

Faster matrix algebra with R

```
set.seed(123)
check = function() {
  delta = sum(slow() - fast())
  m = microbenchmark::microbenchmark(slow(), fast())
  times = tapply(m$time[-(1:10)], m$expr[-(1:10)], mean)
  speedup = as.numeric(times[1] / times[2])
  cat(paste('speedup: ', round(speedup,2), ' (delta: 10^', round(log10(abs(delta)),0), ')')
}
n = 150
A = matrix(runif(n*n), n, n)
B = matrix(runif(n*n), n, n)
v = runif(n)
C = A + t(A)
```

Trace of matrix product

```
slow = function() sum(diag(A %*% B))
fast = function() sum(A * t(B))
check()
```

```
## speedup: 16.21 (delta: 10^-Inf)
```

Scaling of a matrix

```
slow = function() diag(v) %*% A %*% diag(v)
fast = function() tcrossprod(v) * A
check()
```

```
## speedup: 30.9 (delta: 10^-15)
```

Matrix times diagonal matrix

```
slow = function() A %*% diag(v)
fast = function() A * rep(v, each=n)
check()
```

```
## speedup: 4.81 (delta: 10^-Inf)
```

```
slow = function() diag(v) %*% A
fast = function() A * v
check()
```

```
## speedup: 53.42 (delta: 10^-Inf)
```

Matrix times its own transpose

```
slow = function() A %*% t(A)  
fast = function() tcrossprod(A)  
check()
```

```
## speedup: 1.69 (delta: 10^-Inf)
```

```
slow = function() t(A) %*% A  
fast = function() crossprod(A)  
check()
```

```
## speedup: 1.54 (delta: 10^-Inf)
```