213.

Corollary 2.19 Permutation of the permutation tensor indexes

From the definition of the permutation tensor it follows that the permutation of any of its two indexes changes its sign:

$$\varepsilon_{ijk} = -\varepsilon_{ikj} \tag{24}$$

A tensor with this property is called *skew-symmetric*.

Corollary 2.20 Vector product

A vector product (cross-product) of two vectors in vector notation is expressed as

$$\vec{A} = \vec{B} \times \vec{C} \tag{25}$$

which in tensor notation can be expressed as

$$A_i = \varepsilon_{ijk} B_j C_k \tag{26}$$

Remark 2.21 Cross product

Tensor expression (26) is more accurate than its vector counterpart (25), since it explicitly shows how to compute each component of a vector product.

Theorem 2.22 Symmetric identity

If A_{ij} is a symmetric tensor, then the following identity is true:

$$\varepsilon_{ijk} A_{jk} = 0 \tag{27}$$

Proof:

From the symmetry of A_{ij} we have:

$$\varepsilon_{ijk} A_{jk} = \varepsilon_{ijk} A_{kj} \tag{28}$$

Let's rename index j into k and k into j in the RHS of this expression, according to rule (2.16):

$$\varepsilon_{ijk} A_{kj} = \varepsilon_{ikj} A_{jk}$$

Using (24) we finally obtain:

$$\varepsilon_{ikj}A_{jk} = -\varepsilon_{ijk}A_{jk}$$

Comparing the RHS of this expression to the LHS of (28) we have:

$$\varepsilon_{ijk}A_{jk} = -\varepsilon_{ijk}A_{jk}$$

from which we conclude that (27) is true.

Theorem 2.23 Tensor identity

The following tensor identity is true:

$$\varepsilon_{ijk}\varepsilon_{ipq} = \delta_{jp}\delta_{kq} - \delta_{jq}\delta_{kp} \qquad (29)$$

Proof

This identity can be proved by examining the components of equality (29) component-by-component.

Corollary 2.24 Vector identity

Using the tensor identity (29) it is possible to prove the following important vector identity:

$$\vec{A} \times (\vec{B} \times \vec{C}) = \vec{B}(\vec{A} \cdot \vec{C}) - \vec{C}(\vec{A} \cdot \vec{B}) \qquad (30)$$

See Problem 4.4.

2.2 Tensor Derivatives

For Cartesian tensors derivatives introduce the following notation.

Definition 2.25 Time derivative of a tensor

A partial derivative of a tensor over time is designated as

$$\dot{A} \equiv \frac{\partial A}{\partial t}$$

Definition 2.26 **Spatial derivative of a tensor**

A partial derivative of a tensor A over one or its spacial components is denoted as $A_{,i}$:

$$A_{,i} \equiv \frac{\partial A}{\partial x_i} \tag{31}$$

that is, the index of the spatial component that the derivation is done over is delimited by a comma (',') from other indexes. For example, $A_{ij,k}$ is a derivative of a second order tensor A_{ij}.

Definition 2.27 **Nabla**

Nabla operator acting on a tensor A is defined as

$$\nabla_i A \equiv A_{,i} \tag{32}$$

Even though the notation in (31) is sufficient to define the derivative, In some instances it is convenient to introduce the nabla operator as defined above.

Remark 2.28 **Tensor derivative**

In a more general context of non-Cartesian tensors the coordinate independent derivative will have a different form from (31). See the chapter on covariant differentiation in [1].

Remark 2.29 **Rank of a tensor derivative**

The derivative of a zero order tensor (scalar) as given by (31) forms a first order tensor (vector). Generally, the derivative of an m-order tensor forms an $m+1$ order tensor. However, if the derivation index is a dummy index, then the rank of the derivative will be lower than that of the original tensor. For example, the rank of the derivative $A_{ij,j}$ is one, since there is only one free index in this term.

Remark 2.30 **Gradient**

Expression (31) represents a gradient, which in a vector notation is ∇A:

$$\nabla A \longrightarrow A_{,i}$$

Corollary 2.31 Derivative of a coordinate

From (9) it follows that:

$$x_{i,j} = \delta_{ij} \tag{33}$$

In particular, the following identity is true:

$$x_{i,i} = x_{1,1} + x_{2,2} + x_{3,3} = 1 + 1 + 1 = 3 \tag{34}$$

Remark 2.32 Divergence operator

A divergence operator in a vector notation is represented in a tensor notation as $A_{i,i}$:

$$(\nabla \cdot \vec{A}) \longrightarrow A_{i,i}$$

Remark 2.33 Laplace operator

The Laplace operator in vector notation is represented in tensor notation as $A_{,ii}$:

$$\Delta A \longrightarrow A_{,ii}$$

Remark 2.34 Tensor notation

Examples (2.30), (2.32) and (2.33) clearly show that tensor notation is more concise and accurate than vector notation, since it explicitly shows how each component should be computed. It is also more general since it covers cases that don't have representation in vector notation, for example: $A_{ik,kj}$.

3 Curvilinear coordinates

In this section [3] we introduce the idea of *tensor invariance* and introduce the rules for constructing *invariant forms*.

[3] In this section we reinstall the difference between covariant and contravariant indexes.

3.1 Tensor invariance

The distance between the material points in a Cartesian coordinate system is computed as $dl^2 = dx_i dx_i$. The *metric tensor*, g_{ij} is introduced to generalize the notion of distance (39) to curvilinear coordinates.

Definition 3.1 Metric Tensor

The distance element in curvilinear coordinate system is computed as:

$$dl^2 = g_{ij} dx^i dx^j \tag{35}$$

where g_{ij} *is called the* metric tensor*.*

Thus, if we know the metric tensor in a given curvilinear coordinate system then the distance element is computed by (35). The metric tensor is defined as a tensor since we need to preserve the *invariance* of distance in different coordinate systems, that is, the distance should be independent of the coordinate system, thus:

$$dl^2 = g_{ij} dx^i dx^j = \tilde{g}_{ij} d\tilde{x}^i d\tilde{x}^j \tag{36}$$

The metric tensor is symmetric, which can be shown by rewriting (35) as follows:

$$g_{ij} dx^i dx^j = g_{ij} dx^j dx^i = g_{ji} dx^i dx^j$$

where we first swapped places of dx^i and dx^j, and then renamed index i into j and j into i. We can rewrite the equality above as:

$$g_{ij} dx^i dx^j - g_{ji} dx^i dx^j = (g_{ij} - g_{ji}) dx^i dx^j = 0$$

Since the equality above should hold for any $dx^i dx^j$, we get:

$$g_{ij} = g_{ji} \tag{37}$$

The metric tensor is also called the *fundamental tensor.* The inverse of the metric tensor is also called the *conjugate metric tensor*, g^{ij}, which satisfies the relation:

$$g^{ik}g_{kj} = \delta_{ij} \tag{38}$$

Let x^i be a Cartesian coordinate system, and $\tilde{x}^j$ - the new curvilinear coordinate system. Both systems are related by transformation rules (5) and (11). Then from (36) we get:

$$dl^2 = dx^i dx^i = \frac{\partial x^i}{\partial \tilde{x}^j} d\tilde{x}^j \frac{\partial x^i}{\partial \tilde{x}^k} d\tilde{x}^k = \frac{\partial x^i}{\partial \tilde{x}^j} \frac{\partial x^i}{\partial \tilde{x}^k} d\tilde{x}^j d\tilde{x}^k \tag{39}$$

When we transform from a Cartesian to curvilinear coordinates the metric tensor in curvilinear coordinate system, $\tilde{g}_{ij}$ can be determined by comparing relations (39) and (35):

$$\tilde{g}_{ij} = \frac{\partial x^k}{\partial \tilde{x}^i} \frac{\partial x^k}{\partial \tilde{x}^j} \tag{40}$$

Using (38) we can also find its inverse as:

$$\tilde{g}^{ij} = \frac{\partial \tilde{x}^i}{\partial x^k} \frac{\partial \tilde{x}^j}{\partial x^k} \tag{41}$$

Using these expression one can compute g_{ij} and g^{ij} in various curvilinear coordinate systems (see Problem 4.6).

Definition 3.2 Conjugate tensors

For each index of a tensor we introduce the conjugate tensor where this index is transfered to its counterpart (covariant/contravariant) using the relations:

$$A^i = g^{ij} A_j \tag{42}$$
$$A_i = g_{ij} A^j \tag{43}$$

Conjugate tensor is also called the *associate tensor*. Relations (42), (43) are also called as operations of *raising/lowering of indexes*.

Remark 3.3 Tensor invariance

Since the transformation rules defined by (1.1) have a simple multiplicative character, any tensor expression should retain it's original form under transformation into a new coordinate system. Thus if an expression is given in a tensor form it will be invariant *under coordinate transformations.*

Not all the expressions constructed from tensor terms in curvilinear coordinates will be tensors themselves. For example, if vectors A_i and B_i are tensors, then $A_i B_i$ is not generally a tensor[4]. However, if we consider the same operation on a contravariant tensor A^i and a covariant tenso B_i then the product will form an invariant:

$$\bar{A}^i \bar{B}_i = A^i B_i \tag{44}$$

Thus in curvilinear coordinates we have to refine the definition of the scalar product (Corollary 2.13) or the index contraction operation to make it invariant (Problem 4.12).

Definition 3.4 Invariant Scalar Product

The invariant form *of the scalar product between two covariant vectors A_i and B_i is $g^{ij} A_i B_j$. Similarly, the invariant form of a scalar product between two contravariant vectors A^i and B^i is $g_{ij} A^i B^j$, where g_{ij} is the metric tensor (40) and g^{ij} is its conjugate (38).*

Corollary 3.5 Two forms of a scalar product

According to (42), (43) the scalar product can be represented by two invariant forms: $A^i B_i$ and $A_i B^i$. It can be easily shown that these two forms have the same values (see Problem 4.12).

Corollary 3.6 Rules of invariant expressions

To build invariant tensor expressions we add two more rules to Cartesian tensor rules outlined in Section 2.1:

1. *Each free index should keep its vertical position in every term, i.e. if the index is covariant in one term it should be covariant in every other term, and vise versa.*
2. *Every pair of dummy indexes should be complementary, that is one should be covariant, and another contravariant.*

For example, a Cartesian formulation of a *momentum equation* for an incompressible viscous fluid is

$$\dot{u}_i + u_k u_{i,k} = -\frac{P_{,i}}{\rho} + \nu \tau_{ik,k}$$

[4] For Cartesian tensors any product of tensors will always be a tensor, but this is not so for general tensors

The invariant form of this equation is:

$$\dot{u}_i + u^k u_{i,k} = -\frac{P_{,i}}{\rho} + \nu \tau^k_{i,k} \tag{45}$$

where the rising of indexes was done using relation (42): $u^k = g^{kj} u_j$, and $\tau^k_i = g^{kj} \tau_{ij}$.

3.2 Covariant differentiation

A simple scalar value, S, is invariant under coordinate transformations. A partial derivative of an invariant is a first order covariant tensor (vector):

$$A^i = S_{,i} = \frac{\partial S}{\partial x^i}$$

However, a partial derivative of a tensor of the order one and greater is not generally an invariant under coordinate transformations of type (7) and (3).

In curvilinear coordinate system we should use more complex differentiation rules to preserve the invariance of the derivative. These rules are called the rules of *covariant differentiation* and they guarantee that the derivative itself is a tensor. According to these rules the derivatives for covariant and contravariant indices will be slightly different. They are expressed as follows:

$$A_{i,j} \equiv \frac{\partial A_i}{\partial x^j} - \left\{ {k \atop ij} \right\} A_k \tag{46}$$

$$A^i_{,j} \equiv \frac{\partial A^i}{\partial x^j} + \left\{ {i \atop kj} \right\} A^k \tag{47}$$

where the contstruct $\left\{ {k \atop ij} \right\}$ is defined as

$$\left\{ {k \atop ij} \right\} = \frac{1}{2} g^{kl} \left(\frac{\partial g_{il}}{\partial x^j} + \frac{\partial g_{jl}}{\partial x^i} - \frac{\partial g_{ij}}{\partial x^l} \right)$$

and is also known in tensor calculus as *Christoffel's symbol* of the second kind [1]. Tensor g^{ij} represents the inverse of the metric tensor g_{ij} (38). As can be seen differentiation of a single component of a vector will involve all other components of this vector.

In differentiating higher order tensors each index should be treated independently. Thus differentiating a second order tensor, A^{ij}, should be performed as:

$$A_{ij,k} = \frac{\partial A_{ij}}{\partial x^k} - \{{}^{m}_{ik}\}A_{mj} - \{{}^{m}_{jk}\}A_{im}$$

and as can be seen also involves all the components of this tensor. Likewise for the contravariant second order tensor A^{ij} we have:

$$A^{ij}_{,k} = \frac{\partial A^{ij}}{\partial x^k} + \{{}^{i}_{mk}\}A^{mj} + \{{}^{j}_{mk}\}A^{im} \tag{48}$$

And for a general n-covariant, m-contravariant tensor we have:

$$\begin{aligned} A^{j_1..j_m}_{i_1..i_n,p} &= \frac{\partial}{\partial x^p} A^{j_1..j_m}_{i_1..i_n,k} \\ +\{{}^{j_1}_{qp}\}A^{qj_2..j_m}_{i_1..i_n} &+ \cdots + \{{}^{j_m}_{qp}\}A^{j_1..j_{m-1}q}_{i_1..i_n} \\ +\{{}^{q}_{i_1 p}\}A^{j_1..j_m}_{qi_2..i_n} &+ \cdots + \{{}^{q}_{i_n p}\}A^{j_1..j_m}_{i_1..i_{n-1}q} \end{aligned} \tag{49}$$

Despite their seeming complexity, the relations of covariant differentiation can be easily implemented algorithmically and used in numerical solutions on arbitrary curved computational grids (Problem 4.8).

Remark 3.7 Rules of invariant expressions

As was pointed out in Corollary 3.6, the rules to build invariant expressions involve raising or lowering indexes (42), (43). However, since we did not introduce the notation for contravariant derivative, the only way to raise the index of a covariant derivative, say $A_{,i}$, it to use the relation (42) directly, that is: $g^{ij}A_{,j}$.

For example, we can re-formulate the momentum equation (45) in terms of contravariant free index i as:

$$\dot{u}^i + u^k u^i_{,k} = -\frac{g^{ik}P_{,k}}{\rho} + \nu\tau^{ik}_{,k} \tag{50}$$

where the index of the pressure term was raised by means of (42).

Using the invariance of the scalar product one can construct two important differential operators in curvilinear coordinates: *divergence* of a vector $divA \equiv A^i_{,i}$ (51) and *Laplacian*, $\Delta A \equiv g^{ik}A_{,ki}$ (55).

Definition 3.8 Divergence

Divergence of a vector is defined as $A^i_{,i}$*:*

$$divA \equiv A^i_{,i} \tag{51}$$

From this definition and the rule of covariant differentiation (47) we have:

$$A^i_{,i} = \frac{\partial A^i}{\partial x^i} + \{^{\,i}_{ki}\}A^k \tag{52}$$

this can be shown [2] to be equal to:

$$\begin{aligned} A^i_{,i} &= \frac{\partial A^i}{\partial x^i} + \left(\frac{1}{\sqrt{g}}\frac{\partial}{\partial x^i}\sqrt{g}\right)A^i \\ &= \frac{1}{\sqrt{g}}\frac{\partial}{\partial x^i}\left(\sqrt{g}A^i\right) \end{aligned} \tag{53}$$

where g is the determinant of the metric tensor g_{ij}.

The divergence of a covariant vector A_i is defined as a divergence of its conjugate contravariant tensor (42):

$$A^i_{,i} = g^{ij}A_{j,i} \tag{54}$$

Definition 3.9 Laplacian

A Laplace operator or a Laplacian of a scalar A is defined as

$$\Delta A \equiv g^{ik}A_{,ki} \tag{55}$$

The definitions (3.8), (3.9) of differential operators are invariant under coordinate transformations. They can be programmed using a symbolic manipulation packages and used to derive expressions in different curvilinear coordinate systems (Problem 4.9).

3.3 Orthogonal coordinates

3.3.1 Unit vectors and stretching factors

The coordinate system is *orthogonal* if the tangential vectors to coordinate lines are orthogonal at every point.

Consider three unit vectors, a^i, b^i, c^i, each directed along one of the coordinate axis (tangential unit vectors), that is:

$$a^i = \{a^1, 0, 0\} \tag{56}$$
$$b^i = \{0, b^2, 0\} \tag{57}$$
$$c^i = \{0, 0, c^3\} \tag{58}$$

The condition of orthogonality means that the scalar product between any two of these unit vectors should be zero. According to the definition of a scalar product (Definition 3.4) it should be written in form (44), that is, a scalar product between vectors a_i and b_i can be written as: $a^i b_i$ or $a_i b^i$. Let's use the first form for definiteness. Then, applying the operation of rising indexes (42), we can express the scalar product in contravariant components only:

$$\begin{aligned} 0 = a^i b_i = g_{ij} a^i b^j = \\ g_{11} a^1 0 + g_{12} a^1 b^2 + g_{13} 00 \\ g_{21} a^2 b^1 + g_{22} 0 b^2 + g_{23} 00 \\ g_{31} a^3 0 + g_{32} 0 b^2 + g_{33} 00 \\ = (g_{12} + g_{21}) a^1 b^2 = 2 g_{12} a^1 b^2 = 0 \end{aligned} \tag{59}$$

where we used the symmetry of g_{ij}, (37). Since vectors a^1 and b^2 were chosen to be non-zero, we have: $g_{12} = 0$. Applying the same reasoning for scalar products of other vectors, we conclude that the metric tensor has only diagonal components non-zero[5]:

$$g_{ij} = \delta_{ij} g_{(ii)} \tag{60}$$

Let's introduce stretching factors, h_i, as the square roots of these diagonal components of g_{ij}:

$$h_1 \equiv (g_{11})^{1/2}; \quad h_2 \equiv (g_{22})^{1/2}; \quad h_3 \equiv (g_{33})^{1/2}; \tag{61}$$

Now, consider the scalar product of each of the unit vectors (56)-(58) with itself. Since all vectors are unit, the scalar product of each with itself should be one:

[5] We use parenthesis to preclude summation (Proposition 2.12)

$$a^i a_i = b^i b_i = c^i c_i = 1$$

Or, expressed in contravariant components only the condition of unity is:

$$g_{ij} a^i a^j = g_{ij} b^i b^j = g_{ij} c^i c^j = 1$$

Now, consider the first term above and substitute the components of a from (56). The only non-zero term will be:

$$g_{11} a^1 a^1 = (h_1)^2 (a^1)^2 = 1$$

and consequently:

$$a^1 = \pm \frac{1}{h_1} \tag{62}$$

where the negative solution identifies a vector directed into the opposite direction, and we can neglect it for definiteness. Applying the same reasoning for each of the tree unit vectors a_i, b_i, c_i, we can rewrite (56), (57) and (58) as:

$$a^i = \{\frac{1}{h_1}, 0, 0\} \tag{63}$$

$$b^i = \{0, \frac{1}{h_2}, 0\} \tag{64}$$

$$c^i = \{0, 0, \frac{1}{h_3}\} \tag{65}$$

which means that the components of unit vectors in a curved space should be scaled with coefficients h_i. It follows from this that the expression for the element of length in curvilinear coordinates, (35), can be written as:

$$dl^2 = g_{ij} d\tilde{x}^i d\tilde{x}^j = h_i^2 (d\tilde{x}^i)^2 \tag{66}$$

Similarly, we introduce the h^i coefficients for the conjugate metric tensor (38):

$$g^{ij} = \delta_{ij} (h^{(i)})^2 \tag{67}$$

Combining the latter with (38), we obtain: $\delta_{ij} h_{(i)} h^{(i)} = \delta_{ij}$, from which it follows that

$$h_{(i)} = 1/h^{(i)} \tag{68}$$

3.3.2 Physical components of tensors

Consider a direction in space determined by a unit vector e_i. Then the *physical component* of a vector A_i in the direction e_i is given by a scalar product between A_i and e_i (Definition 3.4), namely:

$$A(e) = g^{ij} A_i e_j$$

According to Corollary 3.5 the above can also be rewritten as:

$$A(e) = A_i e^i = A^i e_i \tag{69}$$

Suppose the unit vector is directed along one of the axis: $e^i = \{e^1, 0, 0\}$. From (63) it follows that:

$$e^1 = 1/h_1$$

where h_1 is defined by (61). Thus according to (69) the physical component of vector A_i in direction 1 in orthogonal coordinate system is equal to:

$$A(1) = A_1/h_1$$

or, repeating the argument for other components, we have for the physical components of a covariant vector:

$$A_1/h_1,\;\; A_2/h_2,\;\; A_3/h_3 \tag{70}$$

Following the same reasoning, for the contravariant vector A^i, we have:

$$h_1 A^1,\;\; h_2 A^2\;\; h_3 A^3$$

General rules of covariant differentiation introduced in (Sec.3.2) simplify considerably in orthogonal coordinate systems. In particular, we can define the *nabla* operator by the physical components of a covariant vector composed of partial differentials:

$$\nabla_i = \frac{1}{h_{(i)}} \frac{\partial}{\partial x^i} \tag{71}$$

where the parentheses indicate that there's no summation with respect to index i.

In orthogonal coordinate system the general expressions for divergence (53) and Laplacian (55)) operators can be expressed in terms of stretching factors only [3]:

$$\begin{aligned} A^i_{,i} &= \frac{1}{H} \frac{\partial}{\partial x_i} \left(\frac{H}{h_{(i)}} A_i \right) \\ \Delta A &= \frac{1}{H} \frac{\partial}{\partial x_i} \left(\frac{H}{h_{(i)}} \frac{A}{\partial x_i} \right) \\ H &\equiv \prod_{i=1}^{n} h_i \end{aligned} \tag{72}$$

Important examples of orthogonal coordinate systems are spherical and cylindrical coordinate systems. Consider the example of a cylindrical coordinate system: $x_i = \{x_1, x_2, x_3\}$ and $\tilde{x}_i = \{r, \theta, l\}$:

$$\begin{aligned} x_1 &= r \cos\theta \\ x_2 &= r \sin\theta \\ x_3 &= l \end{aligned}$$

According to (40) only few components of the metric tensor will survive (Problem 4.5). Then we can compute nabla, divergence and Laplacian operators according to (71), (52) and (55), or using simplified relations (72)-(73):

$$\begin{aligned} \nabla &= \left(\frac{\partial}{\partial r}, \frac{1}{r} \frac{\partial}{\partial \theta}, \frac{\partial}{\partial z} \right) \\ divA &= \frac{\partial A_1}{\partial \tilde{x}^1} + \frac{1}{\tilde{x}^1} \frac{\partial A_2}{\partial \tilde{x}^2} + \frac{\partial A_3}{\partial \tilde{x}^3} + \frac{1}{\tilde{x}^1} A_1 \\ &= \frac{\partial A_r}{\partial r} + \frac{1}{r} \frac{\partial A_\theta}{\partial \theta} + \frac{\partial A_z}{\partial z} + \frac{1}{r} A_r \end{aligned}$$

Note, that instead of using the contravariant components as implied by the general definition of the divergence operator (51) we are using the covariant components as dictated by relation (70). The expression of the Laplacian becomes:

$$
\begin{aligned}
\Delta A &= \frac{\partial^2 A}{(\partial \tilde{x}_1)^2} + \frac{1}{\tilde{x}_1^2}\frac{\partial^2 A}{(\partial \tilde{x}_2)^2} + \frac{\partial^2 A}{(\partial \tilde{x}_3)^2} + \frac{1}{\tilde{x}_1}\frac{\partial A}{\partial \tilde{x}_1} \\
&= \frac{\partial^2 A}{(\partial r)^2} + \frac{1}{r^2}\frac{\partial^2 A}{(\partial \theta)^2} + \frac{\partial^2 A}{(\partial z)^2} + \frac{1}{r}\frac{\partial A}{\partial r}
\end{aligned}
$$

(see Problems 4.9,4.10).

The advantages of the tensor approach are that it can be used for any type of curvilinear coordinate transformations, not necessarily analytically defined, like cylindrical (85) or spherical. Another advantage is that the equations above can be easily produced automatically using symbolic manipulation packages, such as Mathematica (wolfram.com) (Problems 4.6,4.7,4.9). For further reading see [1, 2].

4 Problems

Problem 4.1 **Check tensor expressions for consistency**

Check if the following Cartesian tensor expressions violate tensor rules:

$$A_{ij}B_{jk} + B_{pq}C_qD_k = 0$$

$$E_{pqi}F_{kj}C_{pk} + B_{pj}D_{jq}G_q = F_{kp}$$

$$E_{ijk}A_jB_k - D_{ij}A_iB_j = F_{ij}G_{jk}H_{kj}$$

Problem 4.2 **Construct tensor expression**

Construct a valid Cartesian tensor expression, consisting of three terms, each including some of the four tensors: $A_{ijk}, B_{ij}, C_i, D_{ij}$. Term 1 should include tensors A, B, C only, term 2 tensor B, C, D and term 3 tensors C, D, A. The expression should have 2 free indexes, which should always come first among the indexes of a tensor. The free indexes should be at A and B in the first term, at B and C in the second term and C and D in the last term. How many different tensor expressions can be constructed?

Problem 4.3 **Cartesian identity**

Prove identity (15)

Problem 4.4 **Vector identity**

Using tensor identity (29):

$$\varepsilon_{ijk}\varepsilon_{ipq} = \delta_{jp}\delta_{kq} - \delta_{jq}\delta_{kp}$$

prove vector identity (30):

$$\vec{A} \times (\vec{B} \times \vec{C}) = \vec{B}(\vec{A} \cdot \vec{C}) - \vec{C}(\vec{A} \cdot \vec{B})$$

Problem 4.5 **Metric tensor in cylindrical coordinates**

Cylindrical coordinate system $y_i = \{r, \theta, l\}$ (85) is given by the following transformation rules to a Cartesian coordinate system, $x_i = \{x, y, z\}$:

$$x = r\cos\theta$$
$$y = r\sin\theta$$
$$z = l$$

Obtain the components of the metric tensor (40) g_{ij} and its inverse g^{ij} (38) in cylindrical coordinates.

Problem 4.6 Metric tensor in curvilinear coordinates

Using Mathematica Compute the metric tensor, g, (40) and its conjugate, $\hat{g}$, (38) in spherical coordinate system (r, ϕ, θ):

$$x = r\sin\theta\cos\phi$$
$$y = r\sin\theta\sin\phi$$
$$z = r\cos\theta \tag{73}$$

Problem 4.7 Christoffel's symbols with Mathematica

Using the Mathematica package, write the routines for computing Christoffel's symbols.

Problem 4.8 Covariant differentiation with Mathematica

Using the Mathematica package, and the routines developed in Problem 4.7 write the routines for covariant differentiation of tensors up to second order.

Problem 4.9 Divergence of a vector in curvilinear coordinates

Using the Mathematica package and the solution of Problem 4.8, write the routines for computing divergence of a vector in curvilinear coordinates.

Problem 4.10 Laplacian in curvilinear coordinates

Using the Mathematica package and the solution of Problem 4.8, write the routines for computing the Laplacian in curvilinear coordinates.

Problem 4.11 Invariant expressions

Check if any of these tensor expressions are invariant, and correct them if not:

$$A_i B^i_{jk} C^j_{t,k} = D_t \tag{74}$$

$$A^{ij}_{jk} B_{ipq} C^{kq} - F_{kj} G^k_p H^j = H^k A^{jq}_{kj} C^{ti} B_{pit,q} \tag{75}$$

$$E^i B^i_{kp} + D^p_{kq} C^j_{jq} = D_{ki} G_{p,i} \tag{76}$$

Problem 4.12 Contraction invariance

Prove that $A^i B_i$ is an invariant and $A_i B_i$ is not.

A Solutions to problems

Problem 4.1: Check tensor expressions

Check if the following Cartesian tensor expressions violate tensor rules:

$$A_{ij}B_{jk} + B_{pq}C_qD_k = 0$$

Answer: term (1): ik = free, term (2): pk=free

$$E_{pqi}F_{kj}C_{pk} + B_{pj}D_{jq}G_q = F_{kp}$$

Answer: (1): ijq=free (2): p=free (3): kp=free

$$E_{ijk}A_jB_k - D_{ij}A_iB_j = F_{ij}G_{jk}H_{kj}$$

Answer: (1): i=free (2): none, (3): i=free, j = tripple occurrence

---

Problem 4.2: Construct tensor expression

Construct a valid Cartesian tensor expression, consisting of three terms, each including some of the four tensors: $A_{ijk}, B_{ij}, C_i, D_{ij}$. Term 1 should include tensors A, B, C only, term 2 tensor B, C, D and term 3 tensors C, D, A. The expression should have 2 free indexes, which should always come first among the indexes of a tensor. The free indexes should be at A and B in the first term, at B and C in the second term and C and D in the last term. How many different tensor expressions can be constructed?

Solution

One possibility is:

$$A_{ipk}B_{jk}C_p + B_{iq}C_pC_jD_{pq} + C_iD_{jp}A_{pqq} = 0$$

Since there are four locations for dummy indexes in each term, there could be three different combinations of dummies in each term. Thus, the total number of different expression is $3^3 = 27$

Problem 4.3: Cartesian identity

Prove identity (15).

Proof

Integrating (5) in the case of constant transformation marix coefficients, we have:

$$\tilde{x}^i = a^i_k x^k + b^i \tag{77}$$

where the transformation matrix is given by (4):

$$a^i_k \equiv \frac{\partial \tilde{x}^i}{\partial x^k} \tag{78}$$

By the definition of the Cartesian coordinates (79) we have:

$$a^k_i a^k_j = \frac{\partial \tilde{x}^k}{\partial x^i} \frac{\partial \tilde{x}^k}{\partial x^j} = \delta_{ij} \tag{79}$$

Let's multiply the transformation rule (77) by a^i_j. Then we get:

$$a^i_j \tilde{x}^i = a^i_j a^i_k x^k + a^i_j b^i = \delta_{jk} x^k + a^i_j b^i = x^j + a^i_j b^i$$

Differentiation this over $\tilde{x}^i$, we have:

$$a^i_j = \frac{\partial x^j}{\partial \tilde{x}^i}$$

Now rename index j into k:

$$a^i_k = \frac{\partial x^k}{\partial \tilde{x}^i}$$

Comparing this with (78), we have

$$\frac{\partial \tilde{x}^i}{\partial x^j} = \frac{\partial x^j}{\partial \tilde{x}^i}$$

which proves (15).

Problem 4.4: Tensor identity

Using the tensor identity:

$$\varepsilon_{ijk}\varepsilon_{ipq} = \delta_{jp}\delta_{kq} - \delta_{jq}\delta_{kp} \tag{80}$$

prove the vector identity (30):

$$\vec{A} \times (\vec{B} \times \vec{C}) = \vec{B}(\vec{A} \cdot \vec{C}) - \vec{C}(\vec{A} \cdot \vec{B}) \tag{81}$$

Proof

Applying (26) twice to the RHS of (81), we have:

$$\begin{aligned} &\vec{A} \times (\vec{B} \times \vec{C}) \\ &= \varepsilon_{ijk} A_j \varepsilon_{kpq} B_p C_q \\ &= \varepsilon_{ijk}\varepsilon_{kpq} A_j B_p C_q \end{aligned}$$

From (24) it follows that $\varepsilon_{ijk} = -\varepsilon_{ikj} = \varepsilon_{kij}$. Then we have:

$$\varepsilon_{ijk}\varepsilon_{kpq} A_j B_p C_q = \varepsilon_{kij}\varepsilon_{kpq} A_j B_p C_q \tag{82}$$

Now rename the dummy indexes: $k \to i, i \to j, j \to k$, so that the expression looks like one in (29):

$$\begin{aligned} &(\varepsilon_{ijk}\varepsilon_{ipq}) A_k B_p C_q \\ &= (\delta_{jp}\delta_{kq} - \delta_{jq}\delta_{kp}) A_k B_p C_q \\ &= \delta_{jp} B_p \delta_{kq} A_k C_q - \delta_{jq} C_q \delta_{kp} A_k B_p \end{aligned} \tag{83}$$

Using (10), and since $A_j = B_j$ is the same as $A_i = B_i$ the latter can be rewritten as:

$$= B_j A_q C_q - C_j A_p B_p \tag{84}$$

which is the same as

$$\vec{B}(\vec{A} \cdot \vec{C}) - \vec{C}(\vec{A} \cdot \vec{B})$$

Problem 4.5: Metric tensor in cylindrical coordinates.

Cylindrical coordinate system $\tilde{x}^i = \{r, \theta, l\}$ (85) is given by the following transformation rules to a Cartesian coordinate system, $x^i = \{x, y, z\}$:

$$\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta \\ z &= l \end{aligned}$$

Obtain the components of the metric tensor (40) g_{ij} and its inverse g^{ij} (38) in cylindrical coordinates.

Solution:

First compute the derivatives of $x^i = \{x, y, z\}$ with respect to $\tilde{x}^i = \{r, \theta, l\}$:

$$\begin{aligned} \frac{\partial x^1}{\partial \tilde{x}^1} &= \frac{\partial x}{\partial r} \equiv x_r = \cos\theta \\ \frac{\partial x^2}{\partial \tilde{x}^1} &= \frac{\partial y}{\partial r} \equiv y_r = \sin\theta \\ \frac{\partial x^1}{\partial \tilde{x}^2} &= \frac{\partial x}{\partial \theta} \equiv x_\theta = -r\sin\theta \\ \frac{\partial x^2}{\partial \tilde{x}^2} &= \frac{\partial y}{\partial \theta} \equiv y_\theta = r\cos\theta \\ \frac{\partial x^3}{\partial \tilde{x}^3} &= \frac{\partial z}{\partial z} \equiv z_l = 1 \end{aligned} \tag{85}$$

Then the components of the metric tensor are:

$$\begin{aligned} g_{rr} &= x_r x_r + y_r y_r = 1 \\ g_{\theta\theta} &= x_\theta x_\theta + y_\theta y_\theta = r^2 \\ g_{zz} &= 1 \\ g^{rr} &= 1 \\ g^{\theta\theta} &= \frac{1}{r^2} \\ g^{zz} &= 1 \end{aligned}$$

Problem 4.6: Metric tensor in curvilinear coordinates

Using Mathematica, write a procedure to compute metric tensor in curvilinear coordinate system, and use it to obtain the components of metric tensor, g, (40) and its conjugate, $\hat{g}$, (38) in spherical coordinate system (r, ϕ, θ):

$$\begin{aligned} x &= r \sin\theta \cos\phi \\ y &= r \sin\theta \sin\phi \\ z &= r \cos\theta \end{aligned} \tag{86}$$

Solution with Mathematica

```
NX = 3

(* Curvilinear cooridnate system *)
Y = Array[,NX] (* Spherical coordinate system *)
Y[[1]] = r;    (* radius *)
Y[[2]] = th;   (* angle theta *)
Y[[3]] = phi;  (* angle phi *)

(* Cartesian coordinate system *)
X = Array[,NX]
X[[1]] = r Sin[th] Cos[phi];
X[[2]] = r Sin[th] Sin[phi];
X[[3]] = r Cos[th];

(* Compute the Jacobian: dXi/dYj *)
J = Array[,{NX,NX}]
Do[
   J [[i,j]] = D[X[[i]],Y[[j]]],
   {j,1,NX},{i,1,NX}
]

(* Covariant Metric tensor *)
g  = Array[,{NX,NX}] (* covariant *)
Do[
        g [[i,j]] = Sum[J[[k,i]] J[[k,j]],{k,NX}],
        {j,1,NX},{i,1,NX}
```

```
];
g=Simplify[g]

(* Contravariant metric tensor *)
g1 =Array[,{NX,NX}]
g1=Inverse[g]
```

With the result:

$$g = \{\{1,0,0\},\{0,r^2,0\},\{0,0,r^2\sin(\theta)^2\}\}$$
$$\hat{g} = \{\{1,0,0\},\{0,r^{-2},0\},\{0,0,\frac{\csc(\theta)^2}{r^2}\}\}$$

Problem 4.7: Christoffel's symbols with Mathematica

Using the Mathematica package, write the routines to compute Christoffel's symbols

Solution

```
(************* File g.m *************

  The metric tensor
  and Christoffel symbols

************************************)
DIM = 3
(*
        The metric tensor
*)
g  = Array[,{DIM,DIM}] (* covariant *)
g1 =Array[,{DIM,DIM}] (* contravariant *)
Do[
        g [[i,j]] = 0;
        g1[[i,j]] = 0
        ,
        {j,1,DIM},{i,1,DIM}
```

```
]
(*
 Cylindrical coordinates
*)
Z=Array[,DIM]
Z[[1]] = r
Z[[2]] = th
Z[[3]] = z
g [[1,1]] = 1
g [[2,2]] = r^2
g [[3,3]] = 1
g1[[1,1]] = 1
g1[[2,2]] = 1/r^2
g1[[3,3]] = 1
(*
Christoffel symbols of the first and second type
*)
Cr1 = Array[,{DIM,DIM,DIM}]
Cr2 = Array[,{DIM,DIM,DIM}]
Do[
        Cr1[[i,j,k]] = 1/2
        (
           D[ g [[i,k]], Z[[j]] ]
         + D[ g [[j,k]], Z[[i]] ]
         - D[ g [[i,j]], Z[[k]] ]
        ),
        {k,DIM},{j,DIM},{i,DIM}
]
Do[
        Cr2[[l,i,j]] =
                Sum[
                        g1[[l,k]] Cr1[[i,j,k]],
                        {k,DIM}
                ],
        {j,DIM},{i,DIM},{l,DIM}
]
```

Problem 4.8: Covariant differentiation with Mathematica

Using the Mathematica package, write the routines for covariant differentiation of tensors up to second order.

solution

```
(************** File D.m ******************

      Rules of covariant differentiation

*****************************************)
(*
        B.Spain
        Tensor Calculus, 1965
        Eq.(22.2)
*)
D1[N_,A_,k_,X_,j_]:=
(*
   Computes covariant derivative
   of a mixed tensor of second order
   with index k - covariant (upper)
*)
Module[
   {i,s},
   s = Sum[Cr2[[k,i,j]] A[[i]],{i,N}];
   D[A[[k]],X[[j]]] + s
]
Dl1[N_,A_,l_,X_,t_]:=
(*
   Computes covariant derivative
   of a mixed tensor of second order
   with index l - covariant (lower)
*)
Module[
   {s,r},
   s =Sum[Cr2[[r,l,t]] A[[r]],{r,N}];
   D[A[[l]],X[[t]]] - s
]
D1l1[N_,A_,m_,l_,X_,t_]:=
(*
   Computes covariant derivative
   of a mixed tensor of second order
   with index m - contravariant (upper) and
```

```
   index l - covariant (lower)
*)
Module[
   {s1,s2,r},
   s1 =Sum[Cr2[[m,r,t]] A[[r,l]],{r,N}];
   s2 =Sum[Cr2[[r,l,t]] A[[m,r]],{r,N}];
   D[A[[m,l]],X[[t]]] + s1 - s2
]
D2[N_,A_,i_,j_,X_,n_]:=
(*
   Computes covariant derivative
   of second order tensor with
   both m and l contravariant (upper)
   indexes
   B.Spain
   Tensor Calculus, 1965
   Eq.(23.3)
*)
Module[
   {s1,s2,k},
   s1 =Sum[Cr2[[i,k,n]] A[[k,j]],{k,N}];
   s2 =Sum[Cr2[[j,k,n]] A[[i,k]],{k,N}];
   D[A[[i,j]],X[[n]]] + s1 + s2
]
D2l1[N_,A_,i_,j_,k_,X_,n_]:=
(*
   Computes covariant derivative
   of third order tensor with
   i and j contravariant (upper)
   and k contravariant (lower)
   indexes
   B.Spain
   Tensor Calculus, 1965
   Eq.(23.3)
*)
Module[
   {s1,s2,s3,m},
   s1 =Sum[Cr2[[i,m,n]] A[[m,j,k]],{m,N}];
   s2 =Sum[Cr2[[j,m,n]] A[[i,m,k]],{m,N}];
   s3 =Sum[Cr2[[m,k,n]] A[[i,j,m]],{m,N}];
   D[A[[i,j,k]],X[[n]]] + s1 + s2 - s3
```

```
]
D4l1[N_,A_,i1_,i2_,i3_,i4_,i5,X_,i6_]:=
(*
   Computes covariant derivative
   of 5 order tensor with
   4 first indexes contravariant (upper)
   and the last one contravariant (lower)
   B.Spain
   Tensor Calculus, 1965
   Eq.(23.3)
*)
Module[
   {k,s1,s2,s3,s4,s5},
   s1= Sum[Cr2[[i1,k,n]] A[[k,i2,i3,i4,i5]],{k,N}];
   s2= Sum[Cr2[[i2,k,n]] A[[i1,k,i3,i4,i5]],{k,N}];
   s3= Sum[Cr2[[i3,k,n]] A[[i1,i2,k,i4,i5]],{k,N}];
   s4= Sum[Cr2[[i4,k,n]] A[[i1,i2,i3,k,i5]],{k,N}];
   s5=-Sum[Cr2[[k,i5,n]] A[[i1,i2,i3,i4,k]],{k,N}];
   D[A[[i1,i2,i3,i4,i5]],X[[i6]]]+s1+s2+s3+s4+s5
]
```

Problem 4.9: Divergence of a vector in curvilinear coordinates

Using the Mathematica package and the solution of Problem 4.8, write the routines for computing divergence of a vector in curvilinear coordinates.

Solution

Using the algorithms of covariant differentiation developed in Problem 4.8 we have:

```
<<"./g.m" (* The g-tensor and Christoffel symbols *)
<<"./D.m" (* Rules of covariant differentiation *)

(* The original coordinates: *)
NX = DIM
X = Array[,NX]

(* Variables: *)
NV = DIM
```

```
U = Array[,NV]

(* New coordinate system *)
Y = Array[,NX]
Y[[1]] = r;
Y[[2]] = th;
Y[[3]] = z;
X[[1]] = r Cos[th];
X[[2]] = r Sin[th];
X[[3]] = z;

(* Compute the Jacobian *)
J = Array[,{DIM,DIM}]
Do[
   J [[i,j]] = D[X[[i]],Y[[j]]],
   {j,1,DIM},{i,1,DIM}
]
J1=Simplify[Inverse[J]]

(* Derivatives of a vector *)

V0 = Array[,NX]
V0[[1]] = Vr[r,th,z];
V0[[2]] = Vt[r,th,z];
V0[[3]] = Vz[r,th,z];

(*
   Rescaling for physical
  (dimensionally correct) coordinates
(\cite[5.102-5.110]{SyScTC69})
*)
V = Array[,NX]
Do[
   V[[i]] = PowerExpand[V0[[i]]/g[[i,i]]^(1/2)],
   {i,1,NX}
]

(*
   Transform vectors
   as first order contravariant tensors
*)
```

```
U = Array[,NX]
SetAttributes[RV1,HoldAll]
RV1[NX,V,U]
(*
   Compute first covariant derivatives
   of vectors
*)
DV = Array[,{NX,NX}];
Do[
   DV[[i,j]] = D1[NX,V,i,Y,j],
   {j,1,NX},{i,1,NX}
]
(* Divergence *)
div=0
Do[
   div=div+DV[[i,i]],
   {i,NX}
]
div0 = div/.th->0
```

Problem 4.10: Laplacian in curvilinear coordinates

Using the Mathematica package, write the routines for computing Laplacian in curvilinear coordinates.

solution

Using the algorithms of covariant differentiation developed in Problem 4.8 we have:

```
<<"./g.m" (* The g-tensor and Christoffel symbols *)
<<"./D.m" (* Rules of covariant differentiation *)

(* The original coordinates: *)
NX = DIM
X = Array[,NX]

(* Variables: *)
NV = DIM
U = Array[,NV]
```

```
(* New coordinate system *)
Y = Array[,NX]
Y[[1]] = r;
Y[[2]] = th;
Y[[3]] = z;
X[[1]] = r Cos[th];
X[[2]] = r Sin[th];
X[[3]] = z;

(* Compute the Jacobian *)
J = Array[,{DIM,DIM}]
Do[
   J [[i,j]] = D[X[[i]],Y[[j]]],
   {j,1,DIM},{i,1,DIM}
]
J1=Simplify[Inverse[J]]

(* Derivative of a scalar *)

DP = Array[,NX];
Do[
   DP[[i]] = D[p[r,th,z],Y[[i]]],
   {i,1,NX}
]
DDP = Array[,{NX,NX}];
Do[
   DDP[[i,j]] = Dl1[NX,DP,i,Y,j],
   {i,1,NX},{j,1,NX}
]
DDQ = Array[,{NX,NX}];
Do[
   DDQ[[i,j]] = Sum[DDP[[k,l]] J1[[k,i]] J1[[l,j]],{k,NX},{l,NX}],
   {i,1,NX},{j,1,NX}
]

(* Laplacian *)
(***  lap=lap+Sum[g[[i,j]]*Dl1[NX,DS,j,Y,i],{i,1,NX},{j,1,NX}],*)
lap=Sum[DDQ[[i,i]],{i,NX}]
lap0=lap/.th->0
```

Problem 4.11: Invariant expressions

Check if any of these tensor expressions are invariant, and correct them if not:

$$A_i B^i_{jk} C^j_{t,k} = D_t \tag{87}$$

$$A^{ij}_{jk} B_{ipq} C^{kq} - F_{kj} G^k_p H^j = H^k A^{jq}_{kj} C^{ti} B_{pit,q} \tag{88}$$

$$E^i B^i_{kp} + D^p_{kq} C^j_{jq} = D_{ki} G_{p,i} \tag{89}$$

Answers:

A corrected form of (87) is:

$$A_i B^{ik}_j C^j_{t,k} = D_t$$

Equality (89) requires no corrections. A corrected form of (89) is:

$$E_i B^i_{kp} + D_{pkq} C^{jq}_j = D^i_k G_{p,i}$$

Since there are two combinations for an invariant combination of dummy indexes (Corollary 3.5), there can be several different invariant expressions.

Problem 4.12: Contraction invariance

Prove that $A^i B_i = A_i B^i$, and both are invariant, while $A_i B_i$ is not.

Proof

Using the operation of rising/lowering indexes (42), (43), we have

$$A^i B_i = g^{ij} A_j g_{ik} B^k = g^{ij} g_{ik} A_j B^k = \delta_{jk} A_j B^k = A_j B^j$$

which proves that both forms have the same values. If we now consider the first form then:

$$\bar{A}^i \bar{B}_i = \frac{\partial \bar{x}_i}{\partial x_j} A^j \frac{\partial x_k}{\partial \bar{x}_i} B_k = \delta_{jk} A^j B_k = A^j B_j = A^i B_i$$

which proves the point.

Consider now $A_i B_i$:

$$\bar{A}_i \bar{B}_i = \frac{\partial x_j}{\partial \bar{x}_i} A_j \frac{\partial x_k}{\partial \bar{x}_i} B_k$$

which can not be reduced further and, therefore is not invariant, since it has a different form from the LHS.

References

[1] Barry Spain. *Tensor Calculus*. Oliver and Boyd, 1965.

[2] J.L. Synge and A. Schild. *Tensor Calculus*. Dover Publications, 1969.

[3] P. Morse and H. Feshbach. *Methods of Theoretical Physics*. McGraw-Hill, New York, 1953.

Index