rtmbGMACS Development

William Stockhausen
AFSC

January 14, 2026

rtmbGMACS

- Repository: https://github.com/wStockhausen/rtmbGMACS
  - package website exists, but some pages out of date
- Code in state of flux!!
- Parameterization (ctl file, REs, etc) being addressed
  - changes to existing code will be needed
- Population dynamics well-defined, partially coded
  - allometry, growth, mortality, selectivity
- Selectivity functions defined & coded
- likelihood components not yet addressed!
  - input data structures
  - penalties??
- lots of testing of “intermediate” code (easy in R!)
- Will be lots of changes!

R thoughts

- integer codes (e.g., gmacs ctl file) are not very readable (always need a legend)
  - all codes should be readable text
  - R “factors” allow efficient mapping of text labels to integer codes (if needed)

- data.frames provide a convenient tabular format manipulating text and numbers together
  - can’t include RTMB “AD” values because they lose their “type” information

- **tibbles** are an improved version of data.frames
  - “list” columns allow **any** data type to be included in a tabular structure, including RTMB “AD” types

- the RTMB objective function can be incorporated into an R package that also includes follow-on functions for plotting results, etc.

- Not all R functionality is RTMB-compatible!

Dimensions

Temporal structure

- y: year
- s: season

Fleet structure

- f: fleet

Population structure

- r: region
- x: sex
- m: maturity state
- a: age
- p: post-molt age
- z: size

Dimensions

- defined as “DimsMaps”
  - “sparse” or “dense”
- dimensions can be nested
- different DimsMaps can apply to different aspects of a model
  - model structure
  - datasets
- easy to specify aggregation levels

```
#> Create the individual dimensions as vectors or lists (for nested dimensions)
#> --assign the attribute "dmnms", the dimension name(s) (plural if nested)
#> ----model years
vYs = 2014:2025;                    attr(vYs,"dmnms")<-"y";
#> ----model seasons
vSs = c("fall","spring");           attr(vSs,"dmnms")<-"s";
#> ----model sexes
vXs = c("male","female");           attr(vXs,"dmnms")<-"x";
#> ----model sizes, nested with sexes
vZs = list(male=seq(40,60,5),
           female=seq(25,50,5));    attr(vZs,"dmnms")<-c("x","z");
#>--create the dimensions "map"
dmsSparse = createSparseDimsMap(y=vYs,s=vSs,x=vXs,z=vZs);
```

sparse_idx	y	s	x	z
1	2014	fall	female	25
2	2014	fall	female	30
3	2014	fall	female	35
4	2014	fall	female	40
5	2014	fall	female	45
6	2014	fall	female	50
7	2014	fall	male	40
8	2014	fall	male	45
9	2014	fall	male	50
10	2014	fall	male	55
11	2014	fall	male	60
12	2014	spring	female	25
13	2014	spring	female	30
14	2014	spring	female	35

Parameter/function specification (ctl file)

- set up using R script(s) or text file(s)
  - text files can include R code blocks
  - would be nice to collect all in a “notebook”
- readability:
  - parameters/functions definition grouped by model process (growth, M)
  - parameters/functions identified by name, not integer code
- parameter “structure” (e.g., time blocks, RWs)
  - defined using one-sided R formulas
    - ~ 1 + x:m + (y|yblk) + s(z)
    - can define fixed effects, random effects, smooths on covariates
    - can define contrasts for fixed effects
  - associated with a data frame (DimsMap + covariates)
    - determines “model matrix” for parameter
- function “structure”
  - fixed values (vector, matrix, etc) in tabular format (i.e., data.frame)
  - parameters are “mapped” from parameter definitions (allows sharing)
  - can define random effects for function (additive, multiplicative)
    - not associated with a specific parameter: f = f(p1,p2)*exp(RE)

Example parameter/function specification: allometry

values for region
sizes for all males
sizes for imm females
sizes for mat females
list for x,m,p,z

years
seasons

#--DimsMap for stock categories
#--DimsMap for years and seasons
#--DimsMap for combination
#--Size bins added to dmsC

```
require(rtmbGMACS);
vRs = "EBS";
maz = as.character(seq(25,75,5));
fiz    = as.character(seq(25,65,5));
fmz = as.character(seq(30,65,5));
vXs=list(male=list(all=list(`new`=maz,
                            `old`=maz)
                   ),
         female=list(imm=list(`new`=fiz,
                              `old`=fiz),
                     mat=list(`new`=fmz,
                              `old`=fmz)
                     )
        ); attr(vXs,"dmnms")<-c("x","m","p","z");
ys = as.character(2015:2024); names(ys) = ys;
ss = as.character(1:2);              names(ss) = ss;
dmsC    = createSparseDimsMap(r=vRs,x=vXs);
dmsYS   = createSparseDimsMap(y=ys,s=ss);
dmsYSC  = createSparseDimsMap(y=ys,s=ss,r=vRs,x=vXs);
dimsZBC = dmsC |> dplyr::mutate(lft=as.numeric(z)-2.5,mid=as.numeric(z),rgt=as.numeric(z)+2.5);
fleets = c("Directed","Bycatch","Survey"); #--fishery and survey fleets
```

Example parameter/function specification: allometry

```
#--rtmbGMACS allometry specifications----
PROCESS_ALLOMETRY
 #--OPTIONS
 # pwrLaw1: w(z|pA,pB)      = pA*z^pB
 # pwrLaw2: w(z|pLnA,pB,pZ0) = exp(pLnA+pB*log(z/pZ0))
 #--
 CODE
```

```
mfx  = dmsYSC |> dplyr::filter(x=="male")   |>
            dplyr::select(y,s,x) |> dplyr::distinct() |>
            dplyr::mutate(yn=as.numeric(y),
                          yblk=ifelse(yn<2019,"<2019",
                                      ifelse(yn<2022,"<2022",">=2022")));
mff  = dmsYSC |> dplyr::filter(x=="female") |>
            dplyr::select(y,s,x) |> dplyr::distinct() |>
            dplyr::mutate(yn=as.numeric(y));
mfY     = dmsYSC |> dplyr::select(y) |> dplyr::distinct() |>
            dplyr::mutate(yn=as.numeric(y));
contr_none  = "contr.none";
contr_sum   = list(x="contr.sum",m="contr.sum");
```

```
 END_CODE
```

Example parameter/function specification: allometry

```
 # pwrLaw1: w(z|pA,pB)     = pA*z^pB
 # pwrLaw2: w(z|pLnA,pB,pZ0) = exp(pLnA+pB*log(z/pZ0))
PARAM_EQS
  par_id    par_idx   frame   link_fcn   formula          contrasts
  pLnA        1       mfx      ident      ~1              contr_none    #--males, 1 FE, no REs, no smooths
  pB          1       mfx      ident      ~1+ar1(y|yblk)  contr_none    #--males, 1 FE, AR1 y in yblk REs, no smooths
  pZ0         1       mfY      ident      ~1              contr_none    #--all sexes, 1 FE
  pLnA        2       mff      ident      ~1+s            contr_sum     #--females by season, sum-to-zero contrast
  pB          2       mff      ident      ~0+s+s(y)       contr_sum     #--same + smooth on year
  END_PARAM_EQS
  PARAM_VALS
    values specification to be determined (initial value, limits, mirroring, estimation on/off)
  END_PARAM_VALS
  FUNCTIONS
  fcn_idx   fcn_id      fcn_nm    frame    vars      par_idxs            covariates  link    description
    1       males       pwrLaw2   mfx       z    c(pLnA=1,pB=1,pZ0=1)    ~s(temp)    exp     males:_w(z)_=_exp(pLnA+pB*log(z/pZ0))
    2       females     pwrLaw2   mf        z    c(pLnA=2,pB=2,pZ0=1)       NA       NA      females:_w(z)_=_exp(pLnA+pB*log(z/pZ0))
  END_FUNCTIONS
  FUNCTION_VALS  (if not parameterized)
    values specification to be determined (table of values)
  END_FUNCTION_VALS
```

formulas (lme4, glmmTMB format)

- one-sided specification: ~FEs + REs + smooths
- example: ~ 1 + x + m:yblk + (y|yblk) + ar1(temp|yblk) +s(cold_pool)
- + “frame” determines model matrices and RE/smooths covariance structure for parameter
- smooths described as REs ala `mgcv` “trick”

$$p = \boldsymbol{X} \cdot \beta + \boldsymbol{Z} \cdot r$$
$$r \sim N(0, \Sigma)$$

Lots To Do