```
function ocean_MOCW_calc_mean_v5

%{
==========================================================================
Program written to calculate the average Meridional Overturning Circulation
for the time within the max. and min. NADW transport and for the 4 different
runs: LGMG, LGMN, LGMO and NCEP.

Subfunctions used: ocean_file4maxminstartyears
                   ocean_datafiles_zeitser_load_lohmann
                   ocean_vektor2matrix
                   ocean_bekeninfo
                   ocean_curvilinear2kartesian
                   my_nanmean
                   -->/csys/paleo1/pscholz/my_matlab_prog


                   my_om_msec_gr30(written by Martin Butzin)

CAUTION!!!!
The program may contain an error, since the MOCW circulation which is supposed
to be quite closed of the surface was not close enough on the plot of a 20
elements matrix. Indeed the max. of the MOCW was located just 500m too deep.

written: 15.08.07 by Patrick Scholz

14 May 2008: New MOC integration scheme introduced by Martin Butzin.
%}
%==========================================================================
%### LOOP INFO ##############################################################
%==========================================================================
%Select the ocean region:
%oceanregion = 'ato'; % North Atlantic Ocean
oceanregion = 'ata'; % Atlantic Ocean + Arctic Ocean
%oceanregion = 'aso'; % South Atlantic Ocean

oceanvar = 7; %vertical velocity
tt = 0;

%#######################
for runnumber = [1] %Select the runnumber: 1=LGMG, 2=LGMN, 3=LGMO, 4=NCEP
%#######################

%=====================================================================
[ibek_vek  ,...
 xv        ,...
 yv        ,...
 smoothing ,...
 dim_c     ,...
 type      ] = ocean_startup_info;

size(yv)
```

```
    phase_str = 'mean_'
%=====================================================================

    if runnumber == 1

        run_str = 'LGMG'

        %Phase Mean
        start_time = 2495    ;...LGMN-->1940;...LGMG-->2440 ;...LGMO-->1940;
        end_time =   2507    ;...LGMN-->1990;...LGMG-->2490 ;...LGMO-->1990;

    elseif runnumber == 2

        run_str = 'LGMN'

        %Phase Mean
        start_time = 1800   ;...LGMN-->1940;...LGMG-->2440 ;...LGMO-->1940;
        end_time =   1850   ;...LGMN-->1990;...LGMG-->2490 ;...LGMO-->1990;

    elseif runnumber == 3

        run_str = 'LGMO'

        %Phase Mean
        start_time = 1800    ;...LGMN-->1940;...LGMG-->2440 ;...LGMO-->1940;
        end_time =   1850    ;...LGMN-->1990;...LGMG-->2490 ;...LGMO-->1990;

    elseif runnumber == 4

        run_str = 'NCEP'

        %Phase Mean
        start_time = 2100   ;...LGMN-->1940;...LGMG-->2440 ;...LGMO-->1940;
        end_time =   2140    ;...LGMN-->1990;...LGMG-->2490 ;...LGMO-->1990;

    end

    %=========================================================================
    [LGM_startyears] = ocean_file4maxminstartyears(runnumber);
    %=========================================================================
     [k1 ,...
     xi ,...
     yi ] = ocean_bekendata(oceanregion);
    %=========================================================================
    for i = 1:length(LGM_startyears)-1

        if start_time >= LGM_startyears(i,1) && start_time<=LGM_startyears(i+1,1)
            startindex = i
        end

        if end_time > LGM_startyears(i,1) && end_time <=LGM_startyears(i+1,1)
            endindex = i
        end

    end
    %=========================================================================

        count = 1;
```

```
zwischenbreak_str = 'yes';

save_nr = 1;

ocedat_kartesian_xy_time = zeros(length(yi),length(xi),20);

for yearnr = startindex:endindex


    [start_startyear] = ocean_file4startyears(runnumber,yearnr);
    %=====================================================================
    %Download the raw data
    %=====================================================================
    filename = ocean_datafiles_zeitser_load_lohmann(          ...
                                                runnumber , ...
                                                yearnr    , ...
                                                oceanvar  ...
                                               );

    startpath = cd ;
    cd (sprintf('%s%s','/csys/paleo1/pscholz/datafiles_zeitser_lohmann_',run_str));

    ncload(filename);

    cd (startpath)

    %=====================================================================

    MATRIX_var = eval(char(who('var*')));
    lev = eval(char(who('lev')));

    clear var* MATRIX

    [yearmax,depthindex,ind_oceanvek] = size(MATRIX_var)

    %=====================================================================
    %festlegen welcher time_index im .nc files jeweils zum
    %start bzw end Jahr gehört -> aux_start,aux_end
    if yearnr == startindex && yearnr == endindex
       disp 'ich war hier 1'
        aux_start = start_time - LGM_startyears(startindex) + 1;
        aux_end   = end_time - LGM_startyears(endindex) ;
        %aux_end   = end_time - LGM_startyears(endindex) + 1;

    elseif yearnr == startindex
       disp 'ich war hier 2'
        aux_start = start_time - LGM_startyears(startindex) + 1;
        aux_end   = yearmax;

    elseif yearnr == endindex
        disp 'ich war hier 3'
        aux_start = 1;
        aux_end   = end_time - LGM_startyears(endindex) ;
        %aux_end   = end_time - LGM_startyears(endindex) + 1;
    else
        disp 'ich war hier 4'
```

```
    aux_start = 1;
    aux_end = yearmax;
end

aux_start
aux_end

x  = 121:122:ind_oceanvek;
xx = [x x+1];

%-------------------------------------------------------------
% ###  Allocation !!!  ##########################################
%-------------------------------------------------------------
om_mat_kartesian_xy  = zeros(length(yi),length(xi),20)    ;
%-----------------------------------------------------------

for year = aux_start:1:aux_end

    tic

    [om_mat_kartesian_xy] = ocean_vec2mat_xy(  MATRIX_var  ,...
                                xv          ,...
                                yv          ,...
                                xi          ,...
                                yi          ,...
                                k1          ,...
                                ibek_vek    ,...
                                xx          ,...
                                year        ,...
                                depthindex   ,...
                                ind_oceanvek,...
                                smoothing   ...
                                );

%-------------------------------------------------------------------
%ist nur für berechnungvon maxdep ansonsten brauch ich
%nicht und wird auch nur ein einziges mal ausgeführt
if strcmp(zwischenbreak_str,'yes') == 1

    zwischenbreak_str='no'      ;
    smoothing   = 'nearest'     ;
    tslice      = 'lgm'         ;

    ke = 20;


    b = load('om_topo_gr3040');
    beken_info=b.ibek_lgm;
    depto=b.depto_lgm;

    [ommat] = ocean_vektor2matrix(                ...
                                MATRIX_var     , ...
                                year           , ...
                                depthindex     , ...
                                ind_oceanvek ...
```

```
                                    );
    %-----------------------------------------------------
        [d1,d2,d3] = size(ommat);
    %-----------------------------------------------------
        [ocedat_curvilinear_ij,...
         aux_xi,...
         aux_yi                    ] = ocean_bekeninfo_tseries(            ...
                                            ommat       , ...
                                            beken_info  , ...
                                            depto       , ...
                                            depthindex  , ...
                                            oceanregion , ...
                                            lev         ...
                                           );

    [aux_ocedat_cij,...
      aux_ocedat_rxy,...
      aux_zm_ocedat,...
      aux_yi,...
      maxdep,...
      aux_maxdep_test,...
      vlev,...
      aux_kep1,...
      aux_zwp,...
      aux_depto_n,...
      aux_xi,...
      aux_zm_ocedat_test] = my_om_msec_gr30(ocedat_curvilinear_ij,oceanregion,tslice,ke,smoothing,lev);

  %save('ozean_relief_LGM_ata_2.mat','maxdep','yi');
clear ocedat_curvilinear_ij aux_xi aux_yi ...
      aux_ocedat_cij...
      aux_ocedat_rxy...
      aux_zm_ocedat...
      aux_yi...
      aux_maxdep_test...
      vlev...
      aux_kep1...
      aux_zwp...
      aux_depto_n...
      aux_xi...
      aux_zm_ocedat_test...
      b
end

 %-------------------------------------------------------------------
   om_mat_kartesian_xy(isnan(om_mat_kartesian_xy)) = 0;
   ocedat_kartesian_xy_time = ocedat_kartesian_xy_time + om_mat_kartesian_xy;

   count=count+1;

   clear om_mat_kartesian_xy


    t = toc;
```

```
            tt = tt + t;
            ttt = tt/count;
            countdown = ttt * (end_time - start_time - count) / 60;
            disp (sprintf('%s   t=%ss\t countdown=%smin',...
                        num2str(start_startyear + year) ,...
                        num2str(t,'%6.2f')        ,...
                        num2str(countdown,'%6.2f')...
                        ));
            clear t ttt countdown
        end % --> for year = aux_start:aux_end

    end % --> yearnr = startindex:endindex

    ocedat_kartesian_xy_time = ocedat_kartesian_xy_time ./ (count-1);
    [lat, ...
     lon, ...
     depth] = size(ocedat_kartesian_xy_time);

%%% new MOC integration scheme (which needs aux_yi, aux_kep1, and aux_zwp)
    % eliminate land
    knan=find(isnan(ocedat_kartesian_xy_time)==1);
    ocedat_kartesian_xy_time(knan)=0;

    ocedat_kartesian_xy_time(:,:,aux_kep1)=0; % for vertical integration
    dz=diff(aux_zwp);  % vertical grid spacing
    d55=55500;          % length of 0.5 degree in m

    % 3D integration of vertical velocities
    % (1) horizontal integration
    for l=1:aux_kep1,
        % zonal integration
        sw_x(l,:)=sum(ocedat_kartesian_xy_time(:,:,l),2) ...
            *d55.*cos(aux_yi'*pi/180);
        % meridional integration
        if l>1,
            dw(l-1,:)=sw_x(l,:)-sw_x(l-1,:);
            sw_xy(l-1,:)=-cumsum(fliplr(dw(l-1,:)./dz(l-1)),2)*d55;
        end;
    end;

    sw_xy=fliplr(sw_xy);

    for l =1:20
        dz1(l)=dz(l);
    end

    % (2) vertical integration
    %mocw=zeros(l,length(yi));
    for j = 1:length(yi)
        % mocw(:,j)=cumsum(sw_xy(:,j).*dz)/1e6;
        mocw(:,j)=cumsum(sw_xy(:,j).*dz1')/1e6;
    end;
%%% end of MOC integration

    save(sprintf('%s%s%s%s%s%s%s%s%s'               ,...
```

```
                    'MOCW_MEAN_v5'                    ,...(1)
                    run_str                           ,...(2)
                    '_'                               ,...(3)
                    phase_str                         ,...
                    oceanregion                       ,...(4)
                    '_'                               ,...(5)
                    num2str(start_time,'%4.0f')       ,...(6)
                    '-'                               ,...(7)
                    num2str(end_time,'%4.0f')         ,...(8)
                    '.mat'                            ...(9)
                    ),'yi','lev','mocw','ocedat_kartesian_xy_time','maxdep');


          clear ocedat_kartesian_xy_time mocw ...
              aux_yi aux_kep1 aux_zwp dz d55 sw_x dw sw_xy

 end % --> for runnumber = [1 2 3]
clear all


% um allokierten speicher in Matlab wieder vollständig frei zu geben muß
% matlab beendet werden --> quit
%quit
```