```
   ncid = netcdf.open(ncchoice, 'NC_NOWRITE'); ncid_info =
ncinfo(ncchoice)
   % File dimensions, variables and attributes
   [ndims,nvars] = netcdf.inq(ncid);
   % Extracting the list of variable names
   Varnames = [];
   for i = 1:nvars; Varnames{i,1} = netcdf.inqVar(ncid,i-1); end
   for i = 1:nvars; Varnames{i,2} =
strrep(ncid_info.Variables(i).Attributes(strcmp({ncid_info.Variables(i).Attribute
'); end %
   for i = 1:nvars; Varnames{i,3} =
strrep(ncid_info.Variables(i).Attributes(strcmp({ncid_info.Variables(i).Attribute
'); end %
   for i = 1:nvars; [~,fill] = netcdf.inqVarFill(ncid,i-1);
Varnames{i,4} = abs(fill); end
   for i = 1:nvars; Varnames{i,5} =
size(ncid_info.Variables(i).Size,2); end %
   for i = 1:nvars; Varnames{i,6} = i; end %
   Varnames{ismember(Varnames(:,1),{'SAL','SALINITE'}),3} = 'u.s.i.';
   Varnames{ismember(Varnames(:,1),{'TEMP','TEMPERATURE'}),3}
= 'degC'; %
   if strcmp(type,'Sed')
       Varnames(contains(Varnames(:,1),'MES'),3) = {'kg.m-3'};
       Varnames{strcmp(Varnames(:,1),'DZS'),2} = 'Layer thickness';
       Varnames{strcmp(Varnames(:,1),'EPTOT'),2} = 'Total sediment
thickness';
       Varnames{strcmp(Varnames(:,1),'H0'),2} = 'Bathymetry relative
to the mean level';
       Varnames{strcmp(Varnames(:,1),'Nbniv'),2} = 'Number of
levels';
       Varnames{strcmp(Varnames(:,1),'SALINITE'),2} = 'Salinity';
       Varnames{strcmp(Varnames(:,1),'TEMPERATURE'),2}
= 'Temperature';
       Varnames{strcmp(Varnames(:,1),'TENFON'),2} = 'Bed shear
stress';
       Varnames{strcmp(Varnames(:,1),'surf'),2} = 'Mesh size';
   end
   timeorig =
datetime(ncid_info.Variables(find(strcmp('time',Varnames(:,1)))).Attributes(8).Va
   timeCh = 0; timelong =
ncid_info.Variables(find(strcmp('time',Varnames(:,1)))).Size;
   if strcmp(type,'Sed'); min_lev = 0; max_lev = 3;
   else; min_lev = 0; max_lev = 2;%
ncid_info.Variables(strcmp('level',Varnames)).Size;
   end
   Varnames_long=Varnames; % Save the global list of variables...
   Varnames = sortrows(Varnames,5);

   Varnames_List = [{'level'},... #1D ,{'time'},{'SIG'}
       {'H0'},{'surf'},... #2D ,{'latitude'},{'longitude'}
       {'U'},{'V'},{'XE'},{'EPTOT'},{'Nbniv'},{'TENFON'},{'ubot'},...
#3D
```

```
        {'SAL'},{'SALINITE'},{'TEMP'},{'TEMPERATURE'},{'UZ'},{'VZ'},
{'DZS'},... #4D
        {'MES_mediumsand'},{'MES_lightsand'},{'MES_mud'}]';
        if strcmp(type,'Sed')
            Varnames_List = [Varnames_List;{'MES_coarsesand'};
{'MES_gravel'}];
        end
% Selection of extracted vars
    Varnames = Varnames(ismember(Varnames(:,1),Varnames_List),:);
    varch = 1:size(Varnames,1);
    varid = netcdf.inqVarID(ncid,'time');
    tmp = netcdf.getVar(ncid,varid,timeCh,timelong); %,timefilter
    tmp(abs(tmp) == Varnames_long{varid+1,4}) = NaN;
    TIMEdt = seconds(tmp)+timeorig; % Time conversion in seconds to
 datetime
    eval(['Var1D2D.' type '_' 'TIME_' anstr ' = tmp;']);
    eval(['Var1D2D.' type '_' 'TIMEdt_' anstr ' = TIMEdt;']);
    fprintf([etudenc ' ' anstr ' \n time_start = '
 datestr(min(TIMEdt)) ...
        ' \n time_end = ' datestr(max(TIMEdt)) '\n'])
    clear TIMEdt;
    varid = netcdf.inqVarID(ncid,'longitude'); lon_nc =
 netcdf.getVar(ncid,varid)'; % X
      lon_nc(abs(lon_nc) == Varnames_long{varid+1,4}) = NaN;
    varid = netcdf.inqVarID(ncid,'latitude'); lat_nc =
 netcdf.getVar(ncid,varid)'; % Y
      lat_nc(abs(lat_nc) == Varnames_long{varid+1,4}) = NaN;
    lon_nc_tot = lon_nc;
    lon_nc =
 lon_nc(jminid(fz)+1:jmaxid(fz)+1,iminid(fz)+1:imaxid(fz)+1);
    eval(['Var1D2D.' type '_' 'lon_nc_' anstr ' = lon_nc;']);
    lat_nc_tot = lat_nc;
    lat_nc =
 lat_nc(jminid(fz)+1:jmaxid(fz)+1,iminid(fz)+1:imaxid(fz)+1);
    eval(['Var1D2D.' type '_' 'lat_nc_' anstr ' = lat_nc;']);
    for vc=1:length(varch)
        varid = netcdf.inqVarID(ncid,Varnames{vc,1});
        dim=Varnames{vc,5};
        if dim==1
            tmp = netcdf.getVar(ncid,varid); %
            [~,fill] = netcdf.inqVarFill(ncid,varid); tmp(abs(tmp) ==
 abs(fill)) = NaN;
            eval(['Var1D2D.' type '_' Varnames{vc,1} '_' anstr '=
 tmp;']);
        elseif dim==2
            tmp = netcdf.getVar(ncid,varid,[iminid(fz) jminid(fz)],
[range_i(fz) range_j(fz)]); %
            [~,fill] = netcdf.inqVarFill(ncid,varid); tmp(abs(tmp) ==
 abs(fill)) = NaN;
            if strcmp(Varnames{vc,1},'H0'); tmp=tmp'; end
            eval(['Var1D2D.' type '_' Varnames{vc,1} '_' anstr '=
 tmp;']);
        elseif dim==3 && (Simple == 3 || Simple == 34)
            tmp = [];
```

```
                try
                    tmp = netcdf.getVar(ncid,varid,[iminid(fz) jminid(fz)
timeCh],[range_i(fz) range_j(fz) timelong]); %,[1 1 timefilter]
                    [~,fill] = netcdf.inqVarFill(ncid,varid); tmp(abs(tmp)
== abs(fill)) = NaN;
                    tmp = permute(tmp,[2,1,3]);
                catch tmp = NaN(size(lon_nc,1),size(lon_nc,2),timelong);
                end
                eval(['Var3D.' type '_' Varnames{vc,1} '= tmp;']);
                save ([wdworknc, etudenc,'_Var3D_',anstr], 'Var3D');
            elseif dim==4 && (Simple == 4 || Simple == 34)
                tmp_lv = []; tmp = [];
                try
                    for idx = min_lev:max_lev
                        tmp = netcdf.getVar(ncid,varid,[iminid(fz)
jminid(fz) idx timeCh],[range_i(fz) range_j(fz) 1 timelong]); %,[1 1
1 timefilter]
                        [~,fill] = netcdf.inqVarFill(ncid,varid);
tmp(abs(tmp) == abs(fill)) = NaN;
                        tmp_lv = cat(3,tmp_lv,tmp);
                    end
                    tmp_lv = median(tmp_lv,3,'omitnan');
                    tmp_lv = permute(tmp_lv, [2 1 4 3]);
                catch tmp_lv =
NaN(size(lon_nc,1),size(lon_nc,2),timelong);
                end
                eval([type '_' Varnames{vc,1} '= tmp_lv;']);
                save ([wdworknc, etudenc,'_' type '_'
Varnames{vc,1},'_',anstr],[type '_' Varnames{vc,1}]);
                clear tmp tmp_lv
                eval(['clear ' type '_' Varnames{vc,1}]); %
            end
        end % Loop Varnames
        clear tmp
        netcdf.close(ncid) % Closing the file, VERY IMPORTANT
        clear ncid_info
        Varnames(:,1)=strcat([type '_'],Varnames(:,1));
        if job==1
            Varnames_all=Varnames;
        else
            Varnames_all=[Varnames_all; Varnames];
        end
```

Published with MATLAB® R2019b