```
Structure of UFIT
(UFIT from 28jul2020, with selectable modellers)
-------------------
'->' = 'output'

ufit main program:

read setup for UFIT-specific params
get type of model (UTM, ufit_polynome, etc)
     -> fitfunc={chisq_utm,chisq_model}
read input x-y data to be fitted
read vector of initial fit-params 'a'
read scales
opt randomizing of initial fit-params
modify setup for modeler (UTM or other) for use in fitter (tflag 1:
model from x-pts; wflag=0)
opt display stuff (to get the intial stats, first exec of the
'2 chisq-calls':
     chisqmod=call_function(fitfunc=chisq_xxxx);  returns  chisq  from
     model
     chisq=call_function(priorfunct = chisqallprior);adds chisq from
     priors


case amoeba
     get amoeba-sepcific params from setup
     exofast_amoeba, a, funct=amoeba_utm (fct that returns chisq)
     -> aout final (saved to setup-params)

case MCMC
     get mcmc-specific params from setup
     ufit_exofast_demc,a, chi2func= mcmc_utm  (see below)
     -> mcmc chain  (mc_out_pars, mc_out_chisq)
     update scale in setup
     analysis of  mc_out_pars
     -> aout at best chisq

exec of the 2 chisq-calls on best fit to get intrarr, opt. write
model-lc, and adding chisq to setup

save setup with fitted params
opt display of final fit and stats
     once more exec of the 2 chisq-calls
removal of temporary setup

end
```

```
     ------
     exofast_amoeba,funct=amoeba_utm, P0=a_init,scale

     several times:
     call_function (funct=amoeba_utm)
     -> fitted params a
     end
     ------

          amoeba_utm, atest, (returns chisq for atest)
          ; interface from call by amoeba to chisq_utm
          get common variables (x vals, y data, y model)
          save fitset_temp
          exec of the 2 chisq-calls
          display stuff
          -> chisq
          end

------
     ufit_exofast_demc,bestpars=a_init,  chi2func=  mcmc_utm  ,pars
     (=mcmc-chain), [scales]

     prep stuff
     opt. ufit_exofast_getmcmcscale(a_init,chi2func)
     -> scales
     main loop
          call_function(chi2func)
          call_function(chi2func)
          -> chain of parameters (MCMC chain)
          -> chain of chisq vals
     define burn-in
     some output stuff
     end

-----
          mcmc_utm, atest   (returns chisq for atest)
          ; interface from call by demc to chisq_utm
          get common variables (x vals, y data, y model)
          save fitset_temp
          exec of the 2 chisq-calls
          display stuff
          -> chisq
          end


-------
```

```
chisq_utm
inp: xdat (commonbl)
UTM,fitsetmp,inp=xdat,outp=ymod, some aux values
calc chisq
opt display other stat values
-> chisq
-----------
UTM    (modeller)
inp: xdat (as parameter)
from setup: all parameters for UTM (be they fitted or not)
generate model
opt writing of model to file
->  ymod  (as parameter)
also
-> intrarr (on/off eclipse flag)
-> modeloffvalue    (unocculted brightness)

------
chisq_model
inp: xdat, ydat (commonbl)
call_procedure (model=ufit_polynome or other,
          fitsetmp,inp=xdat,outp=ymod)
calc chisq
opt display other stat values
opt writing of model to file
-> chisq
-------------
ufit_polynome (modeller)
inp: xdat (as parameter)
from setup: all parameters to calc model (be they fitted or not)
calc polynome for xdat
->  ymod (as parameter)
to run as standalone modeller such a routine should also contain:
     - module to read xdat from file if not given as input param
     - writing of model to file, turned off if wflag=0
       (remove writing then from the calling chisq_xxxx module)
     -opt graphic display, turned off if dflag = 0
     -opt adding or subtracting of model and data, turned off if
      tflag =1
```