# Exploring performance capabilities of TMB and RTMB

A critical requirement of a candidate modelling framework is its computational performance. WCPO tuna models are typically highly computationally intensive. Implementing such models with an integrated age- and length-structure would magnify this aspect substantially in respect of the matrix operations required. Consequently, the features of TMB and RTMB that optimise matrix operations were examined in a range of simple tests, to provide an indication of their relative performance benefits, and hence their potential capabilities for implementing an integrated age- and length-structured modelling approach.

The testing design was simply to trial a large matrix operation under two calculation scenarios:

* Non-optimised R operators
* optimised R operators (for sparse matrices) in RTMB

The operation was simply a vector \* matrix multiplication that emulates the application of a growth transition matrix to an age-specific vector length distribution. This was replicated over 10 iterations.

for(i in 1:nsim){

cls4 <- dim(mxsp)[2] # age

mx4 <- mxsp

mx4[,] <- 0

for(j in 1:cls4){

tmp <- mxsp[,j] # vector of lengths within each age-class

mx4[,j] <- txsp %\*% tmp

}

}

Where mxsp is a matrix emulating a hypothetical population state having nrows = 1000 (lengths) and ncols = 1000 (ages), and txsp is a matrix emulating a hypothetical growth transition matrix having nrows = 1000 (lengths) and ncols = 1000 (lengths). These matrices were both banded along the diagonal, with txsp being an upper diagonal banded matrix. As such, both were sparse (<1% non-zero), typical of a length- or an age-length-structured model.

For the scenario: non-optimised R operators; the matrices were of the standard class under the BASE package:

mx <- matrix(0,nrow=rws,ncol=cls)

tx <- matrix(0,nrow=rws2,ncol=cls2)

For the scenario: optimised R operators; the matrices were of the sparse class under the Matrix package that is employed within RTMB for optimisation functions:

library(Matrix)

mxsp <- Matrix(0,nrow=rws,ncol = cls,sparse=TRUE)

txsp <- Matrix(0,nrow=rws2,ncol=cls2,sparse = TRUE)

The processing time was measured for each scenario:

|  |  |
| --- | --- |
| Scenario | Processing time |
| Non-optimised | 24.10591 secs |
| Optimised | 5.940471 secs |

The proportional improvement in processing time was 0.75. This is case-specific depending upon the proportion of zero matrix cells (the sparsity).

A similar test was undertaken where the matrix operation was performed within a function being optimised under two calculation scenarios:

* Non-optimised R operators (for sparse matrices) in TMB
* optimised R operators (for sparse matrices) in TMB

The objective function … (*Arni to input the results of his exercise*)