

Example on joint ADMB/TMB development effort

Kasper Kristensen & Anders Nielsen
an@aqua.dtu.dk

June 20, 2016

Joint development TMB/ADMB

- ▶ Focus on a few key improvements of both tools
- ▶ How easy/difficult is it to make these in each tool
- ▶ Possibly learn tricks from one tool to the other
- ▶ Strengthen confidence in both tools

Some focus items

- ▶ Lacking special functions (pbeta, qbeta, besselK, ...)
- ▶ Automatic generator of adaptive p-special functions (Gauss–Kronrod)
- ▶ Automatic generator of adaptive q-special functions
- ▶ Exact (AD) outer hessian
- ▶ Dave's suggestion w.r.t. importance sampling
- ▶ Implement forward sub-sweep
- ▶ Expand the template distributions (in ADMB) to be similar to the TMB density namespace

Example: `pbeta(x,a,b)` in ADMB

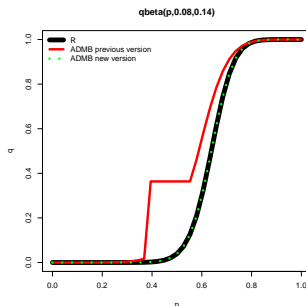
- ▶ Function already in ADMB (as function `betai`)
- ▶ Created alias `pbeta` with same arguments as in **R** to make easier to find
- ▶ Modified the `VECTORIZATION` macros used in TMB to easily expand to all combinations of vectors and scalar calls.
- ▶ Included the `VECTORIZATION` macros in ADMB so can be used for functions defined in template file.

Example: `pbeta(x,a,b)` in TMB

- ▶ Function was not in TMB
- ▶ First attempt use integrate function from **R** — too imprecise
- ▶ Next attempt use code from NR (eventually use code from **R**)
- ▶ Cannot use AD directly.
- ▶ Developed `tiny_ad` to calculate forward AD to any order
- ▶ Useful for such functions with few arguments
- ▶ (also later in ADMB)

Example: $q_{\text{beta}}(x,a,b)$ in ADMB

- ▶ Function already in ADMB (as function `inv_cumd_beta_stable`)
- ▶ Precision not good for all tested cases:



- ▶ Fixed issue
- ▶ Derivatives up to 3th order were hand coded
(`df1b2-separable/df1b2invcumdbeta.cpp`)

Example: Bessel functions in ADMB

- ▶ Modified code from Numerical Recipes.
- ▶ Tested precision
- ▶ Example using von Mises distribution
- ▶ Next try to use `tiny_ad` from TMB