

```

1 function [gn, clm]=get_ijrg(url, modelgrid, theta_s, theta_b, Tcline, N, Vtransform, Vstretchi
ng)
2 %
3 % here we get the indices from the hycom grid,
4 % compare them to the roms grid,
5 % and then just determine a subset of the hycom grid to obtain data from.
6 %
7 % jcw, revised, Feb 10, 2019
8 %
9
10 %
11 % Read ROMS grid info
12 %
13 disp('getting roms grid dimensions ...');
14 Sinp.theta_s =theta_s;      %surface control parameter
15 Sinp.theta_b =theta_b;      %bottom control parameter
16 Sinp.Tcline   =Tcline;      %surface/bottom stretching width
17 Sinp.N        =N;           %number of vertical levels
18 Sinp.Vtransform =Vtransform; %vertical transformation equation
19 Sinp.Vstretching =Vstretching; %vertical stretching function
20 if (Vtransform==1)
21     h=ncread(modelgrid,'h');
22     hmin=min(h(:));
23     hc=min(max(hmin,0),Tcline);
24 elseif (Vtransform==2)
25     hc=Tcline;
26 end
27 Sinp.hc       =hc;          %stretching width used in ROMS
28 gn=get_roms_grid(modelgrid,Sinp);
29 gn.z_r=shiftdim(gn.z_r,2);
30 gn.z_u=shiftdim(gn.z_u,2);
31 gn.z_v=shiftdim(gn.z_v,2);
32 gn.z_w=shiftdim(gn.z_w,2);
33
34 %
35 % Read HYCOM lon lat depth
36 %
37 display(['getting HYCOM grid data from ', url])
38 ncdisp(url);
39 % jsasaki
40 %numX=ncread(url,'X');
41 %numY=ncread(url,'Y');
42 %ncread(url, 'time')
43 %ncread(url,'depth')
44 %ncread(url,'lat')
45
46 %hycom_lon=ncread(url,'Longitude',[1 1],[length(numX) 1]);
47 %hycom_lat=ncread(url,'Latitude',[1 1],[1 length(numY)]);
48 %hycom_depth=ncread(url,'Depth');
49 hycom_lon=ncread(url,'lon');
50 hycom_lat=ncread(url,'lat');
51 hycom_depth=ncread(url,'depth');
52
53 %
54 % Get roms grid limits
55 %
56 disp('getting roms grid dimensions ...');
57 xl=min(min(gn.lon_rho));xr=max(max(gn.lon_rho));
58 yb=min(min(gn.lat_rho));yt=max(max(gn.lat_rho));
59 %
60 % optimize the chunk size to obtain from hycom
61 %
62 disp('optimizing grid dimensions ...');
63 %
64 % now use xg and yg because we are modifying the lon
65 %
66 xg=hycom_lon;
67 xg(xg>=180)=(xg(xg>=180)-360);
68 yg=hycom_lat;

```

} try catch 之 失败后对应可

```
69 | %
70 | % Find the indices of the roms grid (xl xr yb yt) that are inside the
71 | % hycom grid (xg yg)
72 | %
73 | [ii] = find(xg>=xl & xg<=xr);
74 | [jj] = find(yg>=yb & yg<=yt);
75 | %
76 | % Now just take one more to the left and right
77 | %
78 | ig0=(min(ii)-1); ig1=(max(ii)+1); jg0=(min(jj)-1); jg1=(max(jj)+1);
79 | %
80 | % Constrain indexes to lie within the full HYCOM grid.
81 | %
82 | ig0 = max(ig0, 1);
83 | jg0 = max(jg0, 1);
84 | ig1 = min(ig1, length(hycom_lat));
85 | jg1 = min(jg1, length(hycom_lon));
86 | %
87 | % also save indices as strings
88 | %
89 | irg2=[num2str(ig0) ':' num2str(ig1)];
90 | jrg2=[num2str(jg0) ':' num2str(jg1)];
91 | %
92 | clm.lon=double(xg(ig0:ig1));
93 | clm.lat=double(yg(jg0:jg1));
94 | clm.z=double(hycom_depth);
95 | clm.irg2=irg2;
96 | clm.jrg2=jrg2;
97 | clm.ig0=ig0;
98 | clm.ig1=ig1;
99 | clm.jg0=jg0;
100| clm.jg1=jg1;
```