

Report of *ad hoc* Group on Future ADMB Developments

Present: Dave Fournier, John Sibert, Jim Ianelli and Mark Maunder, Noumea, February 23, 2020.

Increasing the ADMB user base was considered the highest long term priority for the ADMB project. Technical changes that might attract non-fisheries users to ADMB included:

- BUGS to ADMB translator. Some sort of bison (or similar) application to translate a BUGS script into an ADMB tpl file.

- Improve existing `glimmADMB` package from (more complete with options not available in other software e.g. modeling the over dispersion with covariates). Creation of "factor" classes might be useful in this context.

- Create a general count model package and add things that are hard to do outside ADMB (e.g. modeling over dispersion)

- Streamlining the interface between ADMB and R. Might require the means to initialize global variables in share libraries. Coordinate with proposed R project Google Summer of Code initiative.

- A set of example applications similar to those available in WinBUGS (i.e. greatly extend the example applications that are already available). These applications provide an illustration of ADMBs capabilities and can be used as a guide for users developing their own models.

- Support for large random effects models linking high performance matrix libraries into ADMB for the purpose of evaluating the quadratic form in the multivariate normal efficiently. Kasper Kristensen has a lot of experience with this.

Parallelization (aka multiprocessing) was identified as the most important technical improvement. It should be implemented in a way that allows users to easily develop code with parallel computations in appropriate situations. Implementation of parallel processing in ADMB requires someone with experience in development of parallel code with thorough knowledge of the available alternatives. Several options are available:

- pthreads. Requires very careful programming and is probably impossible for ADMB users to implement in a transparent way. Might be useful for certain internal bits of AUTODIF, but difficult to implement.

- PVM. Already implemented in ADMB, but new releases of PVM are infrequent, and it is probably not a general long-term solution

- OpenMP. May offer the best option for future development.

MCMC improvements:

- Hybrid MCMC - “The hybrid Markov Chain Monte Carlo technique affords a robust means for sampling multidimensional probability density functions with high efficiency, provided one can calculate the gradient of ϕ = minus-log-probability in a time comparable to calculating the probability itself. The latter condition is met using the technique of adjoint differentiation.”

<http://public.lanl.gov/kmh/publications/valen02.pdf>

- Rewrite current MCMC code to use constant objects. This should increase the computational overhead by a factor of about 3.

Other:

Easy implementation of 2nd and 3rd derivative adjoint code.