Thanks Kate. This is detailed story.
First of all, I made just a rectangle grid and give no forces. So here is the '**.h' file.

```
/* '**.h' file ------------------------------------------------*/
#define SOLVE3D
#define ANA_GRID
#define ANA_INITIAL
#define ANA_SMFLUX
#define UV_LOGDRAG
#define ANA_STFLUX
#define ANA_BTFLUX
#define SALINITY
#ifdef SALINITY
# define NONLIN_EOS
# define ANA_SSFLUX
# define ANA_BSFLUX
# define TS_MPDATA
#endif
#define UV_ADV
#undef UV_COR
#define UV_C4ADVECTION
#undef UV_VIS2
#ifdef UV_VIS2
# define MIX_S_UV
#endif
#define SPLINES
#define TS_C4HADVECTION
#define TS_C4VADVECTION
#undef TS_DIF2
#ifdef TS_DIF2
# define MIX_S_TS
#endif
#define MY25_MIXING
#undef ANA_VMIX
#define DJ_GRADPS
#define EASTERN_WALL
#define WESTERN_WALL
#define NORTHERN_WALL
#define SOUTHERN_WALL
#define AVERAGES
#define OUT_DOUBLE
/* end of '**.h' file ------------------------------------------------*/
```

And I made a grid. This is just an example input files and results. I made 10 km * 10 km * 10 m grids: (15 * 10 * 10 grids). So here is the 'ana_grid.h' file.

```
/* this is the inside of ‘ana_grid.h’ */

#elif defined JW0000000
      Xsize=10000.0
      Esize=10000.0
      depth=10.0_r8
      f0=0.0
      beta=0.0_r8
...

#elif defined JW0000000
            dx=Xsize/REAL(Lm(ng),r8)
            dy=Esize/REAL(Mm(ng),r8)
            do j=Jmin,Jmax
                  do i=Imin,Imax
                        yp(i,j)=(j-1)*1000.0
                        yr(i,j)=yp(i,j)+500.0
                        yu(i,j)=yp(i,j)
                        yv(i,j)=yr(i,j)
                  end do
            end do
            do j=Jmin,Jmax
                  do i=Imin,Imax
                        if(i<6) then
                              dx_old = 1000.0
                              dx_new = 1000.0
                        else
                              dx_old = 1000.0
                              dx_new = 500.0
                        end if

                    if(i==0) then
                  xp(i,j) = (i-1)*dx_old
                        else if(i==6) then
                              xp(i,j)=xp(i-1,j)+dx_old
                        else
                              xp(i,j)=xp(i-1,j)+dx_new
                        end if
                        xr(i,j)=xp(i,j)+dx_new/2.0
            xu(i,j)=xp(i,j)
            xv(i,j)=xr(i,j)
                  end do
            end do
/* end of ‘ana_grid.h’ file. */
```

So it is really simple grid. The result when I use single node, ‘NtileI == 1, NtileJ == 1’ in ‘**.in’ file, shows like below.

This is the result of ‘x_psi’.

0	1000	2000	3000	4000	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
0	1000	2000	3000	4000	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
0	1000	2000	3000	4000	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
0	1000	2000	3000	4000	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
0	1000	2000	3000	4000	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
0	1000	2000	3000	4000	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
0	1000	2000	3000	4000	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
0	1000	2000	3000	4000	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
0	1000	2000	3000	4000	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
0	1000	2000	3000	4000	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
0	1000	2000	3000	4000	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000

This is the result of ‘y_psi’

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000
4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000
5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000
7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000
8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000
9000	9000	9000	9000	9000	9000	9000	9000	9000	9000	9000	9000	9000	9000	9000	9000
10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000

It works perfect! However, if I use multiple nodes such as 'NtileI ==2, Ntilej== 1', the result shows like below.

This is the result of 'x_psi' with parallel run.

0	1000	2000	3000	4000	5000	5500	6000	500	1000	1500	2000	2500	3000	3500	4000
0	1000	2000	3000	4000	5000	5500	6000	500	1000	1500	2000	2500	3000	3500	4000
0	1000	2000	3000	4000	5000	5500	6000	500	1000	1500	2000	2500	3000	3500	4000
0	1000	2000	3000	4000	5000	5500	6000	500	1000	1500	2000	2500	3000	3500	4000
0	1000	2000	3000	4000	5000	5500	6000	500	1000	1500	2000	2500	3000	3500	4000
0	1000	2000	3000	4000	5000	5500	6000	500	1000	1500	2000	2500	3000	3500	4000
0	1000	2000	3000	4000	5000	5500	6000	500	1000	1500	2000	2500	3000	3500	4000
0	1000	2000	3000	4000	5000	5500	6000	500	1000	1500	2000	2500	3000	3500	4000
0	1000	2000	3000	4000	5000	5500	6000	500	1000	1500	2000	2500	3000	3500	4000
0	1000	2000	3000	4000	5000	5500	6000	500	1000	1500	2000	2500	3000	3500	4000
0	1000	2000	3000	4000	5000	5500	6000	500	1000	1500	2000	2500	3000	3500	4000

This is the result of 'y_psi' with parallel run.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000
4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000
5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000
7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000
8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000
9000	9000	9000	9000	9000	9000	9000	9000	9000	9000	9000	9000	9000	9000	9000	9000
10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000

So what do you think about this problem? Look at the red numbers.