# Arrays

Array representation and manipulation routines
Version 1.1
Authors: Doug Burkett, Bhuvanesh Bhatt
Last updated: January 20, 2001
Documentation updated: December 16, 2001

## Introduction:

This is a package to facilitate representation and manipulation of cuboidal arrays with rank>2 (tensors). Tensors are generalizations of scalars, vectors and matrices, and are used in many areas of physics, engineering, and math. The rank is the number of indices needed to specify a particular but arbitrary element of the array. For example, for a matrix we need to specify two numbers – the row and the column, so it is a rank-2 tensor. Tensors can be represented using index notation:

| *Tensor* | *Description* | *Example* |
|---|---|---|
| A | Scalar (no indices) | Temperature |
| $B^{\alpha}$ | Vector (one index) | Velocity |
| $C_{\beta}$ | One-form (dual of vectors) | Gradient |
| $D^{\alpha}{}_{\beta}$ | Matrix (two indices) | Inertia tensor |
| $g^{\alpha\beta}$ | Metric tensor | 3-space Cartesian metric has diagonal elements {1,1,1} |
| $E^{\alpha}{}_{\beta\mu\nu}$ | Rank-4 tensor | Riemann curvature tensor |

Note:

- Greek indices run through the possible dimensions
  (1…3 if dealing with space, 0…3 if dealing with spacetime)
- A superscripted index is called contravariant
- A subscripted index is called covariant
- A tensor with both contravariant and covariant indices is called a mixed tensor

You can perform several operations on tensors:

| *Operation* | *Examples* | *Notes* |
|---|---|---|
| Addition | $A^{\alpha} + B^{\alpha}$<br>$C^{\alpha\beta}{}_{\lambda} + D^{\beta\alpha}{}_{\lambda}$ | Only tensors of the same type can be added; rank remains the same. |
| Multiplication<br>(outer product) | $A^{\alpha}B_{\beta}$ | The new rank is rank(A)+rank(B) |
| Contraction<br>(inner product) | $A^{\alpha}B_{\alpha\beta}$ | The new rank is rank(A)+rank(B)–2 |
| Differentiation | $A^{\alpha}{}_{,\beta}$ | The new rank is rank(A)+1 |
| Covariant differentiation | $A^{\alpha}{}_{;\beta}=A^{\alpha}{}_{,\beta}+\mathbf{\Gamma}^{\alpha}{}_{\beta\mu}A^{\mu}$ | A special kind of tensor derivative; $\mathbf{\Gamma}^{\alpha}{}_{\beta\mu}$ is called a Christoffel symbol. |
| Transposing indices | $A^{\alpha\beta} \rightarrow A^{\beta\alpha}$ | Indices are just shuffled around, so the rank remains the same. |

| | | |
|---|---|---|
| Raising/lowering indices | $g^{\alpha\beta}A_\beta \rightarrow A^\alpha$<br>$g_{\alpha\beta}g_{\mu\nu}B^{\beta\nu} \rightarrow B_{\alpha\mu}$ | Only the position (and labeling) of the indices changes, so the rank remains the same. |
| Common mathematical functions | $\sin(A^{\alpha\beta})$ | The function is applied element-by-element ($\sin(A^{11})$, $\sin(A^{12})$, …) |

We often use the Einstein summation convention, which states that two indices with the same symbol, one contravariant and the other covariant, are implicitly to be summed over. For example, in the expression $A^\alpha B_\alpha$ , the index $\alpha$ is to be summed over its possible values, so this would effectively be a dot product and would return a scalar.

## **Notes for this package**:

The functions in this package are subdivided into three logical levels (not necessarily levels of difficulty). Level-3 operations are specific to general relativity and differential geometry and may take more than 5 minutes. Note that these functions have no notion of contravariant and covariant indices, so you will have to keep track of them by hand. Please do not use the variable 'i' in arrays, as this will give incorrect results. I will try to fix this as soon as I can.

## **Included functions**:

1. Basic array storage/recalling functions:
   - aInd(i,{dims}) returns the array location corresponding to the ith list index for an array of dimensions dims

     Examples: aInd(5,{4,4}) ⇒ {2,1}, aInd(53,{4,4,4}) ⇒ {4,2,1}
     aInd(17,{d}) ⇒ {17}, aInd(992,{2,3,4,5,6}) ⇒ {}
   - aRcl({loc},{dims},{arr}) returns the array element at location loc for an array arr of dimensions dims
   - aSto(val,{loc},{dims},{arr}) returns the array arr of dimensions dims with the value val stored at location loc. To store this changed array, use aSto(val,{loc},{dims},{arr})→myarray
   - newArray({dims}) returns the list corresponding to an array of dimensions dims
2. Fundamental tensor operations:
   - Contract({arr1},{dims1},{arr2},{dims2}) contracts on the last index of arr1 and the first index of arr2. To contract on other indices, first use transpos() to transpose the indices, contract, and then transpose the indices back.

     Example: Contract(seq(1,i,12),{3,2,2},seq(2,i,6),{2,3,1}) ⇒ {4,4,4,4,4,4,…}
   - Diverg({arr},{dims},{coord},i) returns the divergence of arr in the coordinates given by coord wrt index i
   - Gradient({arr},{dims},{coord}) returns the tensor gradient of arr in the coordinates given by coord
   - Inner({arr},{dims},i1,i2) performs an internal inner/dot product on indices i1 and i2 of arr

     Example: Inner({a,b,c,d,e,f,g,h},{2,2,2},1,2) ⇒ {a+g,b+h}
   - Outer(f,{arr1},{dims1},{arr2},{dims2}) performs an outer/tensor/Kronecker product of arr1 and arr2, using the function f to combine elements of arr1 and arr2. f is usually "*" (multiplication).

Examples: Outer(f,{a,b,c},{3},{d,e},{2})
$\Rightarrow$ {f(a,d),f(a,e),f(b,d),f(b,e),f(c,d),f(c,e)}
Outer("*",{a,b,c,d,e,f},{3,2},{k,l,m},{3})
$\Rightarrow$ {a*k,a*l,a*m,b*k,b*l,b*m,c*k,c*l,c*m,d*k,d*l,d*m,
e*k,e*l,e*m,f*k,f*l,f*m}
Note: Please do not use the variable ‘i’ in arrays, as this will give incorrect results. I will try to fix this as soon as I can.

- Transpos({arr},{dims},i1,i2) transposes arr on indices i1 and i2
  Examples: Transpos({a,b,c,d,e,f,g,h},{2,2,2},2,3) $\Rightarrow$ {a,c,b,d,e,g,f,h}
  Transpos({a,b,c,d,e,f,g,h},{2,2,2},1,2) $\Rightarrow$ {a,b,e,f,c,d,g,h}

3. Higher-level tensor operations from general relativity:
   - Christof({g},{coord}) returns the components of the Christoffel symbol of the 2nd kind for the metric g in coordinates coord
     Example: Christof(mat list(diag({-1,1,1,1})),{t,x,y,z}) $\Rightarrow$ {0,0,0,0,0,0,0,0,…}
   - Riemann({g},{coord}) returns the components of the Riemann tensor for the metric g in coordinates coord
     Example: Riemann(mat list(diag({-1,1,1,1})),{t,x,y,z}) $\Rightarrow$ {0,0,0,0,0,0,0,0,…}
   - Ricci({g},{coord}) returns the components of the Ricci tensor for the metric g in coordinates coord
   - RicciSc({g},{coord}) returns the Ricci scalar for the metric g in coordinates coord
   - Einstein({g},{coord}) returns the components of the Einstein tensor for the metric g in coordinates coord

Included programs:

- arrays() creates a custom menu, which also helps with syntax
- aPrint(arr,dims) displays nonzero components of the array arr

Auxiliary functions:

- delElem({list},i) returns list with the ith element deleted
- listSwap({list},i1,i2) swaps the i1 and i2 elements of list

## **Future plans**:

- Common metrics, such as the Minkowski and Schwarzchild metrics
- Curl (using permutation functions from MathTools)
- Covariant derivative
- Geodesic equation
- Weyl tensor

For more details on this array representation method, check the included PDF file. It is a tip from Doug’s tip-list (site hosted by Andrew Cacovean):
http://www.angelfire.com/realm/ti_tiplist/

Questions/comments/suggestions are welcome at bbhatt1@towson.edu