

The `kinner()` function in the `stokes` package

Robin K. S. Hankin



`kinner`

```
## function (o1, o2, M)
## {
##   stopifnot(arity(o1) == arity(o2))
##   k <- arity(o1)
##   if (missing(M)) {
##     M <- diag(nrow = max(c(index(o1), index(o2))))
##   }
##   out <- 0
##   k <- arity(o1)
##   c1 <- elements(coeffs(o1))
##   c2 <- elements(coeffs(o2))
##   for (no1 in seq_len(nterms(o1))) {
##     for (no2 in seq_len(nterms(o2))) {
##       MM <- matrix(0, k, k)
##       for (i in seq_len(k)) {
##         for (j in seq_len(k)) {
##           MM[i, j] <- MM[i, j] + M[index(o1)[no1, i],
##             index(o2)[no2, j]]
##         }
##       }
##       out <- out + det(MM) * c1[no1] * c2[no2]
##     }
##   }
##   return(out)
## }
## <bytecode: 0x55abc7fae4b8>
## <environment: namespace:stokes>
```

Given two k -forms α, β , function `kinner()` returns an inner product $\langle \cdot, \cdot \rangle$ of α and β . If $\alpha = \alpha_1 \wedge \cdots \wedge \alpha_k$ and $\beta = \beta_1 \wedge \cdots \wedge \beta_k$, and we have an inner product $\langle \alpha_i, \beta_j \rangle$ then

$$\langle \cdot, \cdot \rangle = \det \left(\langle \alpha_i, \beta_j \rangle_{ij} \right)$$

We extend this inner product by bilinearity to the whole of $\Lambda^k(V)$.

Some simple examples

Michael Penn uses a metric of

$$\begin{array}{c} dt \quad dx \quad dy \quad dz \\ dt \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \\ dx \\ dy \\ dz \end{array}$$

and shows that

$$\begin{array}{c} dt \wedge dx \quad dt \wedge dy \quad dt \wedge dz \quad dx \wedge dy \quad dx \wedge dz \quad dy \wedge dz \\ dt \wedge dx \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\ dt \wedge dy \\ dt \wedge dz \\ dx \wedge dy \\ dx \wedge dz \\ dy \wedge dz \end{array}$$

so, for example, $\langle dt \wedge dx, dt \wedge dx \rangle = -1$ and $\langle dt \wedge dx, dt \wedge dy \rangle = 0$. We can reproduce this relatively easily in the package as follows. First we need to over-write the default values of `dx`, `dy`, and `dz` (which are defined in three dimensions) and define `dt` `dx` `dy` `dz`:

```
dt <- d(1)
dx <- d(2)
dy <- d(3)
dz <- d(4)
p <- c("dt^dx", "dt^dy", "dt^dz", "dx^dy", "dx^dz", "dy^dz")

mink <- diag(c(1, -1, -1, -1)) # Minkowski metric

M <- matrix(NA, 6, 6)
rownames(M) <- p
colnames(M) <- p

do <- function(x){eval(parse(text=x))}
for(i in seq_len(6)){
  for(j in seq_len(6)){
    M[i,j] <- kinner(do(p[i]), do(p[j]), M=mink)
  }
}
M
```

```
##      dt^dx dt^dy dt^dz dx^dy dx^dz dy^dz
## dt^dx   -1    0    0    0    0    0
## dt^dy    0   -1    0    0    0    0
## dt^dz    0    0   -1    0    0    0
## dx^dy    0    0    0    1    0    0
## dx^dz    0    0    0    0    1    0
## dy^dz    0    0    0    0    0    1
```

Slightly slicker:

```
outer(p,p,Vectorize(function(i,j){kinner(do(i),do(j),M=mink)}))
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]  -1   0   0   0   0   0
## [2,]   0  -1   0   0   0   0
## [3,]   0   0  -1   0   0   0
## [4,]   0   0   0   1   0   0
## [5,]   0   0   0   0   1   0
## [6,]   0   0   0   0   0   1
```

Tidyup

It is important to remove the `dt`, `dx`, `dt`, `dx` as created above because they will interfere with the other vignettes:

```
rm(dt,dx,dy,dz)
```