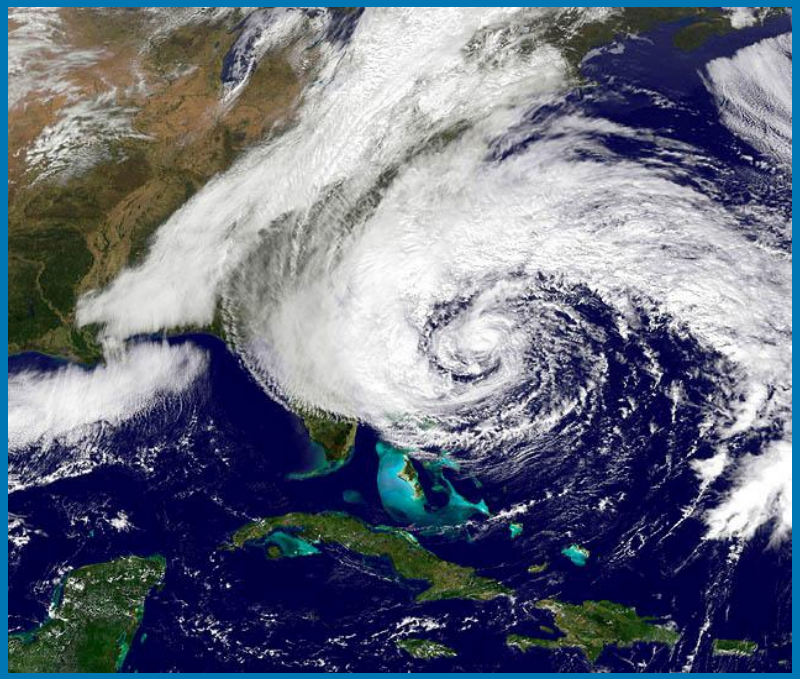
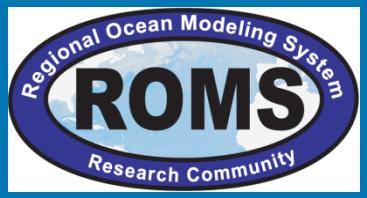
USGS
science for a changing world

ROMS Application Tutorial



Projects/Sandy- Hurricane Sandy example

U.S. Department of the Interior
U.S. Geological Survey

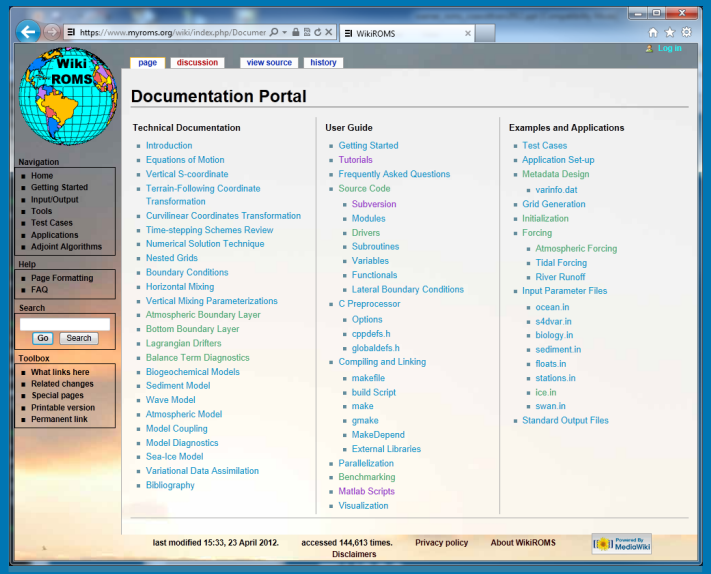


Regional Ocean Modeling System

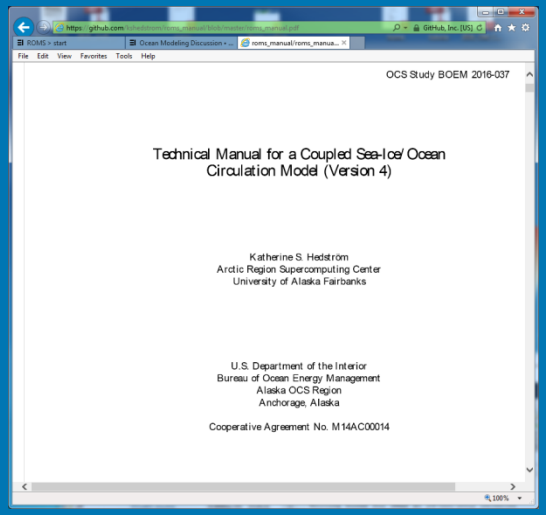
- Free surface, hydrostatic ocean model
- Finite-difference 3D Reynolds-averaged Navier-Stokes equations
- Horizontal orthogonal curvilinear Arakawa C grid
- Vertical stretched terrain-following Sigma coordinates
- Wide range of advection schemes: (e.g. 3rd-order upstream-biased, 4^{th}-order)
- Wide range of open boundary conditions: (e.g. Radiation, clamped, nudged)
- CF-compliant NetCDF I/O
- Wide range of vertical mixing schemes (k-epsilon, k-omega, MY2.5, KPP, GLS)
- Ice model
- Biological modules
- Model adjoint for data assimilation
- Fortran 90; Runs on Unix, Mac, and Windows
- Parallel code in MPI and OpenMP

USGS

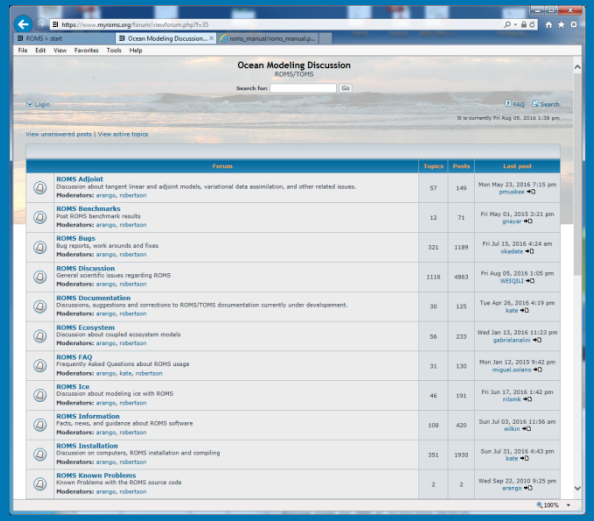
ROMS information



https://www.myroms.org/wiki



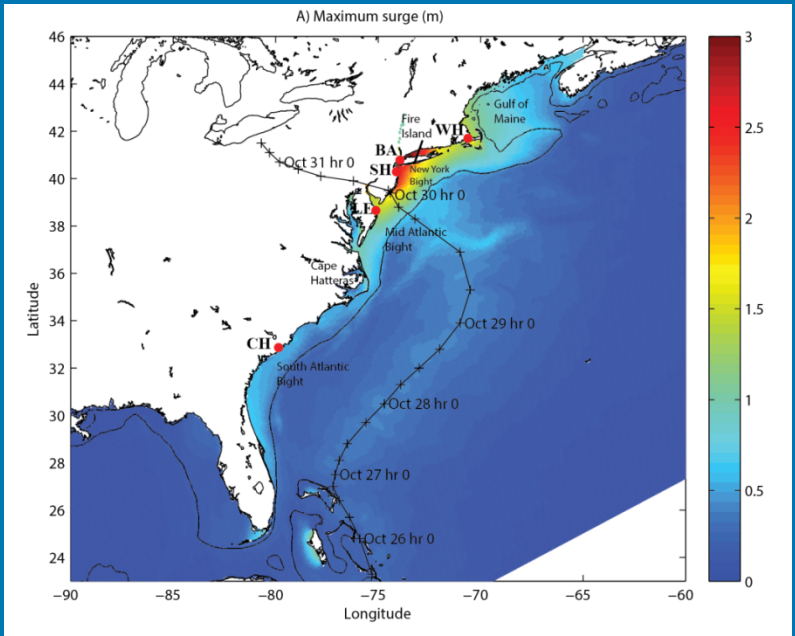
https://github.com/kshedstrom/roms_manual/blob/master/roms_manual.pdf



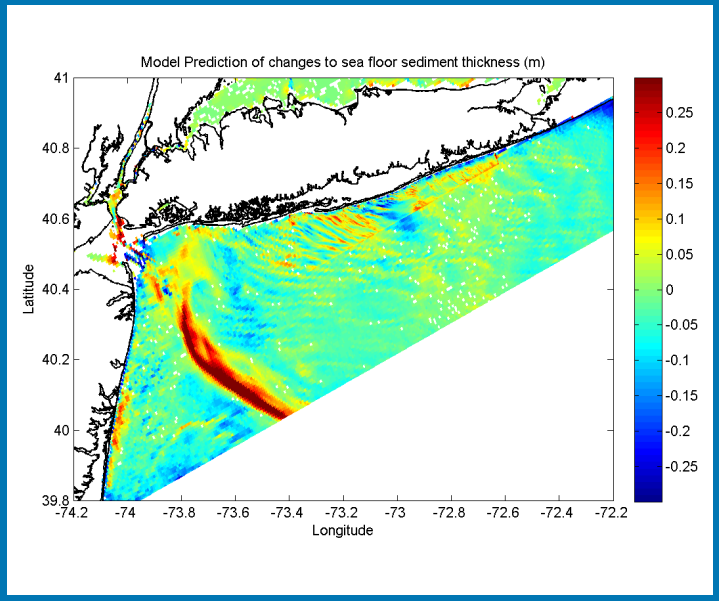
https://www.myroms.org/forum/index.php

USGS

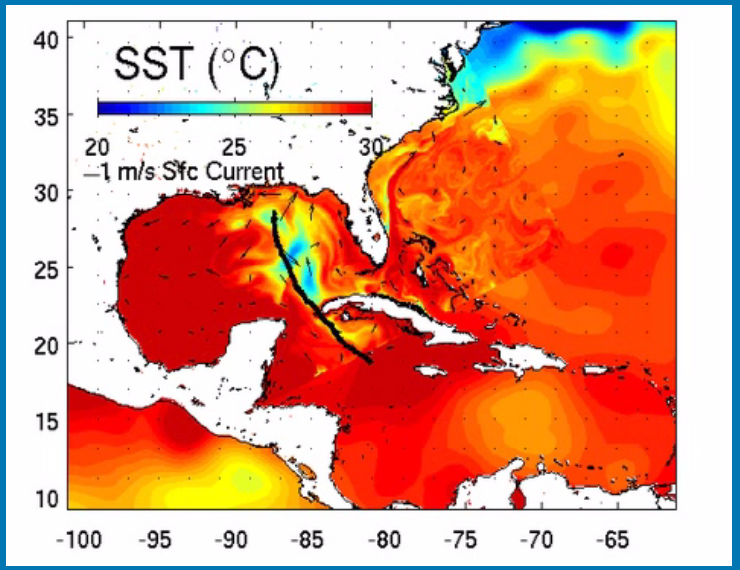
Wide Range of realistic Applications



Storm surge (H. Sandy)



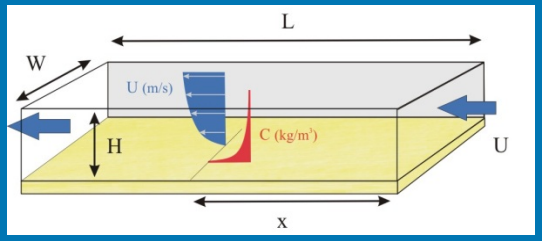
seafloor bed thickness change (H. Sandy)

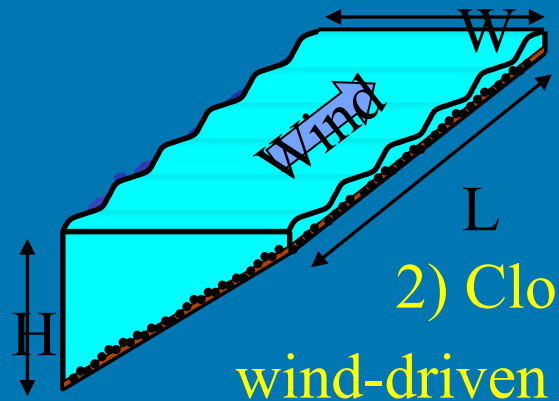


SST (H. Ivan)

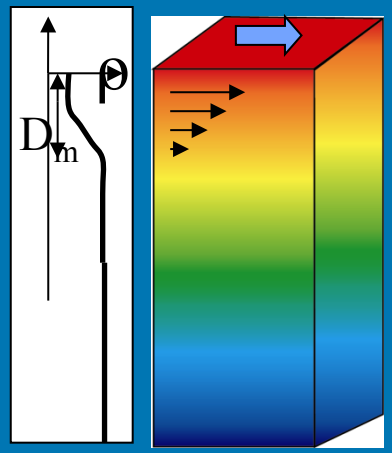
Test Cases

1) Open channel flow

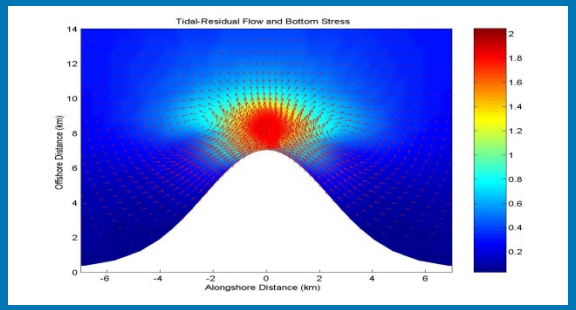




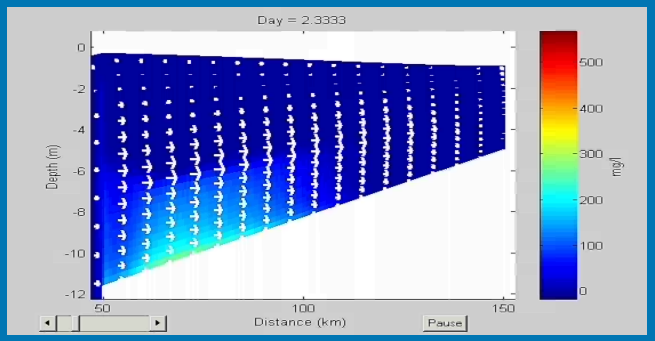
2) Closed basin, wind-driven circulation



3) Mixed layer deepening



4) Tidal flow around a headland



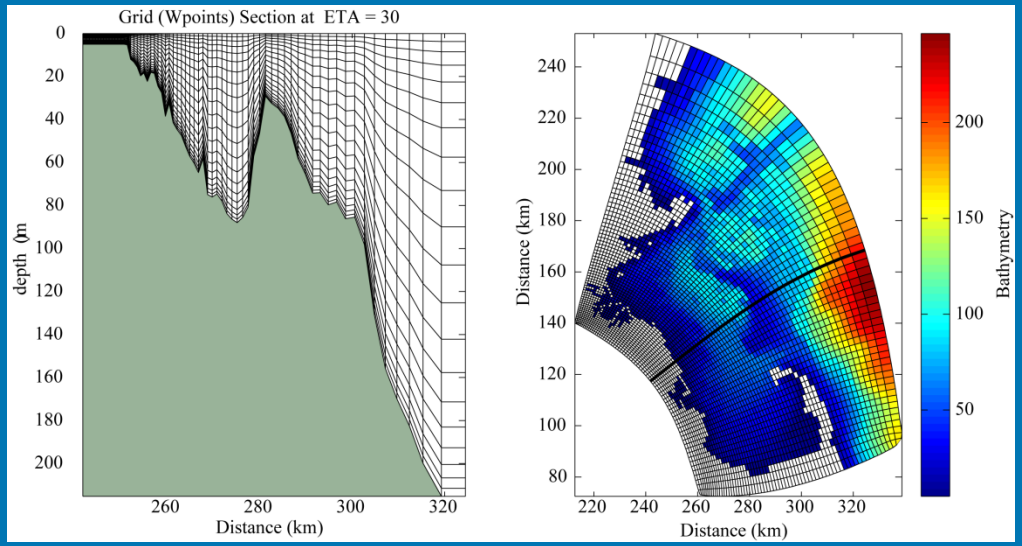
5) Estuarine circulation

USGS

http://woodshole.er.usgs.gov/project-pages/sediment-transport/

ROMS Grid

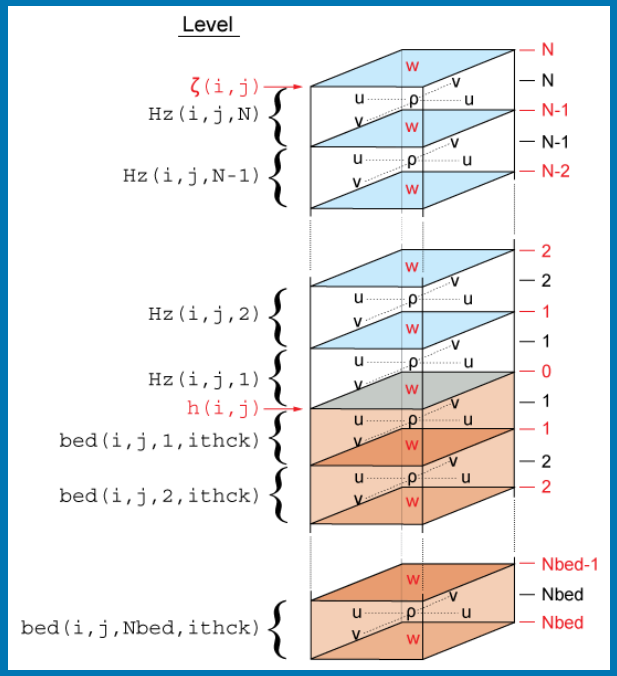
terrain following coordinates

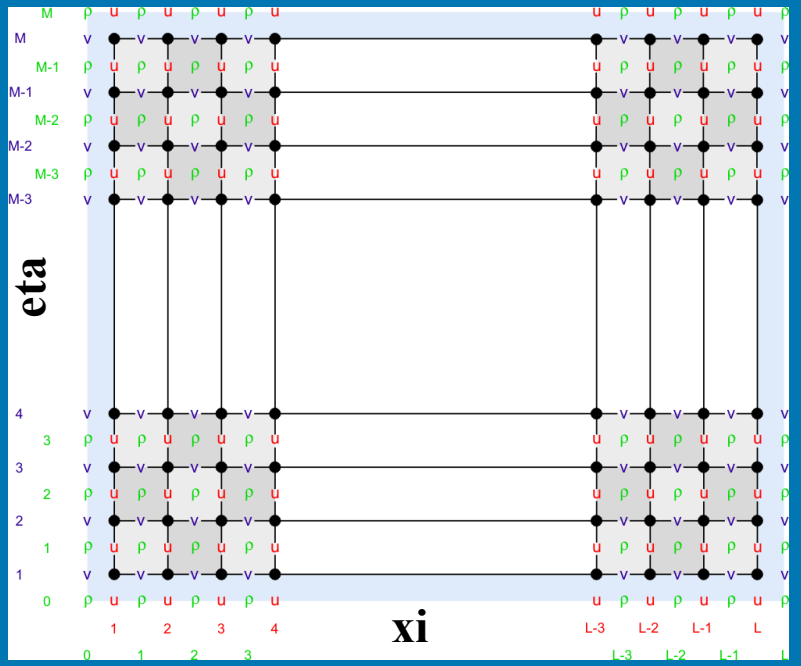


- masking
- curvature
- stretching

Horizontal
Arakawa "C" grid

Vertical

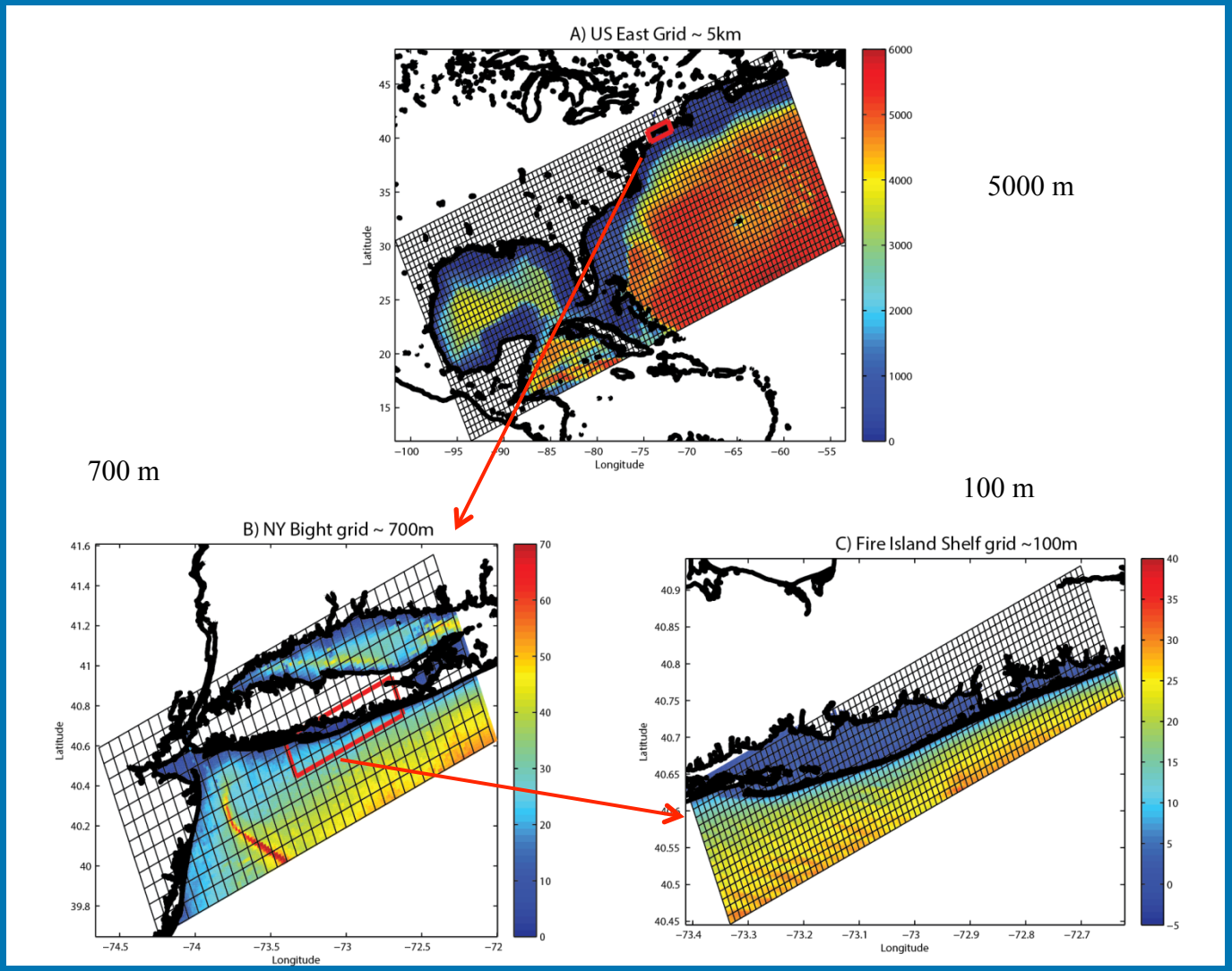




USGS

Grid resolution – horizontal static nests

For structured grids, use grid refinement to increase resolution.



Want to increase resolution in areas of strong gradients and in areas of changing processes.

Terrain following transformations

(Vtransform + Vstrectching)

Vtransform

Transformation Equations

The following vertical coordinate transformations are available:

$$z(x,y,\sigma,t) = S(x,y,\sigma) + \zeta(x,y,t)\left[1 + \frac{S(x,y,\sigma)}{h(x,y)}\right], \qquad (1)$$

$$S(x,y,\sigma) = h_c\,\sigma + [h(x,y) - h_c]\,C(\sigma)$$

1

or

$$z(x,y,\sigma,t) = \zeta(x,y,t) + [\zeta(x,y,t) + h(x,y)]\;S(x,y,\sigma), \qquad (2)$$

$$S(x,y,\sigma) = \frac{h_c\,\sigma + h(x,y)\,C(\sigma)}{h_c + h(x,y)}$$

2

where $S(x,y,\sigma)$ is a nonlinear vertical transformation functional, $\zeta(x,y,t)$ is the time-varying free-surface, $h(x,y)$ is the unperturbed water column thickness and $z = -h(x,y)$ corresponds to the ocean bottom, σ is a fractional vertical stretching coordinate ranging from $-1 \le \sigma \le 0$, $C(\sigma)$ is a nondimensional, monotonic, vertical stretching function ranging from $-1 \le C(\sigma) \le 0$, and h_c is a positive thickness controlling the stretching. In sediment applications, $h = h(x,y,t)$ is changed at every time-step since it is affected by erosion and deposition processes.

USGS

https://www.myroms.org/wiki/index.php/Vertical_S-coordinate

Vstretch (1 of 4)

Available Stretching Functions:

1. Song and Haidvogel (1994) function available in ROMS since its beginning, Vstretching = 1. $C(\sigma)$ is defined as:

$$C(\sigma) = (1 - \theta_B)\frac{\sinh(\theta_S\,\sigma)}{\sinh\theta_S} + \theta_B\left[\frac{\tanh[\theta_S(\sigma + \frac{1}{2})]}{2\tanh(\frac{1}{2}\theta_S)} - \frac{1}{2}\right] \tag{5}$$

where θ_S and θ_B are the surface and bottom control parameters. Their ranges are $0 < \theta_S \leq 20$ and $0 \leq \theta_B \leq 1$, respectively. This function has the following features:

- It is infinitely differentiable in σ.
- The larger the value of θ_S, the more resolution is kept above h_c.
- For $\theta_B = 0$, the resolution all goes to the surface as θ_S is increased.
- For $\theta_B = 1$, the resolution goes to both the surface and the bottom equally as θ_S is increased.
- For $\theta_S \neq 0$, there is a subtle mismatch in the discretization of the model equations, for instance in the horizontal viscosity term. We recommend that you stick with *reasonable* values of θ_S, say $\theta_S \leq 8$.
- Some applications turn out to be sensitive to the value of θ_S used.

USGS

Vstretch (2 of 4)

2. A. Shchepetkin (2005) UCLA-ROMS deprecated function, Vstretching = 2. $C(\sigma)$ is defined as a piecewise function:

$$C(\sigma) = \mu\, C_{sur}(\sigma) + (1-\mu)\, C_{bot}(\sigma),$$

$$C_{sur}(\sigma) = \frac{1-\cosh(\theta_S\,\sigma)}{\cosh(\theta_S)-1}, \qquad \text{for } \theta_S > 0, \qquad C_{bot}(\sigma) = \frac{\sinh[\theta_B\,(\sigma+1)]}{\sinh(\theta_B)} - 1,$$

$$\mu = (\sigma+1)^{\alpha}\left[1 + \frac{\alpha}{\beta}\left(1-(\sigma+1)^{\beta}\right)\right]$$

This function is similar in meaning to the Song and Haidvogel (1994), but note that hyperbolic function in C_{sur} is $\cosh$ instead of $\sinh$ and

$$\frac{\partial C}{\partial \sigma} \to 0 \qquad \text{as} \quad \sigma \to 0.$$

USGS

Vstretch (3 of 4)

3. R. Geyer function for high bottom boundary layer resolution in relatively shallow applications, Vstretching = 3. $C(\sigma)$ is defined as a piecewise function:

$$C(\sigma) = \mu\, C_{bot}(\sigma) + (1-\mu)\, C_{sur}(\sigma),$$

$$C_{sur}(\sigma) = -\frac{\log[\cosh(\gamma\, \mathrm{abs}(\sigma)^{\theta_S}]}{\log[\cosh(\gamma)]}, \qquad C_{bot}(\sigma) = \frac{\log[\cosh(\gamma\,(\sigma+1)^{\theta_B}]}{\log[\cosh(\gamma)]} - 1,$$

$$\mu = \frac{1}{2}\left[1 - \tanh\left(\gamma\left(\sigma + \frac{1}{2}\right)\right)\right]$$

where the power exponents θ_S and θ_B are the surface and bottom control parameters specified in standard input file ocean.in, as before. Here, γ is a scale factor for all hyperbolic functions. Currently, $\gamma = 3$. Typical values for the control parameters are:

$\theta_S = 0.65$	minimal increase of surface resolution
$\theta_S = 1.0$	significant surface amplification
$\theta_B = 0.58$	no bottom amplification
$\theta_B = 1.0$	significant bottom amplification

USGS

Vstretch (4 of 4)

4. A. Shchepetkin (2010) UCLA-ROMS current function, Vstretching = 4. $C(\sigma)$ is defined as a continuous, double stretching function:

Surface refinement function:

$$C(\sigma) = \frac{1 - \cosh(\theta_S\,\sigma)}{\cosh(\theta_S) - 1}, \qquad \text{for } \theta_S > 0, \qquad C(\sigma) = -\sigma^2, \qquad \text{for } \theta_S \leq 0$$

Bottom refinement function:

$$C(\sigma) = \frac{\exp(\theta_B\,C(\sigma)) - 1}{1 - \exp(-\theta_B)}, \qquad \text{for } \theta_B > 0 \tag{9}$$

Notice that the bottom function (9) is the second stretching of an already stretched transform (8). The resulting stretching function is continuous with respect to θ_S and θ_B as their values approach zero. The range of meaningful values for θ_S and θ_B are:

$$0 \leq \theta_S \leq 10 \qquad \text{and} \qquad 0 \leq \theta_B \leq 4$$

However, users need to pay attention to extreme r-factor (rx1) values near the bottom.

Due to its functionality and properties Vtransform = 2 and Vstretching = 4 are now the default values for ROMS.

USGS

Equations in Mass Flux form

$$\left(\mathbf{u}^{l},\omega^{l}\right)=\left(\mathbf{u},\omega\right)+\left(\mathbf{u}^{St},\omega^{St}\right)$$

u, v, ωs = Eulerian Velocity
u^{l}, v^{l}, ωsl = Lagrangian Velocity
u^{st}, v^{st}, ωsSt = Stokes Velocity
f = Coriolis parameter
φ = Dynamic pressure
H_z = Grid cell thickness
$\mathcal{F}\uparrow\eta;\mathcal{F}\uparrow\xi$ = Non wave body force
$D\uparrow\eta;D\uparrow\xi$ = Momentum mixing terms
$\mathcal{F}\uparrow w\eta;\mathcal{F}\uparrow w\xi$ = Non-conservative wave force

Continuity

$$\frac{\partial}{\partial t}\left(\frac{H_z}{mn}\right)+\frac{\partial}{\partial \xi}\left(\frac{H_z u^{l}}{n}\right)+\frac{\partial}{\partial \eta}\left(\frac{H_z v^{l}}{m}\right)+\frac{\partial}{\partial s}\left(\frac{\omega_s^{l}}{mn}\right)=0$$

xi-direction Momentum Balance

$$\underbrace{\frac{\partial}{\partial t}\left(\frac{H_z}{mn}u\right)}_{\text{ACC}}+\underbrace{\frac{\partial}{\partial \xi}\left(\frac{H_z u}{n}u\right)+\frac{\partial}{\partial \eta}\left(\frac{H_z v}{m}u\right)+u\frac{\partial}{\partial \xi}\left(\frac{H_z u^{St}}{n}\right)+u\frac{\partial}{\partial \eta}\left(\frac{H_z v^{St}}{m}\right)}_{\text{HA}}$$

$$+\underbrace{\frac{\partial}{\partial s}\left(\frac{\omega_s}{mn}u\right)+u\frac{\partial}{\partial s}\left(\frac{\omega_s^{St}}{mn}\right)}_{\text{VA}}-\underbrace{H_z\left(\frac{fv}{mn}\right)}_{\text{COR}}-\underbrace{H_z\left(\frac{fv^{st}}{mn}\right)}_{\text{StCOR}}=\underbrace{-\frac{H_z}{n}\frac{\partial \varphi^{c}}{\partial \xi}\bigg|_{z}}_{\text{PG}}+\underbrace{H_z v^{st}\left(\frac{1}{n}\frac{\partial v}{\partial \xi}-\frac{1}{m}\frac{\partial u}{\partial \eta}\right)-\omega_s^{St}\frac{\partial}{\partial s}\left(\frac{u}{mn}\right)}_{\text{HVF}}$$

$$+\underbrace{\frac{H_z F^{\xi}}{mn}}_{\text{BF}}+\underbrace{\frac{H_z F^{w\xi}}{mn}}_{\text{BA+RA+BtSt+SuSt}}+\underbrace{\frac{H_z D^{\xi}}{mn}}_{\text{HM}}-\underbrace{\frac{\partial}{\partial s}\left(\overline{u'w'}-\frac{\nu}{H_z}\frac{\partial u}{\partial s}\right)}_{\text{VM}}+\underbrace{\hat{F}^{u}}_{\text{FCurv}}$$

eta-Direction Momentum Balance

$$\underbrace{\frac{\partial}{\partial t}\left(\frac{H_z}{mn}v\right)}_{\text{ACC}}+\underbrace{\frac{\partial}{\partial \xi}\left(\frac{H_z u}{n}v\right)+\frac{\partial}{\partial \eta}\left(\frac{H_z v}{m}v\right)+v\frac{\partial}{\partial \xi}\left(\frac{H_z u^{St}}{n}\right)+v\frac{\partial}{\partial \eta}\left(\frac{H_z v^{St}}{m}\right)}_{\text{HA}}$$

$$+\underbrace{\frac{\partial}{\partial s}\left(\frac{\omega_s}{mn}v\right)+v\frac{\partial}{\partial s}\left(\frac{\omega_s^{St}}{mn}\right)}_{\text{VA}}+\underbrace{H_z\left(\frac{fu}{mn}\right)}_{\text{COR}}+\underbrace{H_z\left(\frac{fu^{st}}{mn}\right)}_{\text{StCOR}}=\underbrace{-\frac{H_z}{m}\frac{\partial \varphi^c}{\partial \eta}\bigg|_z}_{\text{PG}}\underbrace{-H_z u^{st}\left(\frac{1}{n}\frac{\partial v}{\partial \xi}-\frac{1}{m}\frac{\partial u}{\partial \eta}\right)-\omega_s^{St}\frac{\partial}{\partial s}\left(\frac{v}{mn}\right)}_{\text{HVF}}$$

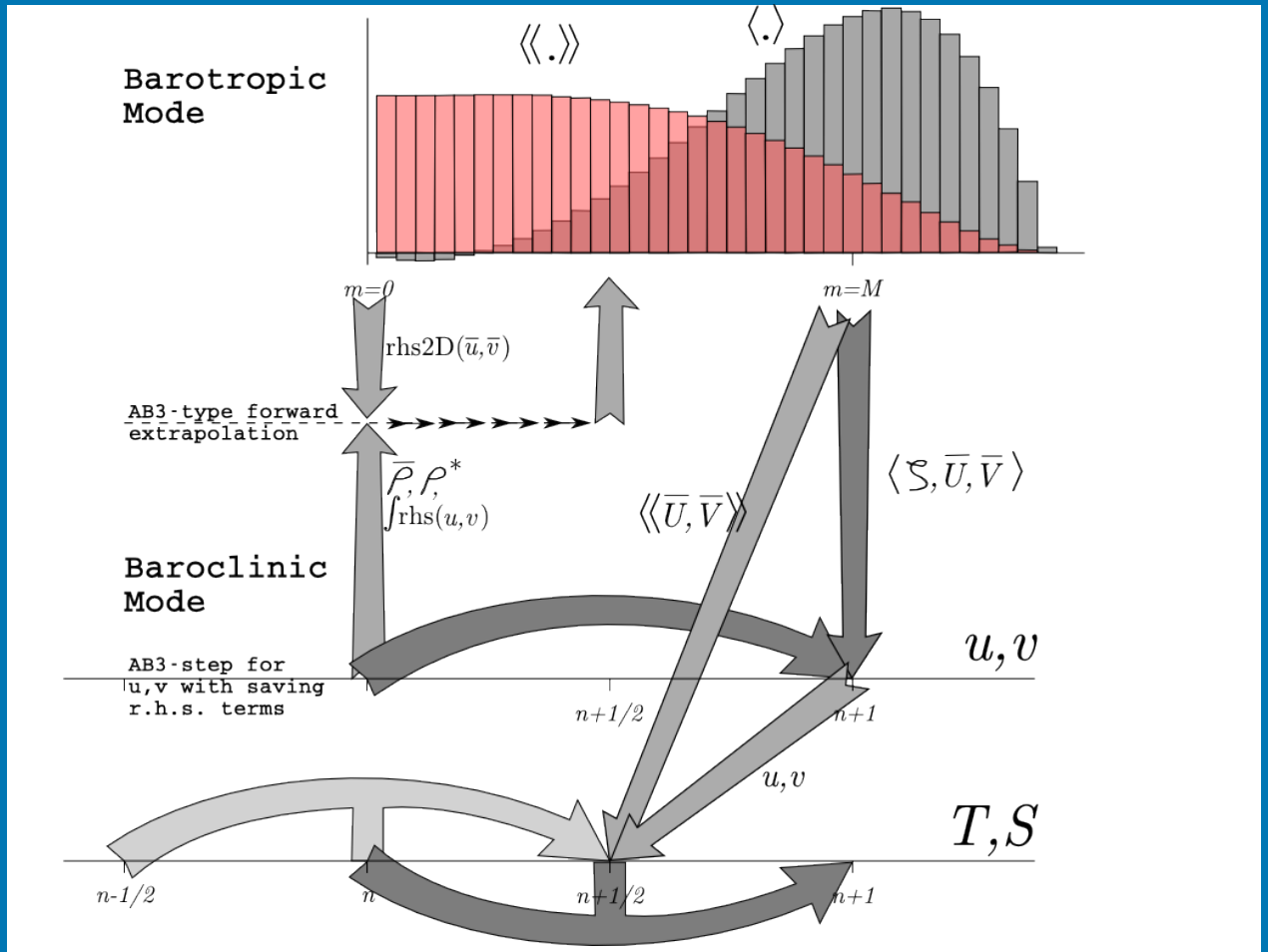
$$+\underbrace{\frac{H_z F^{\eta}}{mn}}_{\text{BF}}+\underbrace{\frac{H_z F^{w\eta}}{mn}}_{\text{BA+RA+BtSt+SuSt}}+\underbrace{\frac{H_z D^{\eta}}{mn}}_{\text{HM}}\underbrace{-\frac{\partial}{\partial s}\left(\overline{v'w'}-\frac{\nu}{H_z}\frac{\partial v}{\partial s}\right)}_{\text{VM}}+\underbrace{\hat{F}^{v}}_{\text{FCurv}}$$

Description of Terms

ACC	= Local Acceleration
HA	= Horizontal Advection
VA	= Vertical Advection
COR	= Coriolis Force
StCOR	= Stokes-Coriolis Force
PG	= Pressure Gradient
HVF	= Horizontal Vortex Force
BF	= Body Force
BA+RA+BuSt+SuSt	= Breaking Acceleration+ Roller Acceleration+ Bottom Streaming+ Surface Streaming
HM	= Horizontal Mixing
VM	= Vertical Mixing
Fcurv	= Curvilinear terms

Kumar, N., Voulgaris, G., Warner, J.C., and M., Olabarrieta (2012). Implementation of a vortex force formalism in a coupled modeling system for inner-shelf and surf-zone applications. Ocean Modelling, 47, 65-95.

Solution techniques- mode splitting



Shchepetkin, A. F., and J. C. McWilliams, 2008: Computational kernel algorithms for fine-scale, multi-process, long-time oceanic simulations. In: *Handbook of Numerical Analysis: Computational Methods for the Ocean and the Atmosphere*, eds. R. Temam & J. Tribbia, Elsevier Science, ISBN-10: 0444518932, ISBN-13: 978-0444518934.

USGS

Numerical algorithms

$$\frac{\partial}{\partial t}\frac{H_z C}{mn} = -\frac{\partial}{\partial \xi}F^{\xi} - \frac{\partial}{\partial \eta}F^{\eta} - \frac{\partial}{\partial \sigma}F^{\sigma}$$

Advection schemes

- **2nd order centered** →
- **4th order centered**
- **4th order Akima**
- **TSU3 3rd order**
- **MPDATA**
- **HSIMT**

$$F^{\xi} = \frac{\overline{H_z}^{\xi} u \overline{C}^{\xi}}{\overline{n}^{\xi}}$$
$$F^{\eta} = \frac{\overline{H_z}^{\eta} v \overline{C}^{\eta}}{\overline{m}^{\eta}}$$
$$F^{\sigma} = \frac{\overline{H_z}^{\sigma} \Omega \overline{C}^{\sigma}}{mn}.$$

many choices, see manual for details. ……

USGS

How do we select different schemes

- c pre-processor definitions (list them in your project *.h “header” file)
- during compilation, .F → .f90s
- compiles f90's into objects
- compiles objects to libs
- ar the libs to make 1 exe
- for coupling, it makes wrf, roms, and swan as libs, then pull them together for coupling and only produces one exe → coawstM [.exe]

USGS

Build errors

- During the build, the error “can not make wrf.exe” or ‘undefined reference to MAIN” is ok.

libdev/frtl/src/libfor/for_main.c:(.text+0x2a): undefined reference to `MAIN__'
make[2]: [em_wrf] Error 1 (ignored)
This “Error” is ok.

If you get other errors, do this:

scrip build.txt
./coawst.bash
exit
and that will create a file build.txt
Edit that file and search for the word “error”

USGS

COAWST specific cpp options

You need to use at least 1 model of the 3:

- #define ROMS_MODEL — if you want to use the ROMS model
- #define SWAN_MODEL — if you want to use the SWAN model
- #define WRF_MODEL — if you want to use the WRF model

- #define MCT_LIB — if you have more than one model selected

Use these 3 INTERP calls if the models are on different grids.

- #define MCT_INTERP_WV2AT — allows grid interpolation between the wave and atmosphere models
- #define MCT_INTERP_OC2AT — allows grid interpolation between the ocean and atmosphere models
- #define MCT_INTERP_OC2WV — allows grid interpolation between the ocean and wave models

- #define NESTING — allows grid refinement in roms or in swan
  For now, I suggest if you couple ROMS+SWAN and use NESTING, then use the same grids for R+S.

USGS

COAWST specific cpp options

If you activate ROMS + WRF, then pick one of these:

- #define ATM2OCN_FLUXES — provide consistent fluxes between atm and ocn. WRF send USTRESS,VSTRESS, LH, HFX, GSW, GLW

--- OR ---

If you don't use this, then

- #define BULK_FLUXES — this will send Uwind, Vwind, Patm, RH, Tair, cloud, GSW, GLW

Related:

- #define EMINUSP — send EVAP and RAIN from WRF to ROMS
- #define CLOUDS — send cloud fraction from WRF to ROMS
- #define ATM_PRESS — send MSLP (Patm) from WRF to ROMS

These 3 options use SWAN wave data for computation of ocean surface stress in ROMS bulk_fluxes and in WRF myjsfc and mynn surface layer schemes :

- #define COARE_TAYLOR_YELLAND — wave enhanced roughness (swan to roms or wrf)
- #define COARE_OOST — wave enhanced roughness (swan to roms or wrf)
- #define DRENNAN — wave enhanced roughness (swan to roms or wrf)

- #define DRAGLIM_DAVIS — feature added to WRF and SWAN to limit the ocean roughness drag to be a maximum of 2.85E-3

USGS

COAWST specific cpp options

If you couple ROMS + WRF then you can select this for testing:

- #define SST_CONST — do not allow SST from ROMS to affect WRF

Wave Options

- #define UV_CONST — send vel = 0 from the ocn to wave model
- #define ZETA_CONST — send zeta = 0 from the ocn to wave model
- #define UV_KIRBY — compute "depth-avg" current based on Hwave to be sent from the ocn to the wav model for coupling
- #define WEC_MELLOR — radiation stress terms from Mellor 08
- #define WEC_VF — wave-current stresses from Uchiyama et al.
- #define WDISS_THORGUZA — wave dissipation from Thorton/Guza
- #define WDISS_CHURTHOR — wave dissipation from Church/Thorton
- #define WDISS_WAVEMOD — wave dissipation from a wave model
- #define WDISS_INWAVE — wave dissipation from a InWave model
- #define ROLLER_SVENDSEN — wave roller based on Svendsen
- #define ROLLER_MONO — wave roller for monochromatic waves
- #define ROLLER_RENIERS — wave roller based on Reniers
- #define BOTTOM_STREAMING — wave enhanced bottom streaming
- #define SURFACE_STREAMING — wave enhanced surface streaming

USGS

COAWST specific cpp options

Wave Options

#define CHARNOK	Charnok surface roughness from wind stress
#define CRAIG_BANNER	Craig and Banner wave breaking surface flux
--- or ---	
#define ZOS_HSIG	surface roughness from wave amplitude
#define TKE_WAVEDISS	wave breaking surface flux from wave amplitude

Vegetation Options

# define VEGETATION	Activate vegetation module
# define VEG_DRAG	Drag terms Luhar M. et.al (2011)
# define VEG_FLEX	Flexible vegetation terms
# define VEG_TURB	Turbulence terms, Uittenbogaard R. (2003)
# define VEG_SWAN_COUPLING	Exchange of VEG data btwn. ROMS and SWAN
# define VEG_STREAMING	Wave streaming effects
# define MARSH_WAVE_THRUST	Wave thrust on marshes, Tonelli, M. et al. (2010)

Tracer Advection Option

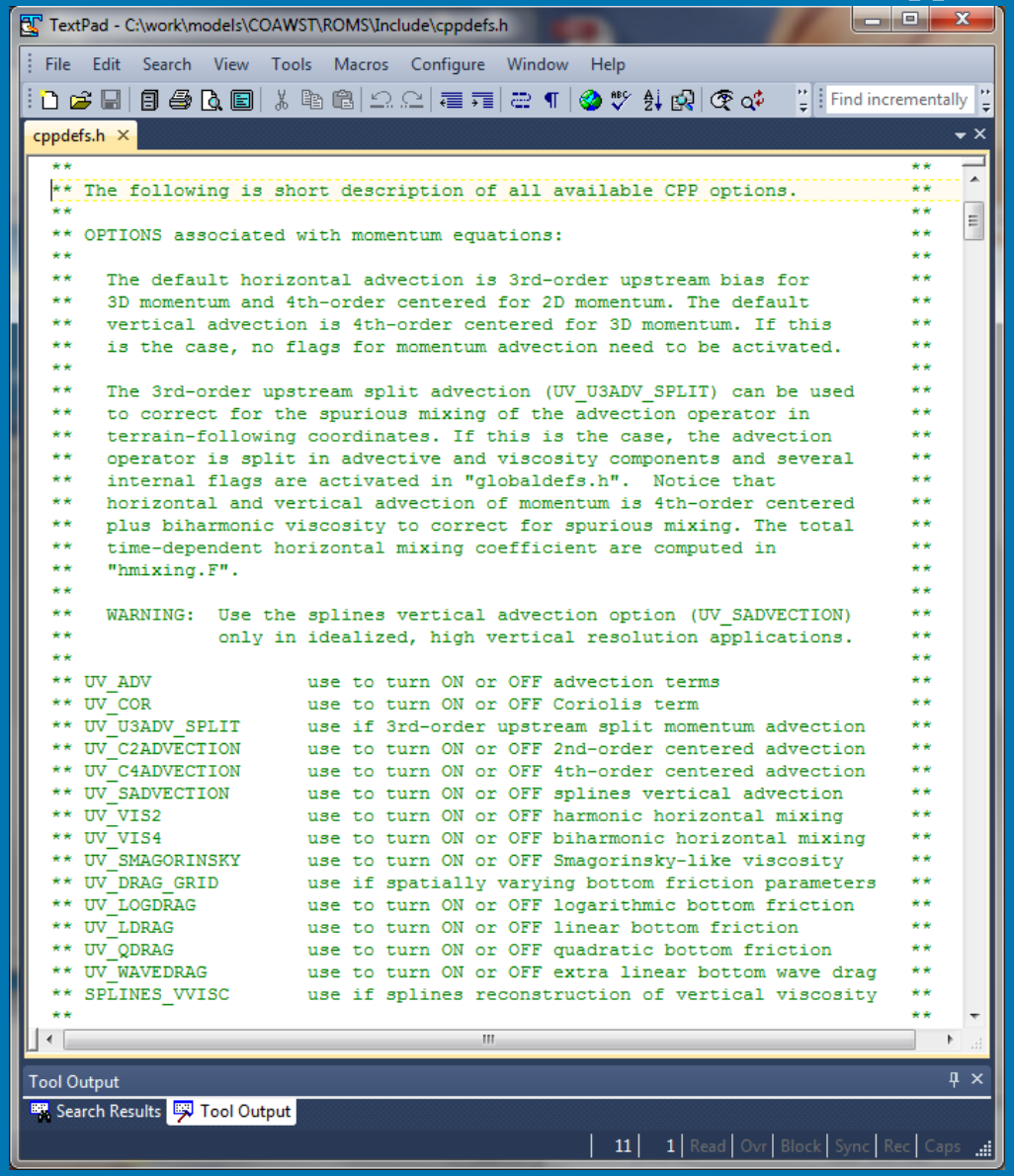
# define TS_HSIMT	Positive definite tracer advection (Wu and Zhu, OM 2010)

USGS

+ ROMS Specific CPP options …..

ROMS/Include/cppdefs.h

See cppdefs for a complete list of ROMS options



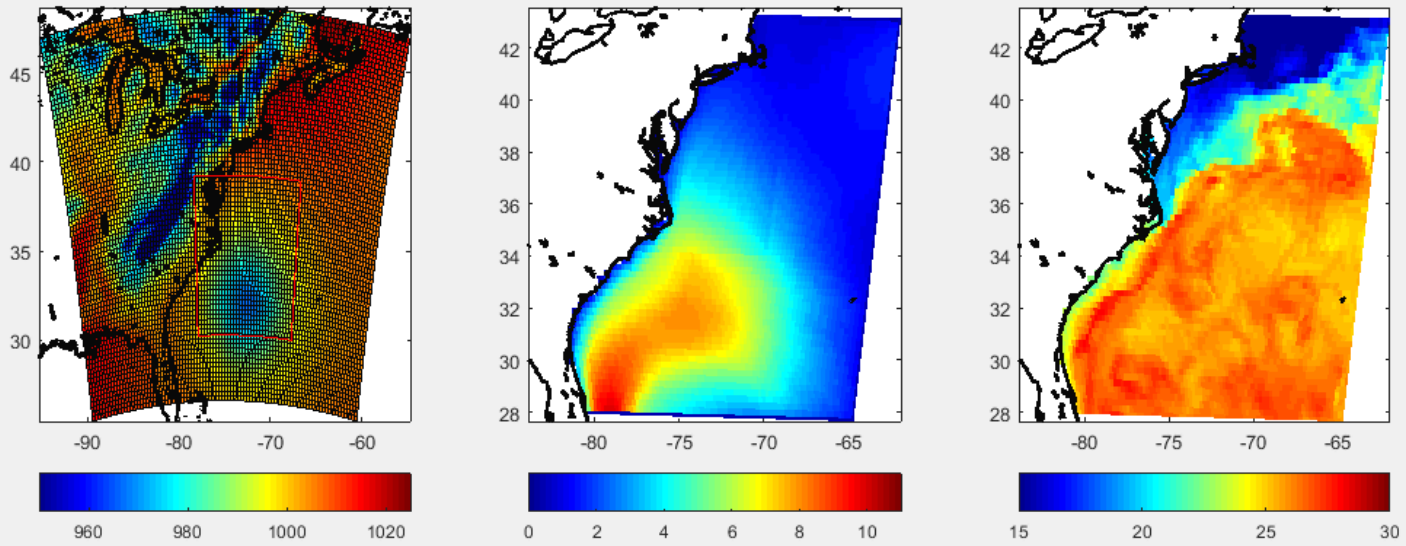
This list goes on for many many more pages …..

Projects/Sandy Application

WRF

SWAN

ROMS



PSFC (mb)

Hsig (m)

SST (C)

USGS

Steps to create ROMS application

1) parent grid
2) masking
3) bathymetry
4) child grid
5) 3D: BC's (u,v,temp,salt), init, and climatology
6) 2D: BC's (ubar, vbar, zeta) = tides
7) Surface forcing (heat and momentum fluxes)
8) sandy.h and ocean_sandy.in
9) coawst.bash
10) run it

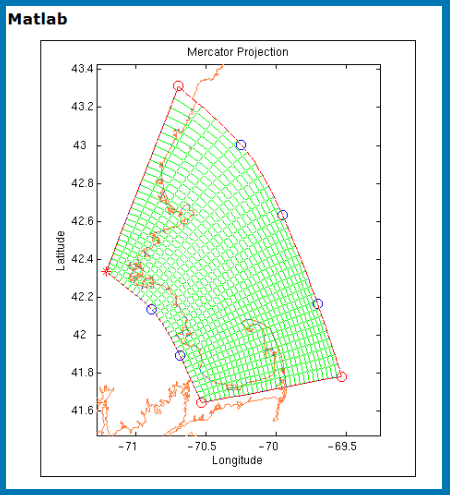
- Classroom tutorial will follow
Projects/Sandy/create_sandy_application.m

USGS

1) Grid generation tools

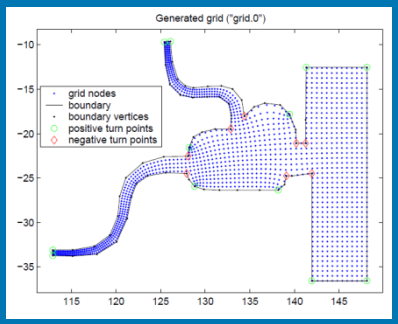
- **Seagrid - matlab**

http://woodshole.er.usgs.gov/operations/modeling/seagrid/
(needs unsupported
netcdf interface)

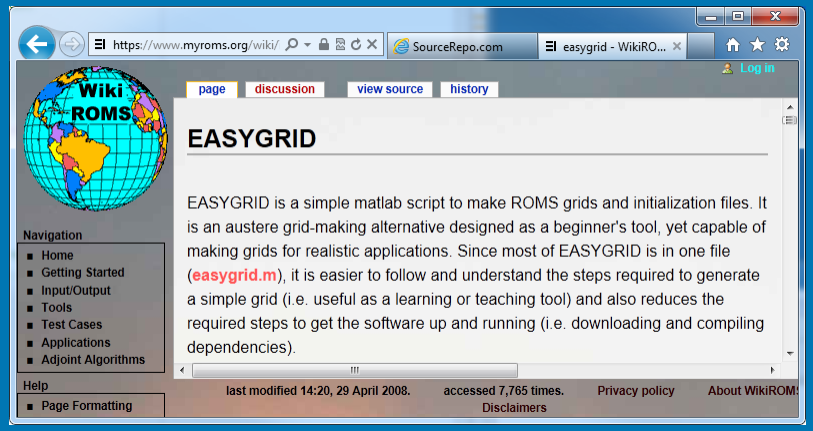


- **gridgen - command line**

http://code.google.com/p/gridgen-c/

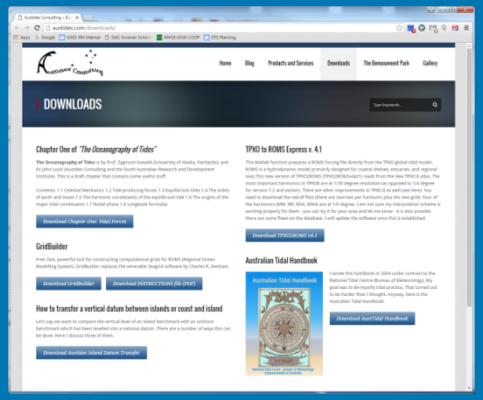


- **EASYGRID**



USGS

1) Grid generation tools



- **Austides**

http://austides.com/downloads/

- **COAWST/Tools/mfiles/mtools/wrf2roms _mw.m**

function wrf2roms_mw(theWRFFile, theROMSFile)
Generates a ROMS grid from a WRF grid.

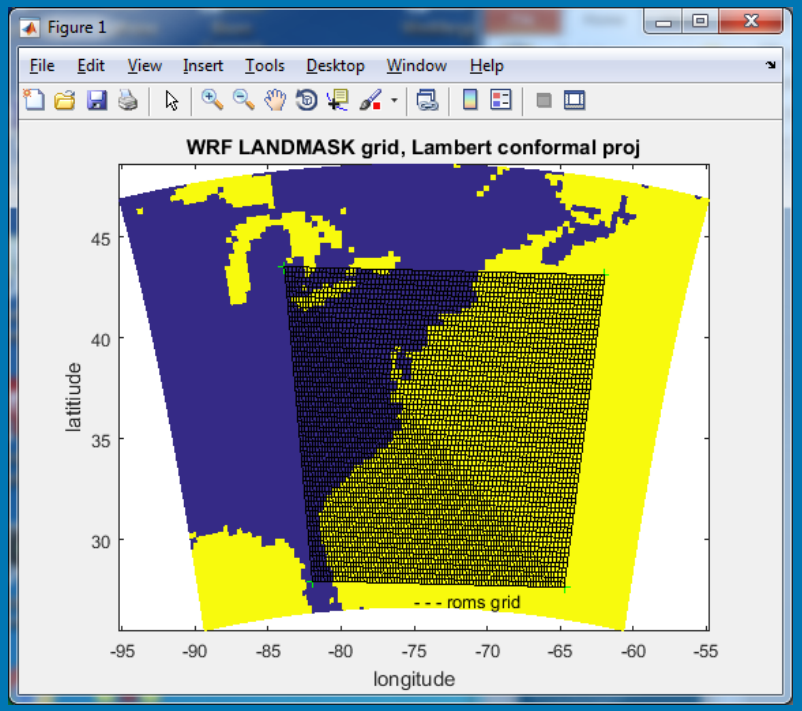
- **COAWST/Tools/mfiles/mtools/create_roms_xygrid.m**

intended for simple rectilinear grids

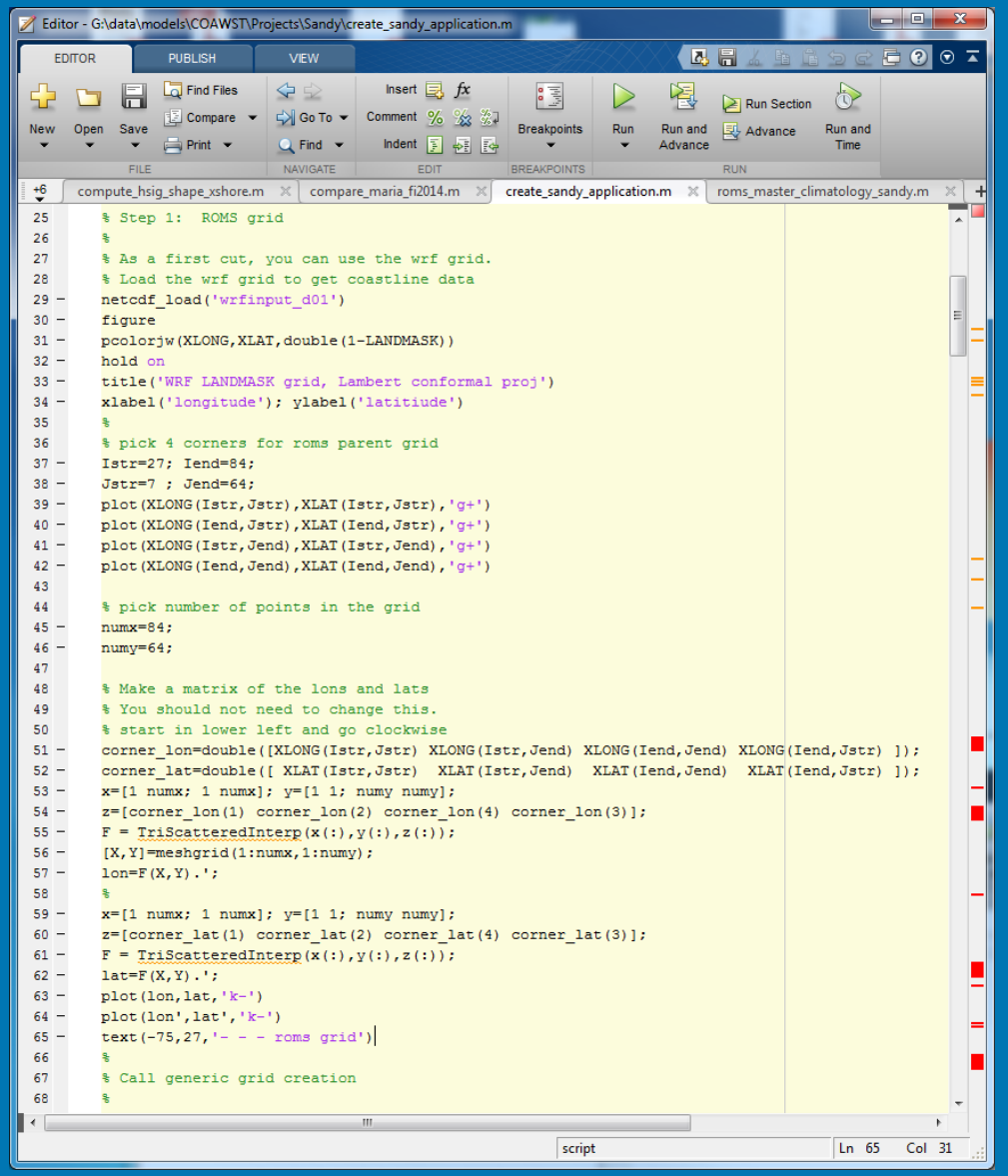
- **or any other method that you know**

USGS

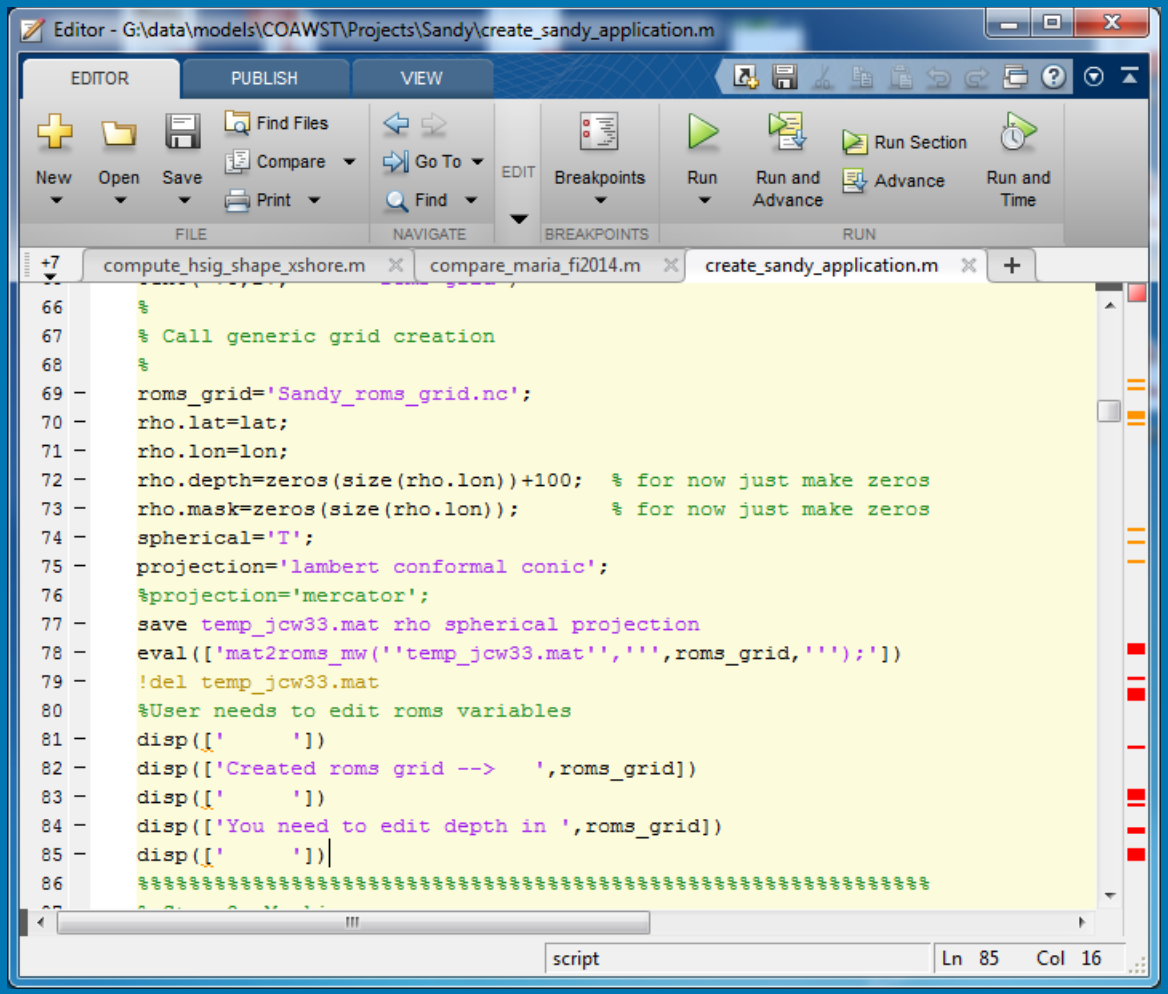
1) Grid generation tools – m file



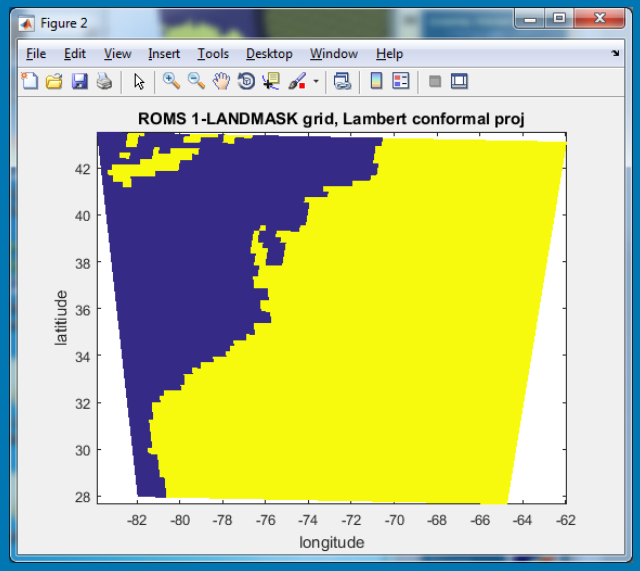
select 4 corners



1) Grid generation tools



2) masking – first base it on WRF



Editor - G:\data\models\COAWST\Projects\Sandy\create_sandy_application.m

EDITOR PUBLISH VIEW

New Open Save Find Files Compare Print Go To Find Insert Comment Indent Breakpoints Run Run and Advance Run Section Advance Run and Time

FILE NAVIGATE EDIT BREAKPOINTS RUN

plot_sfcr_wave_refract.m | compute_hsig_shape_xshore.m | compare_maria_fi2014.m | create_sandy_application.m

```
87  % Step 2: Masking
88  %
89  % base this on the WRF mask.
90  F = TriScatteredInterp(double(XLONG(:)),double(XLAT(:)), ...
91                         double(1-LANDMASK(:)),'nearest');
92  roms_mask=F(lon,lat);
93  figure
94  pcolorjw(lon,lat,roms_mask)
95  title('ROMS 1-LANDMASK grid, Lambert conformal proj')
96  xlabel('longitude'); ylabel('latitiude')
97  %
98  % compute mask on rho, u, v, and psi points.
99  water = double(roms_mask);
100 u_mask = water(1:end-1,:) & water(2:end,:);
101 v_mask= water(:,1:end-1) & water(:,2:end);
102 psi_mask= water(1:end-1,1:end-1) & water(1:end-1,2:end) & water(2:end,1:end-1) & water(2:end,2:end);
103 ncwrite('Sandy_roms_grid.nc','mask_rho',roms_mask);
104 ncwrite('Sandy_roms_grid.nc','mask_u',double(u_mask));
105 ncwrite('Sandy_roms_grid.nc','mask_v',double(v_mask));
106 ncwrite('Sandy_roms_grid.nc','mask_psi',double(psi_mask));
107 %
108 % use coastline tool to get coast:
109 % use GEODAS software http://www.ngdc.noaa.gov/mgg/geodas/geodas.html
110 % select lat bounds of 45 / 25 lon of -84 / -60
111 %
112 lon=coastline(:,1);
113 lat=coastline(:,2);
114 save coastline.mat lon lat
115 %
116 % To create better mask, use editmask with the
117 % grid file Sandy_roms_grid.nc and
118 % the coastline file coastline.mat
119 %
120 editmask
121 %
```

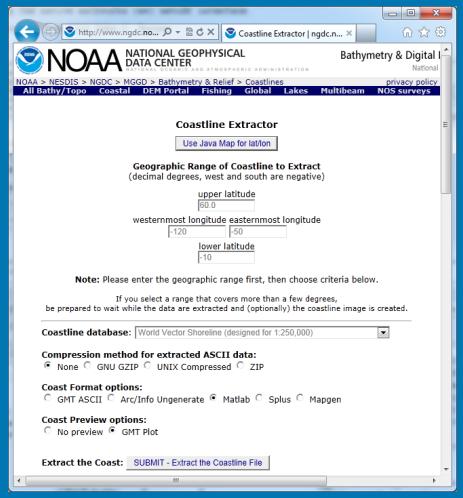
script Ln 99 Col 27

USGS

2) Masking – get coastline

May also need a coastline, can obtain this here:

http://www.ngdc.noaa.gov/mgg/coast/

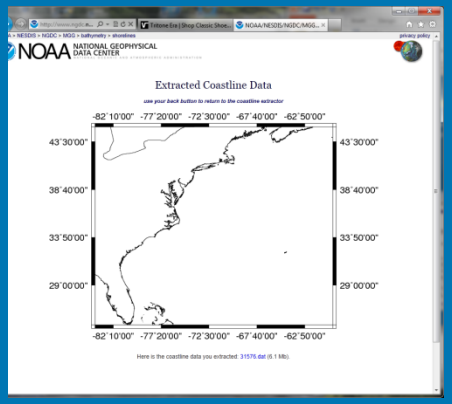


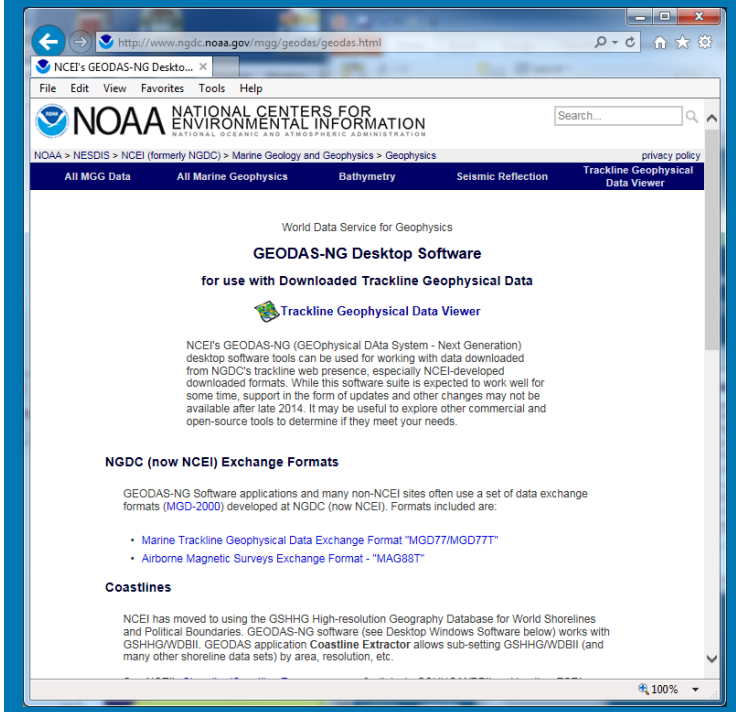
45

-84 -60

25

No longer active.





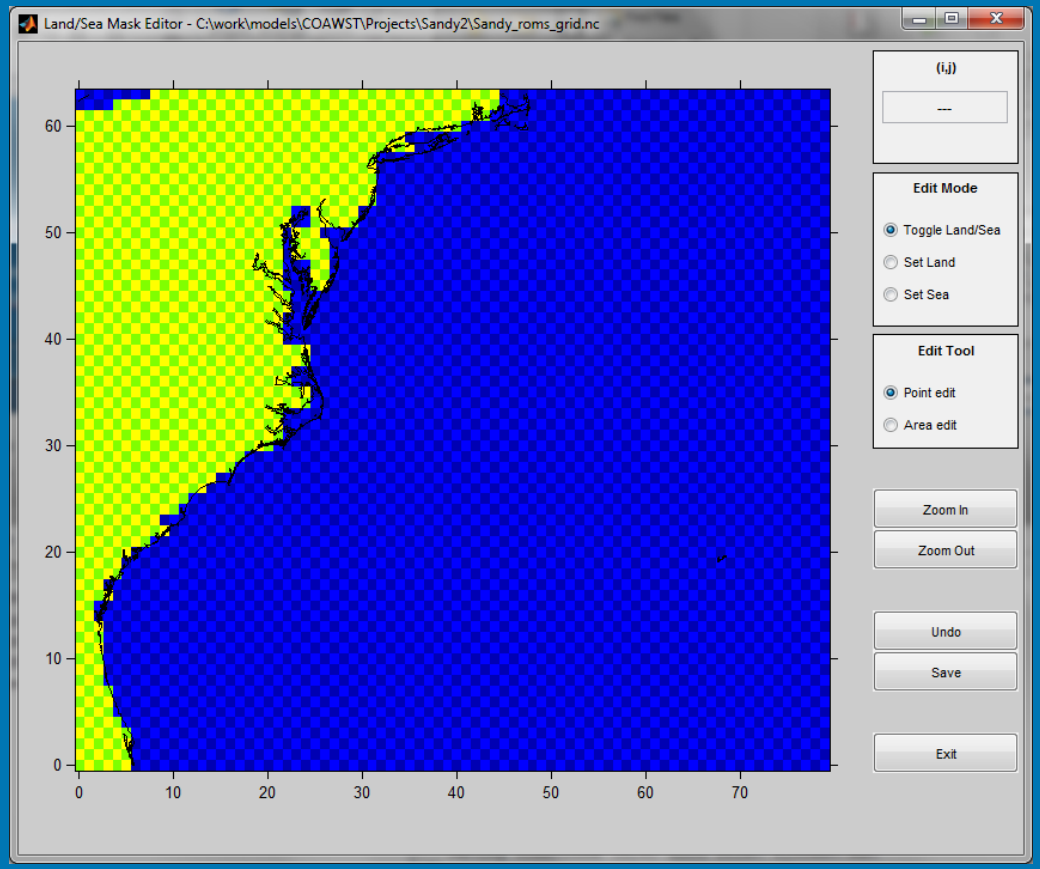
http://www.ngdc.noaa.gov/mgg/geodas/geodas.html

```
lon=coastline(:,1);
lat=coastline(:,2);
save coastline.mat lon lat
```

2) Masking

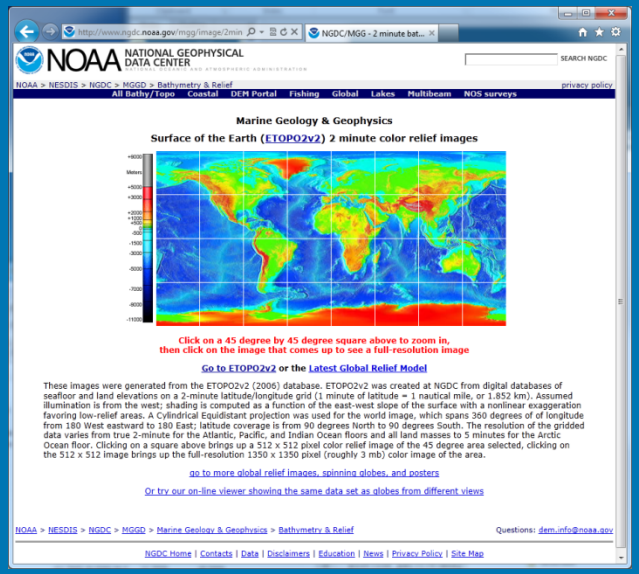
use COAWST/Tools/mfiles/mtools/editmask m file

(from Rutgers, but I changed it to use native matlab netcdf)
editmask('USeast_grd.nc','coastline.mat')

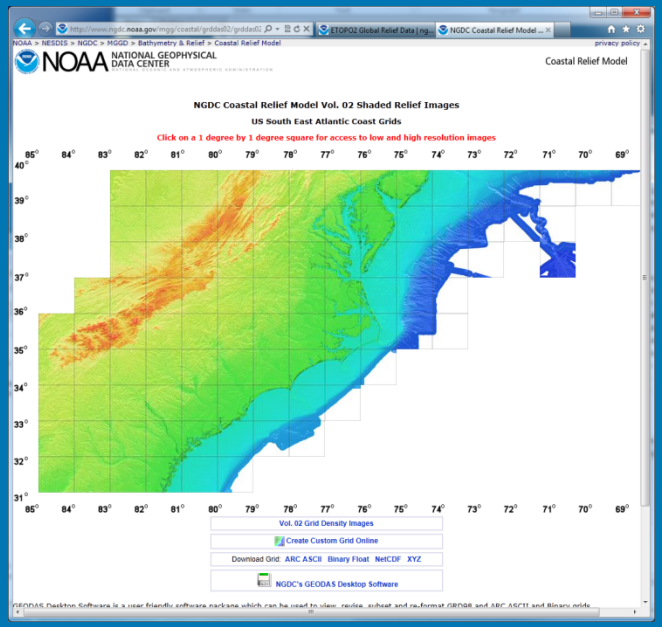


3) bathymetry

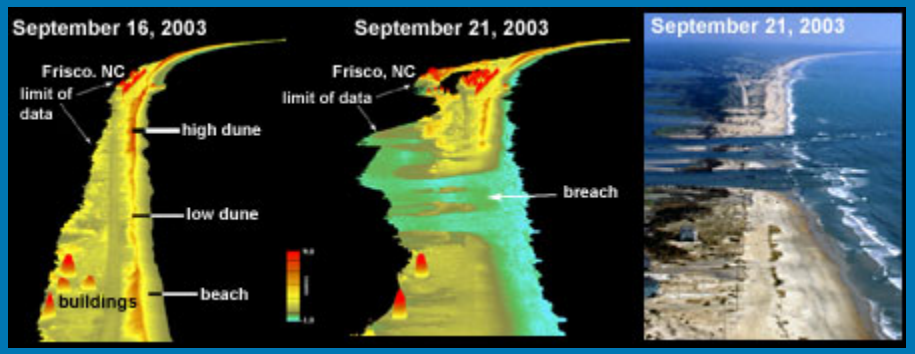
many sources



ETOPO2



Coastal Relief Model



LIDAR

USGS

3) bathymetry

Editor - G:\data\models\COAWST\Projects\Sandy\create_sandy_application.m

EDITOR | PUBLISH | VIEW

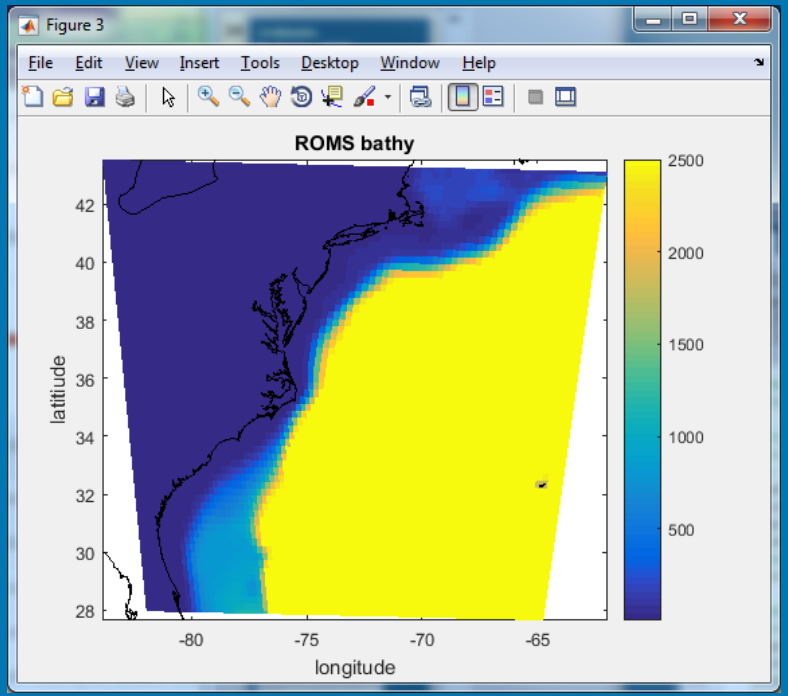
New | Open | Save | Find Files | Compare | Print | Go To | Find | Insert | Comment | Indent | Breakpoints | Run | Run and Advance | Run Section | Advance | Run and Time

FILE | NAVIGATE | EDIT | BREAKPOINTS | RUN

plot_sfcr_wave_refract.m | compute_hsig_shape_xshore.m | compare_maria_fi2014.m | create_sandy_application.m

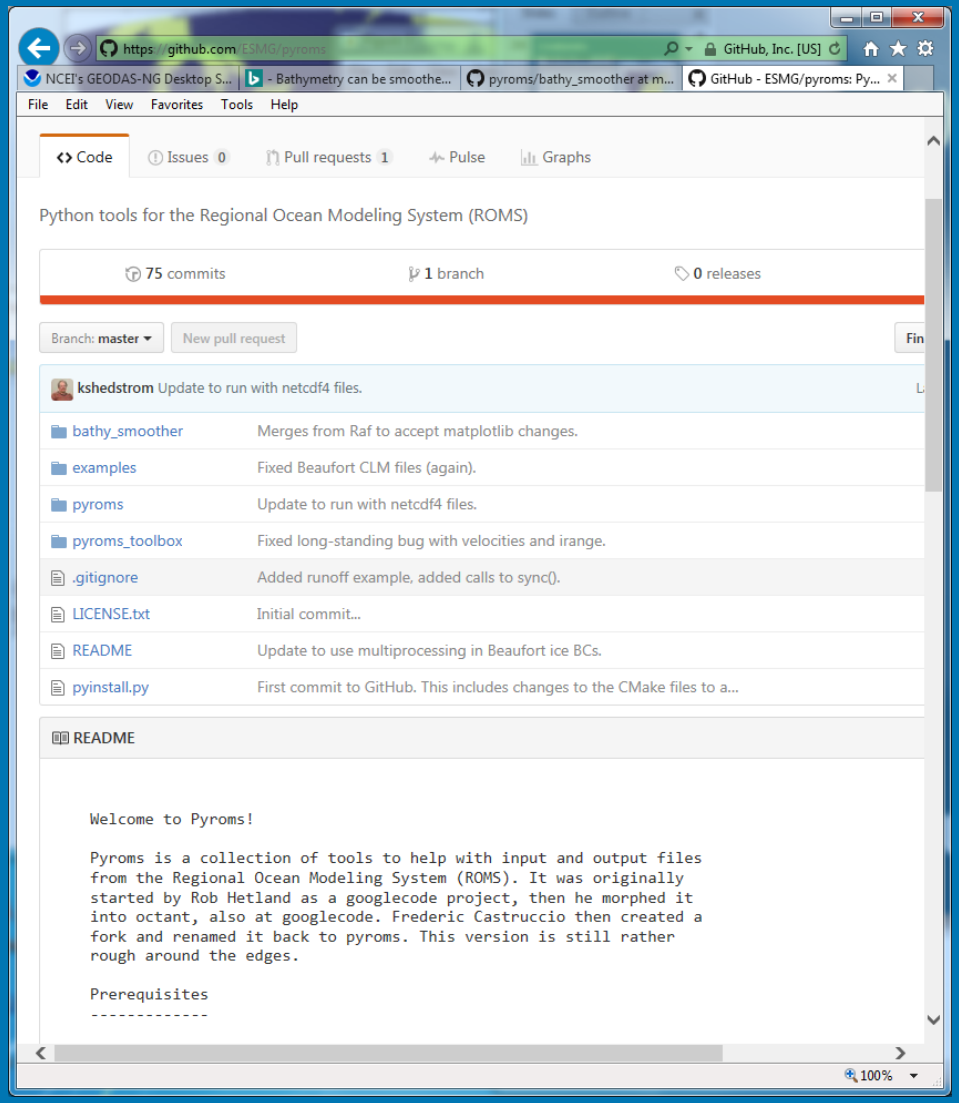
```
122  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
123  % Step 3: Bathymetry
124  %
125  % You can use a source like this:
126  % http://www.ngdc.noaa.gov/mgg/global/global.html
127  % i am using data from a local file
128  load USeast_bathy.mat
129  netcdf_load(roms_grid)
130  h=griddata(h_lon,h_lat,h_USeast,lon_rho,lat_rho);
131  h(isnan(h))=5;
132  %smooth h a little
133  h(2:end-1,2:end-1)=0.2*(h(1:end-2,2:end-1)+h(2:end-1,2:end-1)+h(3:end,2:end-1)+ ...
134                          h(2:end-1,1:end-2)+h(2:end-1,3:end));
135  % Bathymetry can be smoothed using
136  % http://drobilica.irb.hr/~mathieu/Bathymetry/index.html
137  figure
138  pcolorjw(lon_rho,lat_rho,h)
139  hold on
140  load coastline.mat
141  plot(lon,lat,'k')
142  caxis([5 2500]); colorbar
143  title('ROMS bathy')
144  xlabel('longitude'); ylabel('latitiude')
145  ncwrite(roms_grid,'h',h);
146  %
```

script | Ln 131 | Col 15



USGS

3) bathymetry



Bathy smoothing:

https://github.com/ESMG/pyroms

Grid Parameters

Beckman & Haidvogel number (1993)

$$r_{xo} = \max\left(\frac{\Delta h}{2\bar{h}}\right) = \max\left(\frac{|h_i - h_{i-1}|}{h_i + h_{i-1})}\right)$$

should be < 0.2 but can be fine up to ~ 0.4

determined only by smoothing

Haney number (1991)

$$r_{x1} = \max\left(\frac{z_{i,j,k} - z_{i-1,j,k} + z_{i,j,k-1} - z_{i-1,j,k-1}}{z_{i,j,k} + z_{i-1,j,k} - z_{i,j,k-1} - z_{i-1,j,k-1}}\right)$$

should be < 9 but can be fine up to ~ 16 in some cases

determined by smoothing AND vertical coordinate functions

If these numbers are too large, you will get large pressure gradient errors and Courant number violations and the model will typically blow up right away

4) Child grid

Editor - G:\data\models\COAWST\Projects\Sandy\create_sandy_application.m

EDITOR | PUBLISH | VIEW

New | Open | Save | Find Files | Compare | Print | Go To | Find | Insert | Comment | Indent | Breakpoints | Run | Run and Advance | Run Section | Advance | Run and Time

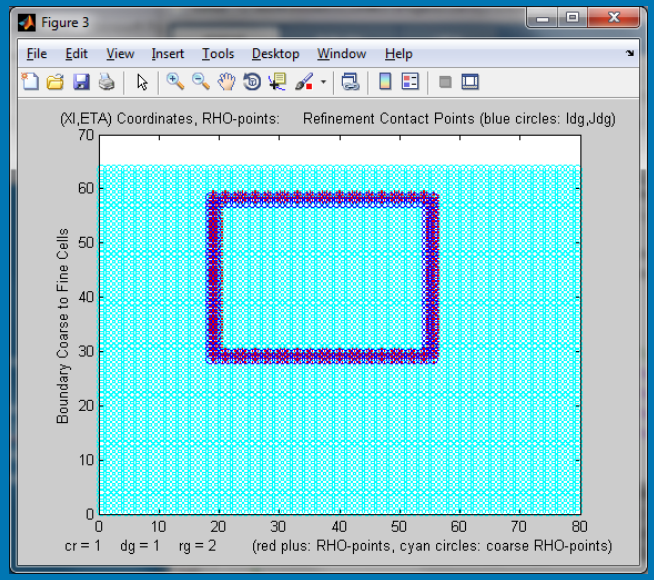
FILE | NAVIGATE | EDIT | BREAKPOINTS | RUN

compute_hsig_shape_xshore.m | compare_maria_fi2014.m | create_sandy_application.m

```
146     %
147     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
148     % Step 4: roms child grid
149     %
150     % plot wrf d01 and d02 grids
151 -   netcdf_load('wrfinput_d01')
152 -   figure
153 -   pcolorjw(XLONG,XLAT,double(1-LANDMASK))
154 -   hold on
155 -   netcdf_load('wrfinput_d02')
156 -   pcolorjw(XLONG,XLAT,double(1-LANDMASK))
157 -   plot(XLONG(1,:),XLAT(1,:),'r'); plot(XLONG(end,:),XLAT(end,:),'r')
158 -   plot(XLONG(:,1),XLAT(:,1),'r'); plot(XLONG(:,end),XLAT(:,end),'r')
159     % plot roms parent grid
160 -   plot(lon_rho(1,:),lat_rho(1,:),'k'); plot(lon_rho(end,:),lat_rho(end,:),'k')
161 -   plot(lon_rho(:,1),lat_rho(:,1),'k'); plot(lon_rho(:,end),lat_rho(:,end),'k')
162 -   text(-75,29,'roms parent grid')
163 -   text(-77,27,'wrf parent grid')
164 -   text(-77.2,34,'wrf child grid')
165 -   title('LANDMASKS')
166 -   xlabel('longitude'); ylabel('latitiude')
167     %
168     % Select child indices
169 -   Istr=22; Iend=60; Jstr=26; Jend=54;
170     %now make some plots and call the coarse-> fine
171 -   plot(lon_rho(Istr,Jstr),lat_rho(Istr,Jstr),'m+')
172 -   plot(lon_rho(Istr,Jend),lat_rho(Istr,Jend),'m+')
173 -   plot(lon_rho(Iend,Jstr),lat_rho(Iend,Jstr),'m+')
174 -   plot(lon_rho(Iend,Jend),lat_rho(Iend,Jend),'m+')
175 -   ref_ratio=3;
176 -   roms_child_grid='Sandy_roms_grid_ref3.nc';
177 -   F=coarse2fine('Sandy_roms_grid.nc','Sandy_roms_grid_ref3.nc', ...
178                   ref_ratio,Istr,Iend,Jstr,Jend);
179 -   Gnames={'Sandy_roms_grid.nc','Sandy_roms_grid_ref3.nc'};
180 -   [S,G]=contact(Gnames,'Sandy_roms_contact.nc');
```

script | Ln 177 | Col 66

```
ref_ratio=3;
roms_child_grid='Sandy_roms_grid_ref3.nc';

F=coarse2fine('Sandy_roms_grid.nc','Sandy_roms_grid_ref3.nc', ...
              ref_ratio,Istr,Iend,Jstr,Jend);
Gnames={'Sandy_roms_grid.nc','Sandy_roms_grid_ref3.nc'};
[S,G]=contact(Gnames,'Sandy_roms_contact.nc');
```



4) Child grid- bathy + masking

Editor - G:\data\models\COAWST\Projects\Sandy\create_sandy_application.m

EDITOR | PUBLISH | VIEW

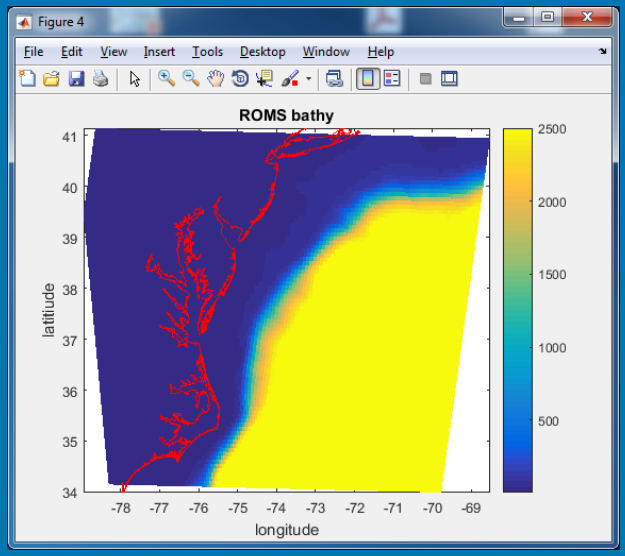
New | Open | Save | Find Files | Compare | Print | Go To | Find | Insert | Comment | Indent | Breakpoints | Run | Run and Advance | Run Section | Advance | Run and Time

FILE | NAVIGATE | EDIT | BREAKPOINTS | RUN

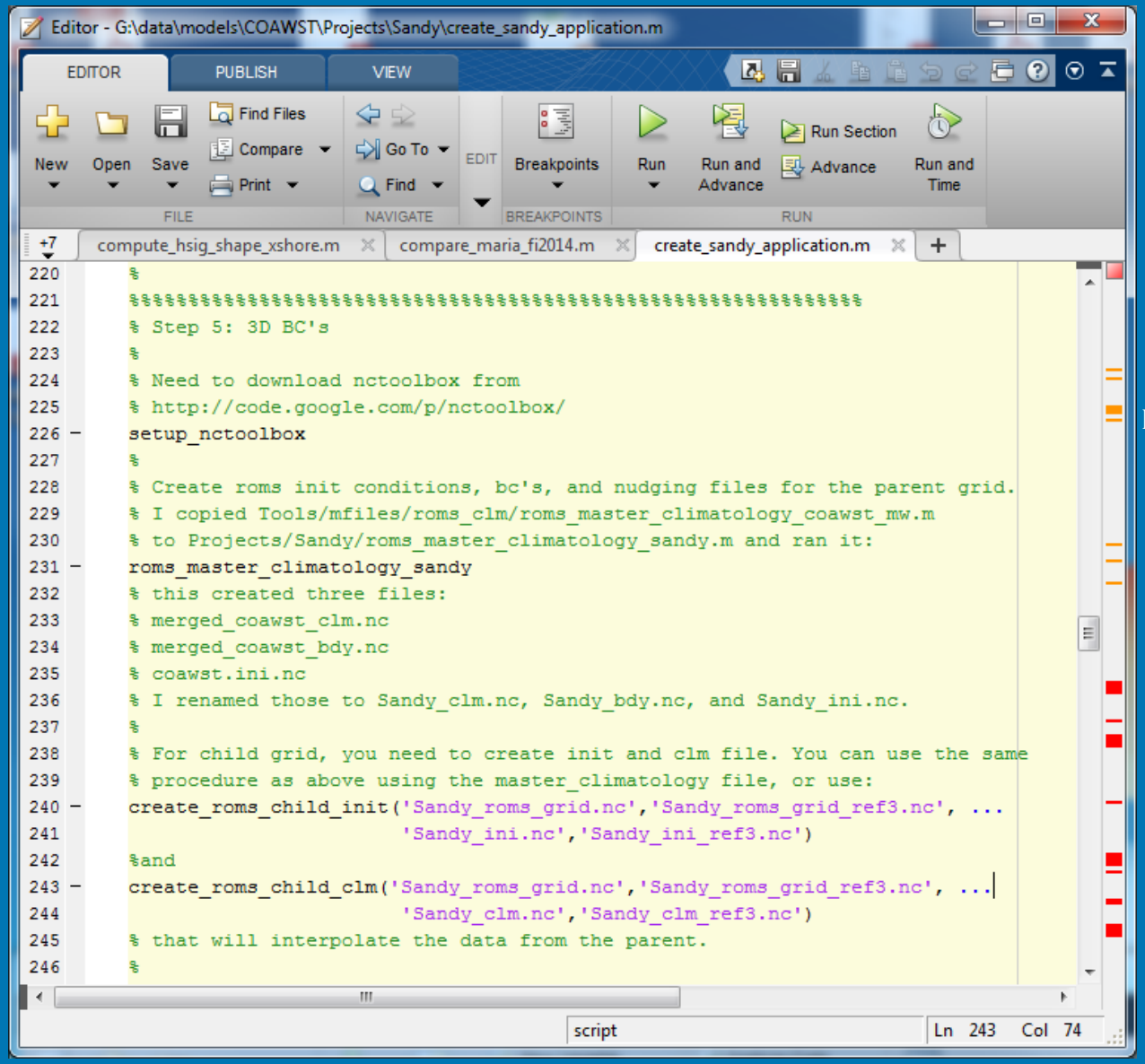
plot_sfcr_wave_refract.m | compute_hsig_shape_xshore.m | compare_maria_fi2014.m | create_sandy_application.m

```
182     % Recompute h.
183     %
184 -   netcdf_load('Sandy_roms_grid_ref3.nc')
185 -   load USeast_bathy.mat
186 -   h=griddata(h_lon,h_lat,h_USeast,lon_rho,lat_rho);
187 -   h(isnan(h))=5;
188 -   h=h;
189 -   h(2:end-1,2:end-1)=0.2*(h(1:end-2,2:end-1)+h(2:end-1,2:end-1)+h(3:end,2:end-1)+ ...
190                             h(2:end-1,1:end-2)+h(2:end-1,3:end));
191 -   figure
192 -   pcolorjw(lon_rho,lat_rho,h)
193 -   hold on
194 -   plot(lon,lat,'r')
195 -   caxis([5 2500]); colorbar
196 -   title('ROMS bathy')
197 -   xlabel('longitude'); ylabel('latitiude')
198 -   ncwrite('Sandy_roms_grid_ref3.nc','h',h);
199     %
200     % recompute child mask based on WRF mask
201     %
202 -   netcdf_load('wrfinput_d02');
203 -   F = TriScatteredInterp(double(XLONG(:)),double(XLAT(:)), ...
204                            double(1-LANDMASK(:)),'nearest');
205 -   roms_mask=F(lon_rho,lat_rho);
206 -   figure
207 -   pcolorjw(lon_rho,lat_rho,roms_mask)
208 -   title('ROMS child mask')
209 -   xlabel('longitude'); ylabel('latitiude')
210 -   hold on
211 -   plot(lon,lat,'r')
212 -   water = double(roms_mask);
213 -   u_mask = water(1:end-1,:) & water(2:end,:);
214 -   v_mask= water(:,1:end-1) & water(:,2:end);
215 -   psi_mask= water(1:end-1,1:end-1) & water(1:end-1,2:end) & water(2:end,1:end-1) & water(2:end,2:end);
216 -   ncwrite('Sandy_roms_grid_ref3.nc','mask_rho',roms_mask);
217 -   ncwrite('Sandy_roms_grid_ref3.nc','mask_u',double(u_mask));
218 -   ncwrite('Sandy_roms_grid_ref3.nc','mask_v',double(v_mask));
219 -   ncwrite('Sandy_roms_grid_ref3.nc','mask_psi',double(psi_mask));
220     %
```

script | Ln 201 Col 1



5) 3D BC's (u,v,temp,salt), init, and clima.

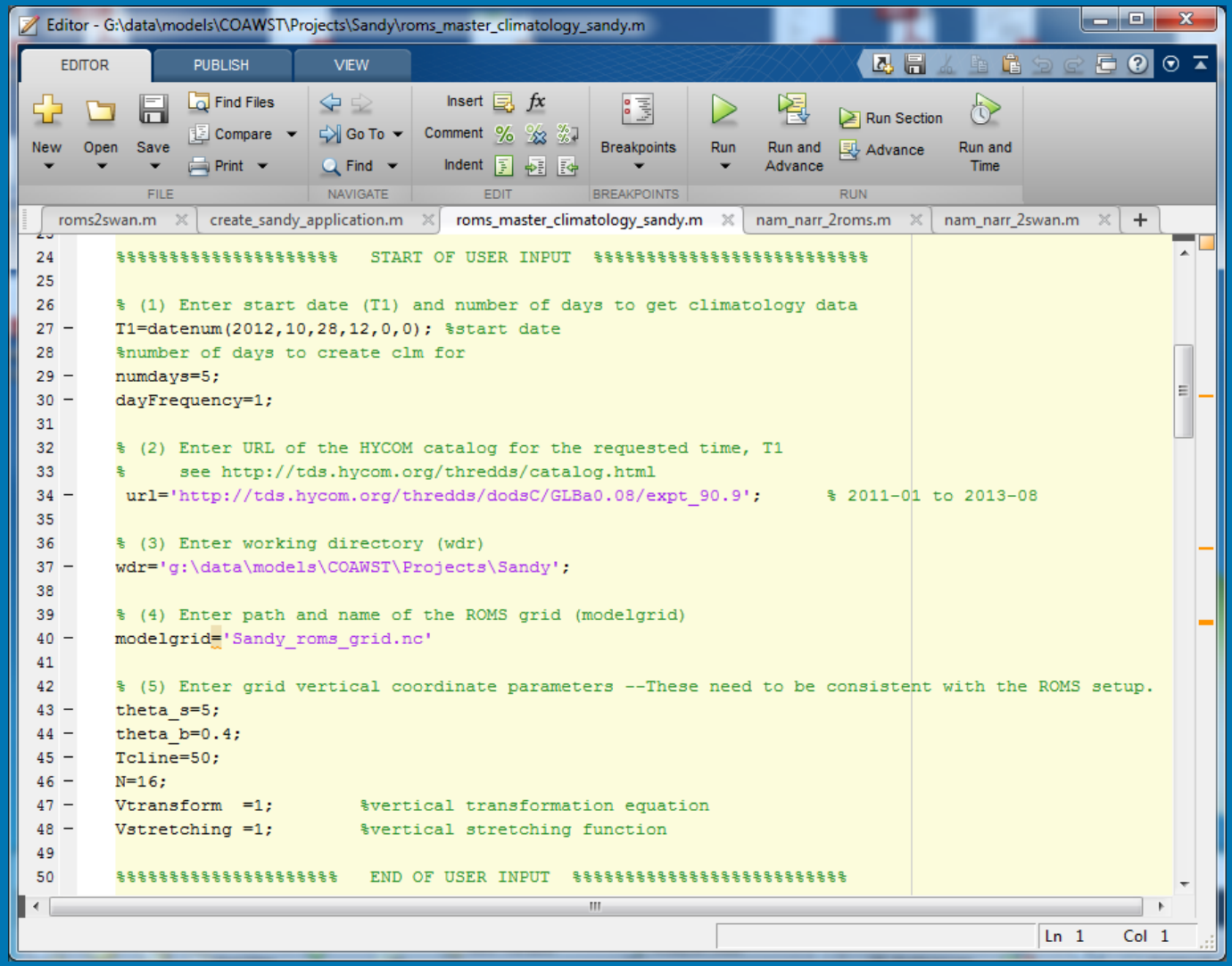


Needs nctoolbox to access data via thredds server:

https://github.com/nctoolbox/nctoolbox

5) 3D BC's (u,v,temp,salt), init, and clima.

Projects/Sandy/roms_master_climatology_sandy.m



This works for multiple days.

5) 3D BC's (u,v,temp,salt), init, and clima.

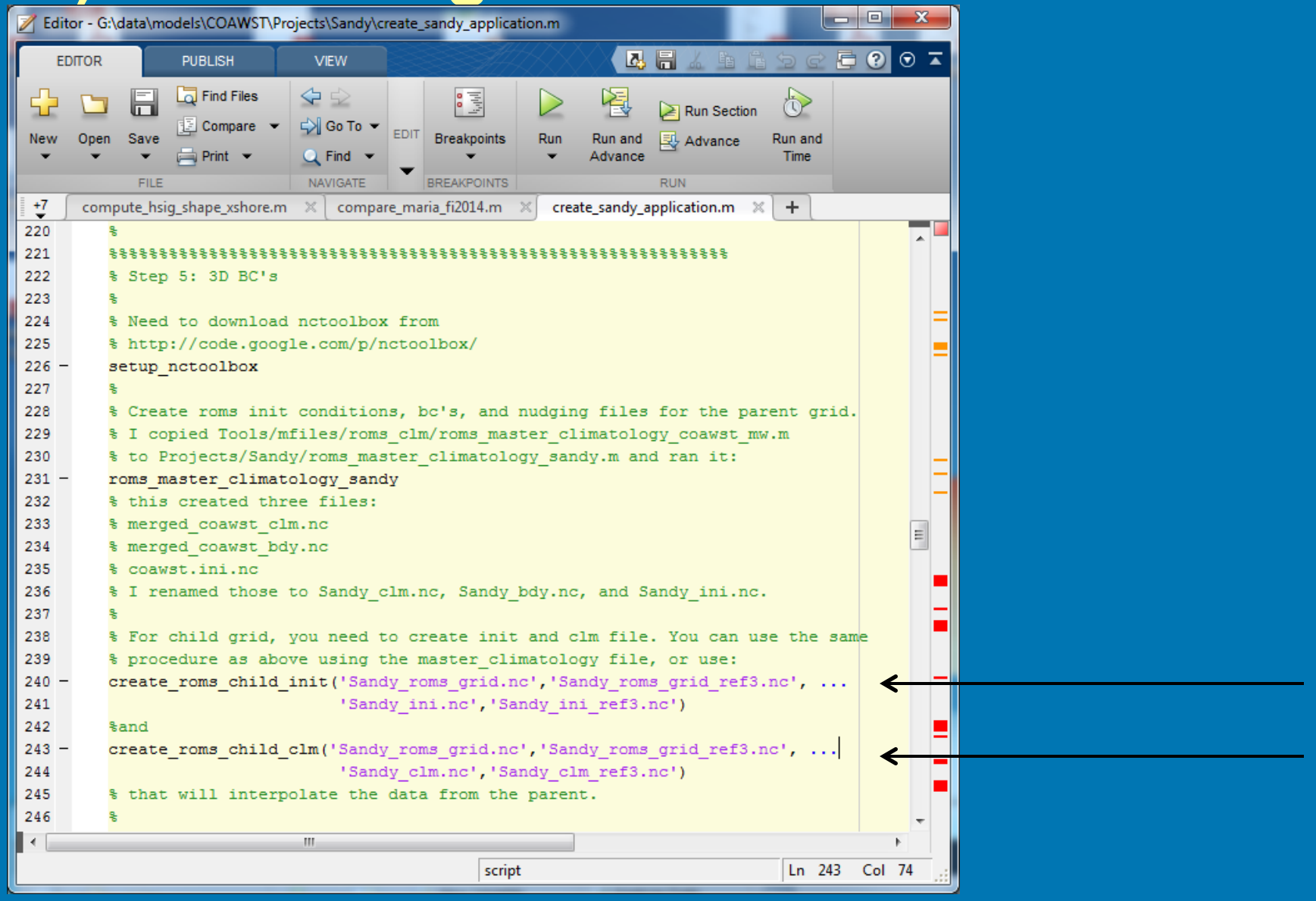
% I copied Tools/mfiles/roms_clm/roms_master_climatology_coawst_mw.m
% to Projects/Sandy/roms_master_climatology_sandy.m and ran it:

roms_master_climatology_sandy

% this created three files:
% merged_coawst_clm.nc
% merged_coawst_bdy.nc
% coawst.ini.nc
% I renamed those to Sandy_clm.nc, Sandy_bdy.nc, and Sandy_ini.nc.
%

USGS

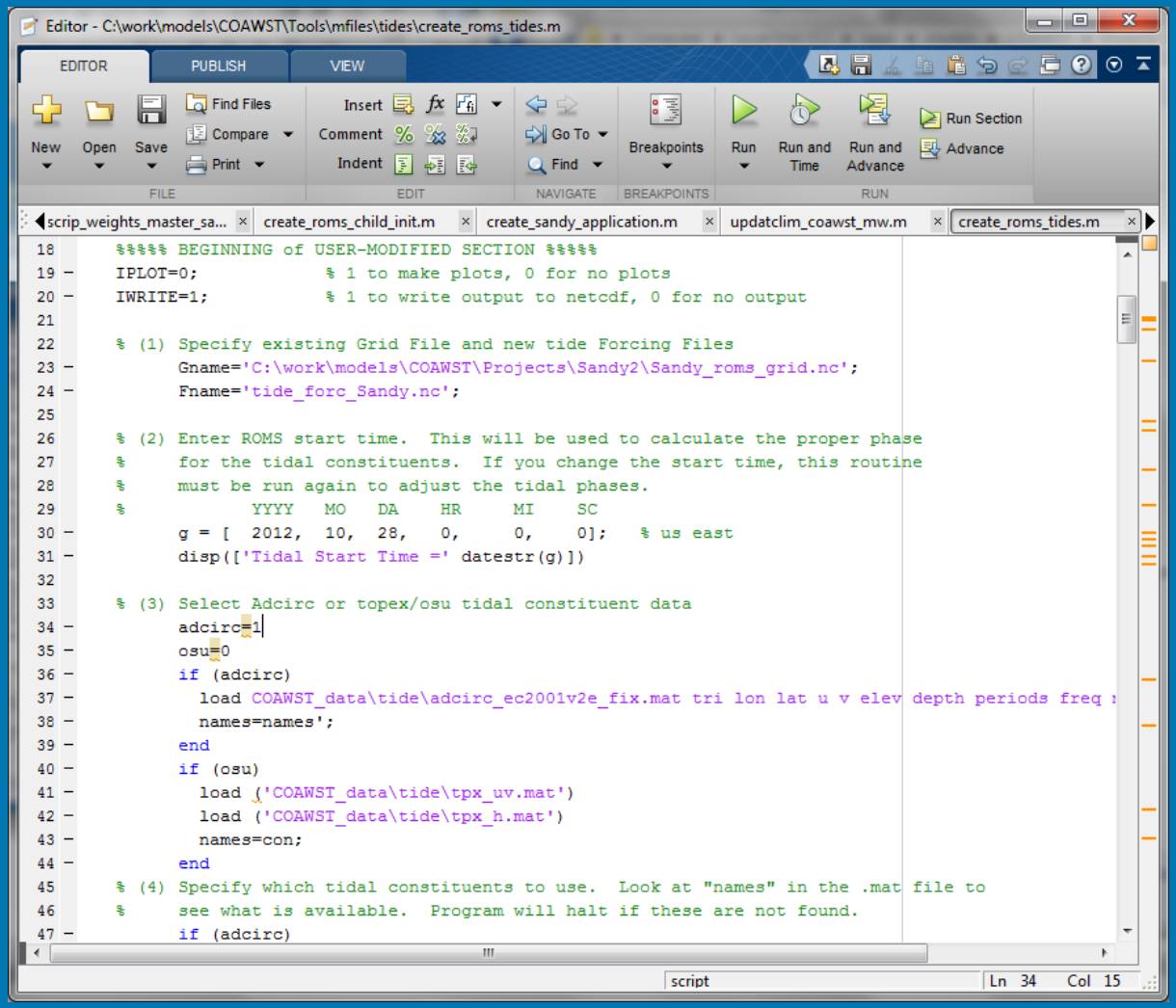
5) For child grid need init and clim.



6) 2D: BC's (ubar, vbar, zeta) = tides

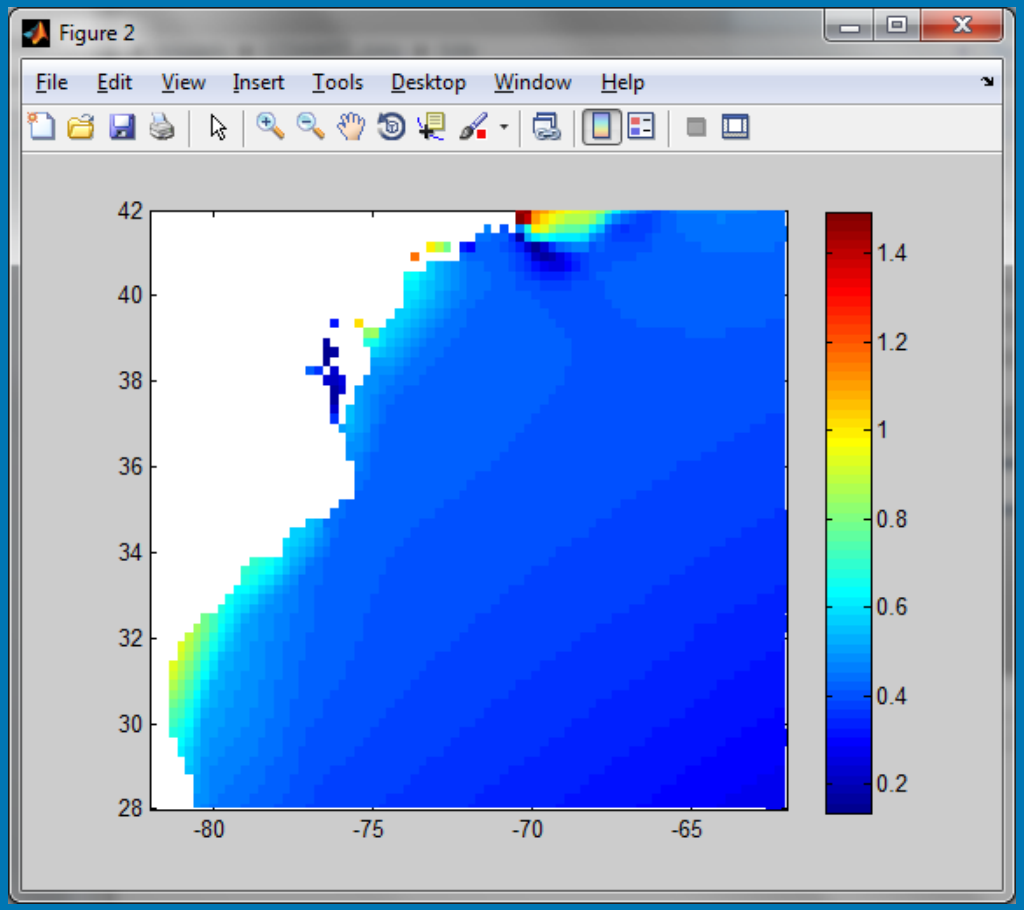
6) 2D: BC's (ubar, vbar, zeta) = tides

1) Get the tidal data at
svn checkout https://coawstmodel.sourcerepo.com/coawstmodel/data .
2) edit Tools/mfiles/tides/create_roms_tides



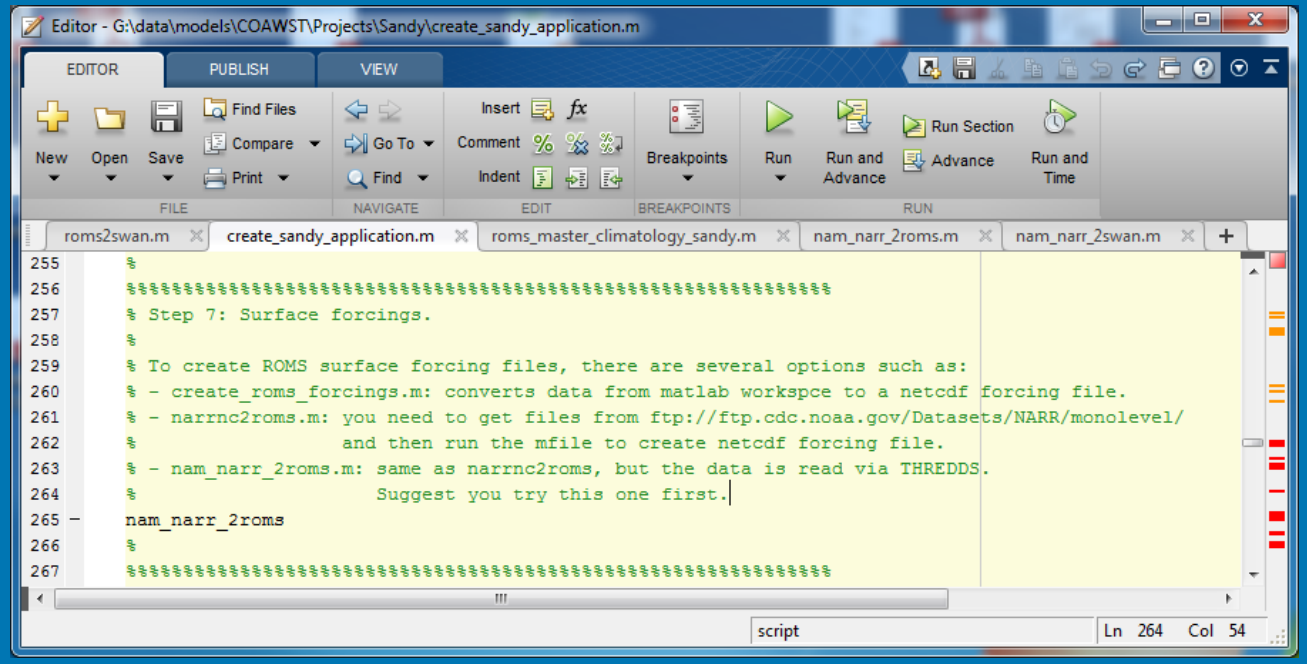
6) 2D: BC's (ubar, vbar, zeta) = tides

```
netcdf_load('Sandy_roms_grid.nc')
pcolorjw(lon_rho,lat_rho,squeeze(tide_Eamp(:,:,1)))
```



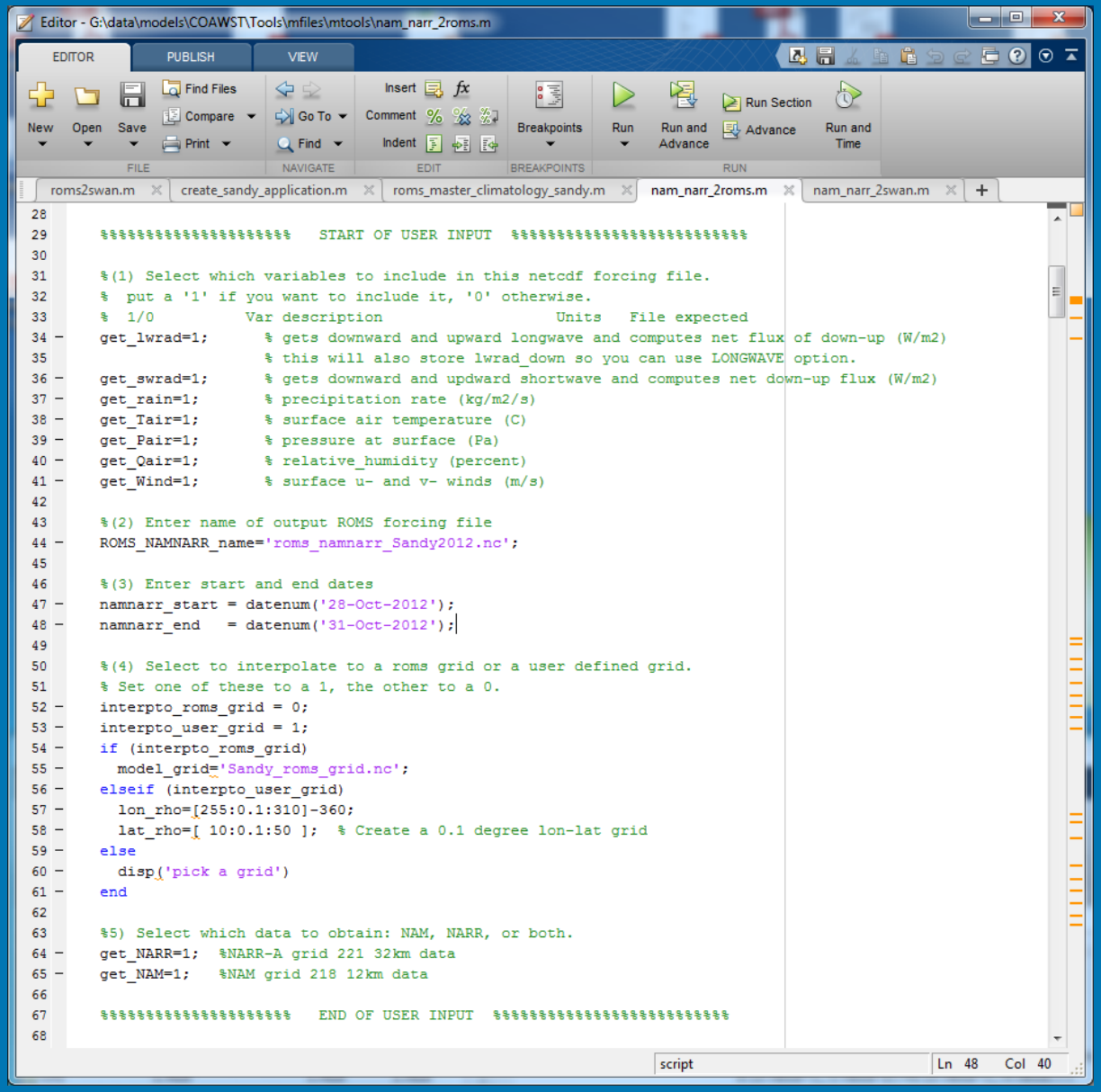
M2 tidal amplitude

7) Surface forcings



Many options. We will use nam_narr_2roms.m

7) Surface forcings - nam_narr_2roms.m



COAWST/Tools/mfiles/mtools/
nam_narr_2roms
creates netcdf forcing files for ROMS.

Many others tools available.

8) sandy.h

TextPad - G:\data\models\COAWST\Projects\Sandy\sandy.h *

File Edit Search View Tools Macros Configure Window Help

Find incrementally

sandy.h *

```
/*
** svn $Id: sandy.h 25 2007-04-09 23:43:58Z jcwarner $
******************************************************************
** Copyright (c) 2002-2007 The ROMS/TOMS Group
**   Licensed under a MIT/X style license
**   See License_ROMS.txt
******************************************************************
**
** Options for Sandy Test.
**
** Application flag:   SANDY
*/

#define ROMS_MODEL
#define NESTING
#undef  WRF_MODEL
#undef  SWAN_MODEL
#undef  MCT_LIB
#undef  MCT_INTERP_OC2AT
#undef  MCT_INTERP_WV2AT
#undef  MCT_INTERP_OC2WV

#if defined WRF_MODEL && defined SWAN_MODEL
# define DRAGLIM_DAVIS
# define COARE_TAYLOR_YELLAND
#endif

#ifdef ROMS_MODEL
/* Physics + numerics */
# define UV_ADV
# define UV_COR
# define UV_VIS2
# define MIX_S_UV
# undef  TS_FIXED
# define TS_U3HADVECTION
# define TS_C4VADVECTION
# undef  TS_MPDATA

# undef SSW_BBL
# ifdef SSW_BBL
#  define SSW_CALC_ZNOT
# /*define ANA_SEDIMENT*/
# else
#  define UV_LOGDRAG
# endif
# if !defined SWAN_MODEL && defined SSW_BBL
#  define ANA_WWAVE
# endif

# define DJ_GRADPS
# define TS_DIF2
# define MIX_GEO_TS
# define CURVGRID

# define SALINITY
# define SOLVE3D
# define SPLINES_VDIFF
# define SPLINES_VVISC
# define AVERAGES
# define NONLIN_EOS
```

Tool Output

22 1 Read Ovr Block Sync Rec Caps

TextPad - G:\data\models\COAWST\Projects\Sandy\sandy.h

File Edit Search View Tools Macros Configure Window Help

Find incrementally

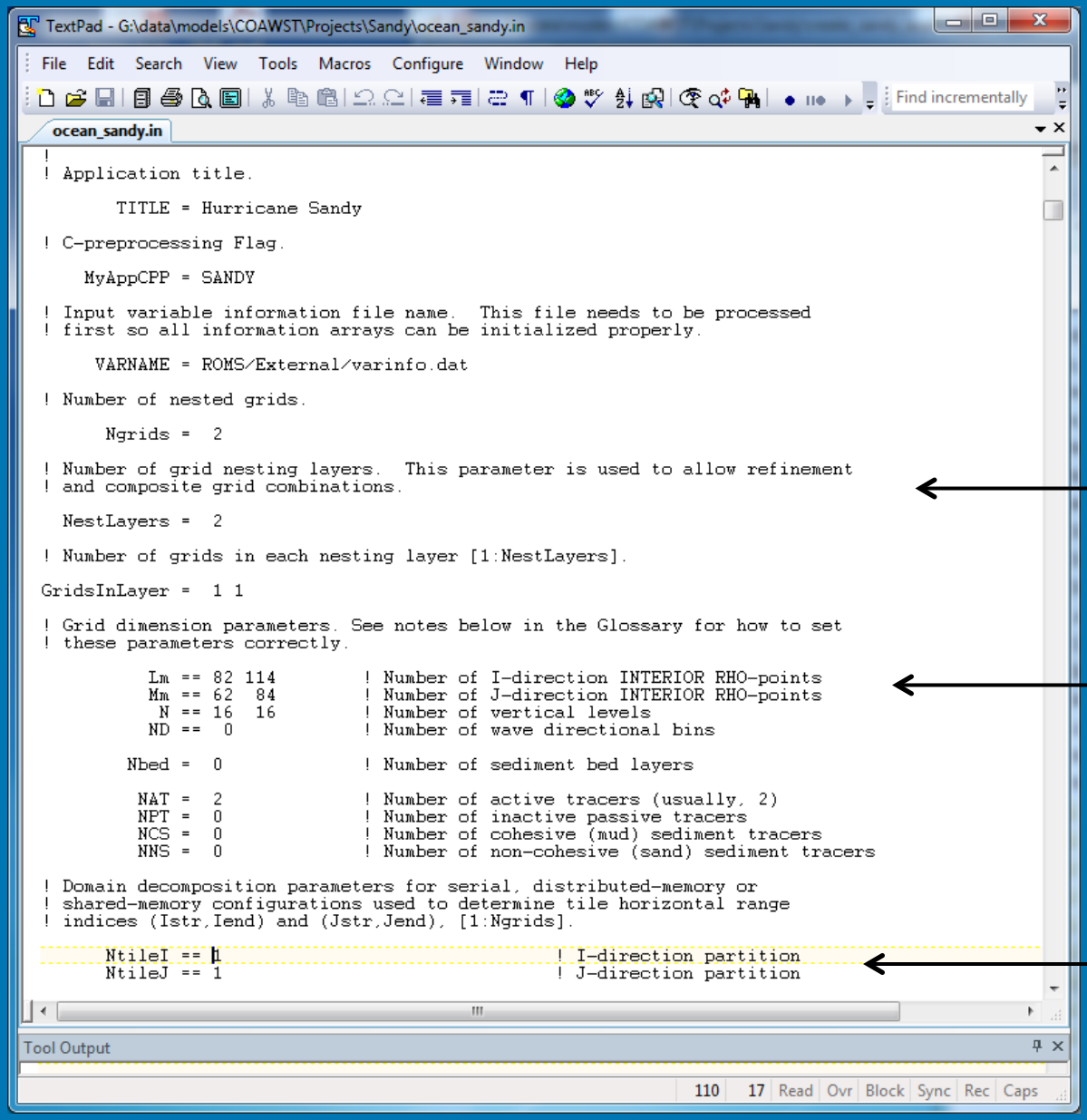
sandy.h

```
/* Grid and Initial */
# define MASKING

/* Forcing */
# ifdef WRF_MODEL
#  undef  BULK_FLUXES
#  define ATM2OCN_FLUXES
#  define ANA_SSFLUX
#  undef  LONGWAVE_OUT
# else
#  define BULK_FLUXES
# endif
# define ATM_PRESS
# define ANA_BTFLUX
# define ANA_BSFLUX
# define ANA_BPFLUX
# define ANA_SPFLUX
# define ANA_SRFLUX
# define EMINUSP
# define SOLAR_SOURCE

/* Turbulence closure */
# define GLS_MIXING
# undef  MY25_MIXING
# define AKLIMIT

# if defined GLS_MIXING || defined MY25_MIXING
#  define KANTHA_CLAYSON
#  define N2S2_HORAVG
#  define RI_SPLINES
# endif

# define SSH_TIDES
# define UV_TIDES
# define RAMP_TIDES
# define ANA_FSOBC
# define ANA_M2OBC

/* Output */
# undef DIAGNOSTICS_UV
# undef DIAGNOSTICS_TS
#endif
```

Tool Output

46 21 Read Ovr Block Sync Rec Caps

8) sandy_ocean.in



Nesting params

Grid sizes

NtileI NtileJ

8) sandy_ocean.in

TextPad - G:\data\models\COAWST\Projects\Sandy\ocean_sandy.in

File Edit Search View Tools Macros Configure Window Help

ocean_sandy.in

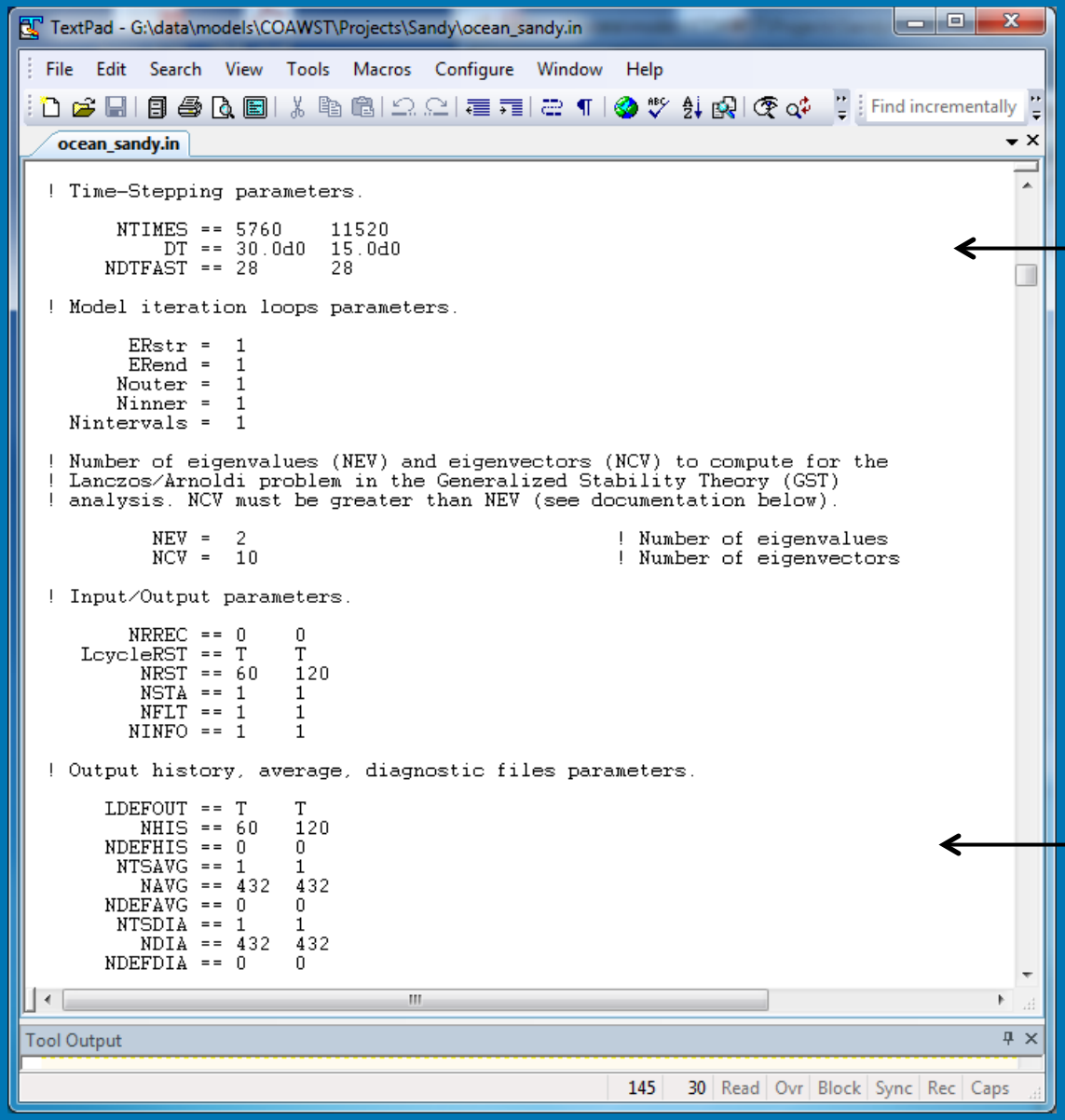
```
!
!   Keyword    Lateral Boundary Condition Type
!
!   Cha        Chapman_implicit (free-surface)
!   Che        Chapman_explicit (free-surface)
!   Cla        Clamped
!   Clo        Closed
!   Fla        Flather (2D momentum)                    _____N_____      j=Mm
!   Gra        Gradient                                |     4     |
!   Nes        Nested (refinement)                     |           |
!   Nud        Nudging                              1 W            E 3
!   Per        Periodic                                |           |
!   Rad        Radiation                               |_____S_____|
!   Red        Reduced Physics (2D momentum)                 2            j=1
!   Shc        Shchepetkin (2D momentum)             i=1          i=Lm
!
!                    W       S       E       N
!                    e       o       a       o
!                    s       u       s       r
!                    t       t       t       t
!                            h               h
!
!                    1       2       3       4

   LBC(isFsur) ==   Clo     Cha     Cha     Cha \        ! free-surface
                    Nes     Nes     Nes     Nes
   LBC(isUbar) ==   Clo     Fla     Fla     Fla \        ! 2D U-momentum
                    Nes     Nes     Nes     Nes
   LBC(isVbar) ==   Clo     Fla     Fla     Fla \        ! 2D V-momentum
                    Nes     Nes     Nes     Nes
   LBC(isUvel) ==   Clo     RadNud  RadNud  RadNud \     ! 3D U-momentum
                    Nes     Nes     Nes     Nes
   LBC(isVvel) ==   Clo     RadNud  RadNud  RadNud \     ! 3D V-momentum
                    Nes     Nes     Nes     Nes
   LBC(isMtke) ==   Clo     Gra     Gra     Gra \        ! mixing TKE
                    Nes     Nes     Nes     Nes

   LBC(isTvar) ==   Gra     RadNud  RadNud  Gra \        ! temperature
                    Gra     RadNud  RadNud  Gra          ! salinity
                    Nes     Nes     Nes     Nes \
                    Nes     Nes     Nes     Nes

! Wec boundary conditions

   LBC(isU2Sd) ==   Clo     Gra     Gra     Gra \        ! 2D U-stokes
                    Nes     Nes     Nes     Nes
   LBC(isV2Sd) ==   Clo     Gra     Gra     Gra \        ! 2D V-stokes
                    Nes     Nes     Nes     Nes
   LBC(isU3Sd) ==   Clo     Gra     Gra     Gra \        ! 3D U-stokes
                    Nes     Nes     Nes     Nes
   LBC(isV3Sd) ==   Clo     Gra     Gra     Gra \        ! 3D V-stokes
                    Nes     Nes     Nes     Nes
```

Tool Output

For Help, press F1 | 85 | 18 | Read | Ovr | Block | Sync | Rec | Caps

BC's

8) sandy_ocean.in



Time stepping

nhis, navg, etc

8) sandy_ocean.in

TextPad - G:\data\models\COAWST\Projects\Sandy\ocean_sandy.in

File Edit Search View Tools Macros Configure Window Help

Find incrementally

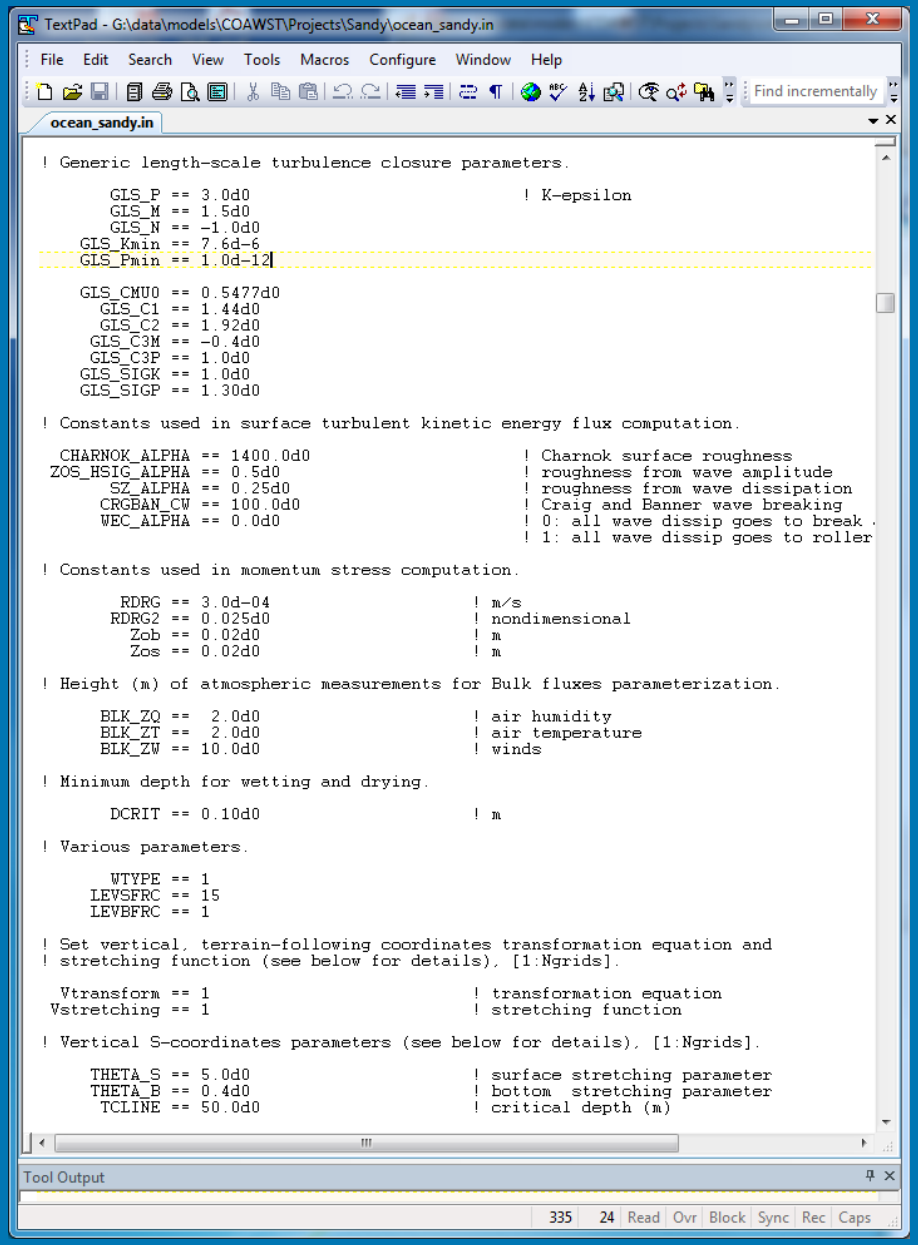
ocean_sandy.in

```
! Harmonic/biharmonic horizontal diffusion of tracer for nonlinear model
! and adjoint-based algorithms: [1:NAT+NPT,Ngrids].

        TNU2 == 0.2d0   0.2d0 0.2d0   0.2d0        ! m2/s
        TNU4 == 0.0d0   0.0d0                      ! m4/s

     ad_TNU2 == 0.0d0   0.0d0                      ! m2/s
     ad_TNU4 == 0.0d0   0.0d0                      ! m4/s

! Harmonic/biharmonic, horizontal viscosity coefficient for nonlinear model
! and adjoint-based algorithms: [Ngrids].

       VISC2 == 0.10d0 0.10d0                      ! m2/s
       VISC4 == 0.0d0                              ! m4/s

    ad_VISC2 == 0.0d0                              ! m2/s
    ad_VISC4 == 0.0d0                              ! m4/s

! Logical switches (TRUE/FALSE) to increase/decrease horizontal viscosity
! and/or diffusivity in specific areas of the application domain (like
! sponge areas) for the desired application grid.

    LuvSponge == F                                 ! horizontal momentum
LtracerSponge == F F                               ! temperature, salinity, inert

! Vertical mixing coefficients for tracers in nonlinear model and
! basic state scale factor in adjoint-based algorithms: [1:NAT+NPT,Ngrids]

     AKT_BAK == 1.0d-6  1.0d-6 1.0d-6  1.0d-6      ! m2/s

  ad_AKT_fac == 1.0d0   1.0d0                      ! nondimensional

! Vertical mixing coefficient for momentum for nonlinear model and
! basic state scale factor in adjoint-based algorithms: [Ngrids].

     AKV_BAK == 1.0d-5  1.0d-5                     ! m2/s

  ad_AKV_fac == 1.0d0                              ! nondimensional

! Turbulent closure parameters.

     AKK_BAK == 5.0d-6                             ! m2/s
     AKP_BAK == 5.0d-6                             ! m2/s
      TKENU2 == 0.0d0                              ! m2/s
      TKENU4 == 0.0d0                              ! m4/s
```

Tool Output

232 1 Read Ovr Block Sync Rec Caps

8) sandy_ocean.in



GLS params

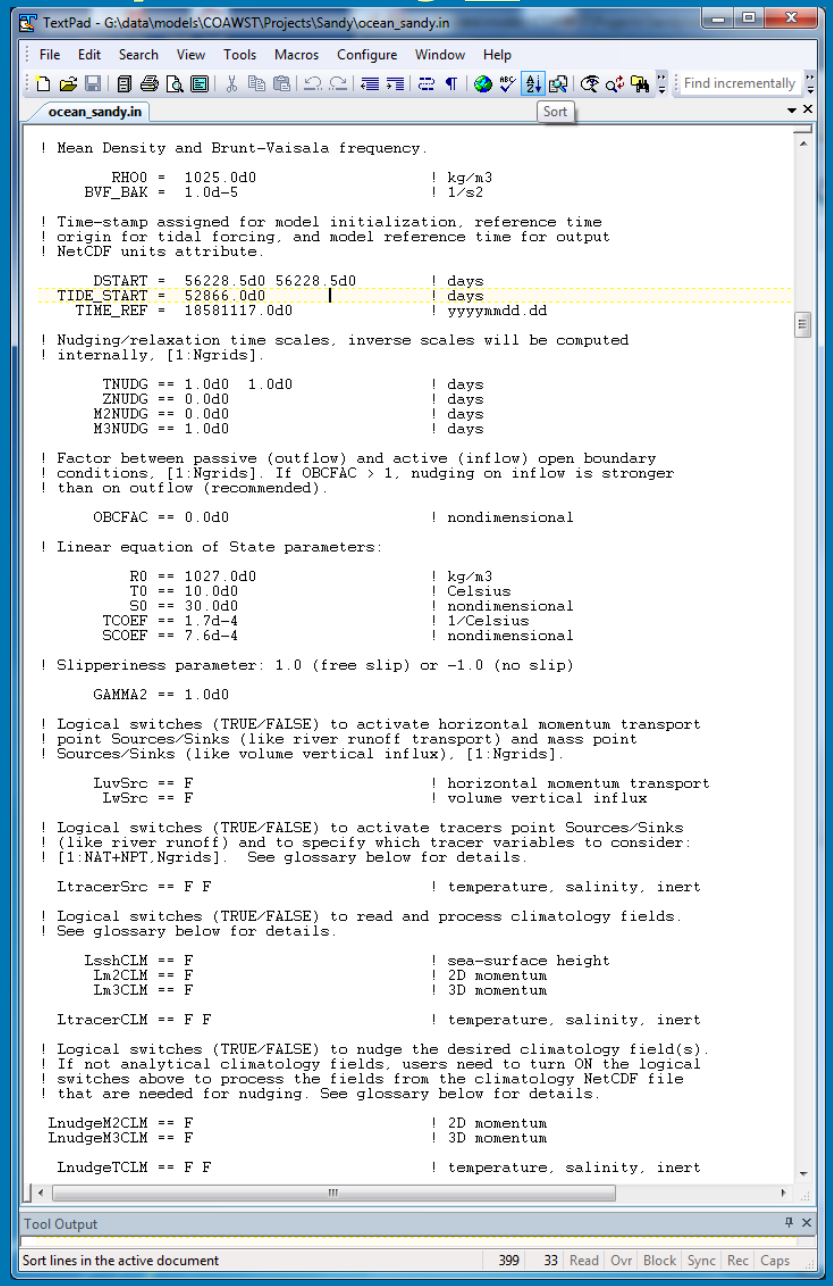
Wave breaking params

Bottom roughness

Bulk flux heights

Grid stretching params

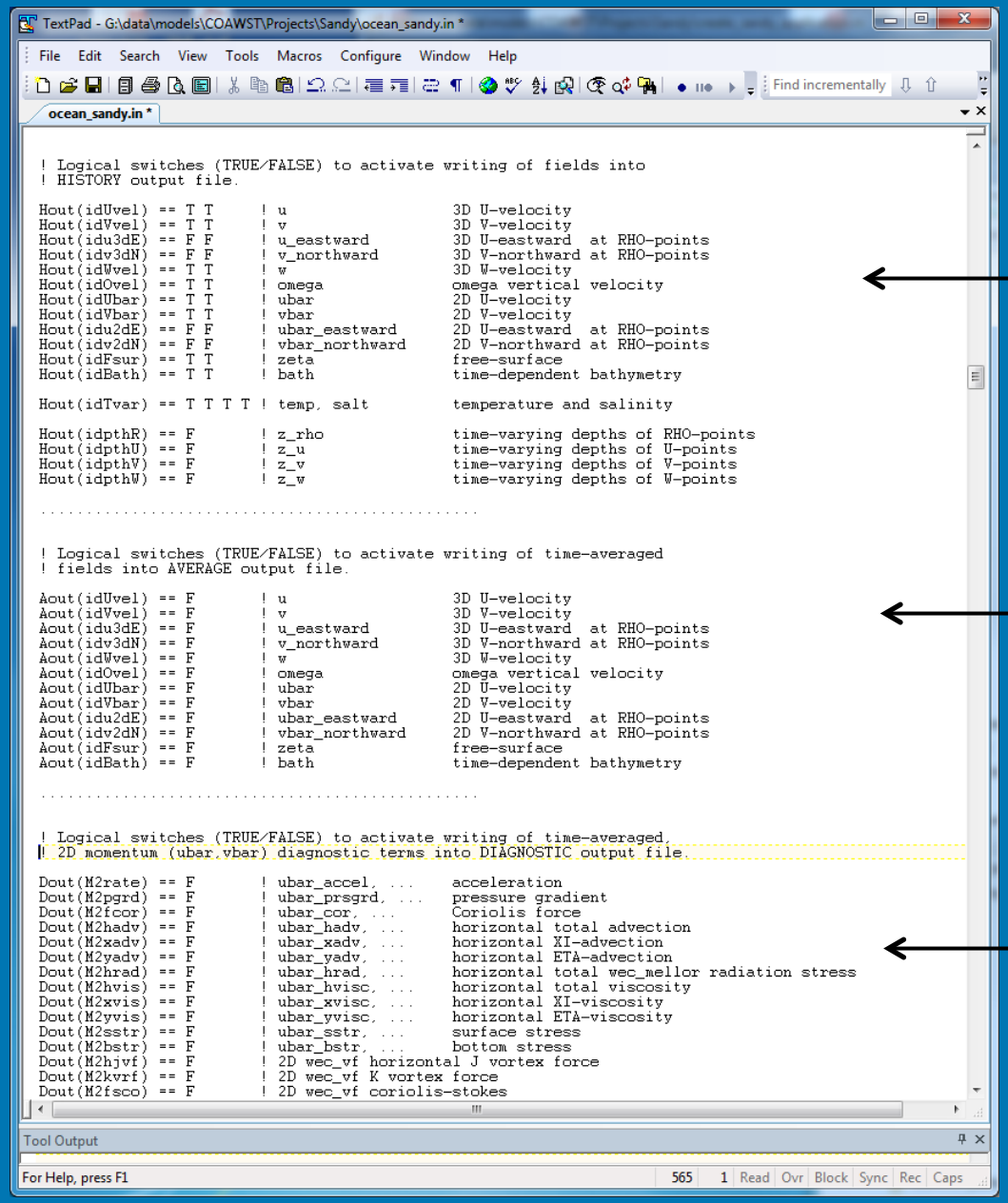
8) sandy_ocean.in



Time starts, tide start

BC tnudge

8) sandy_ocean.in

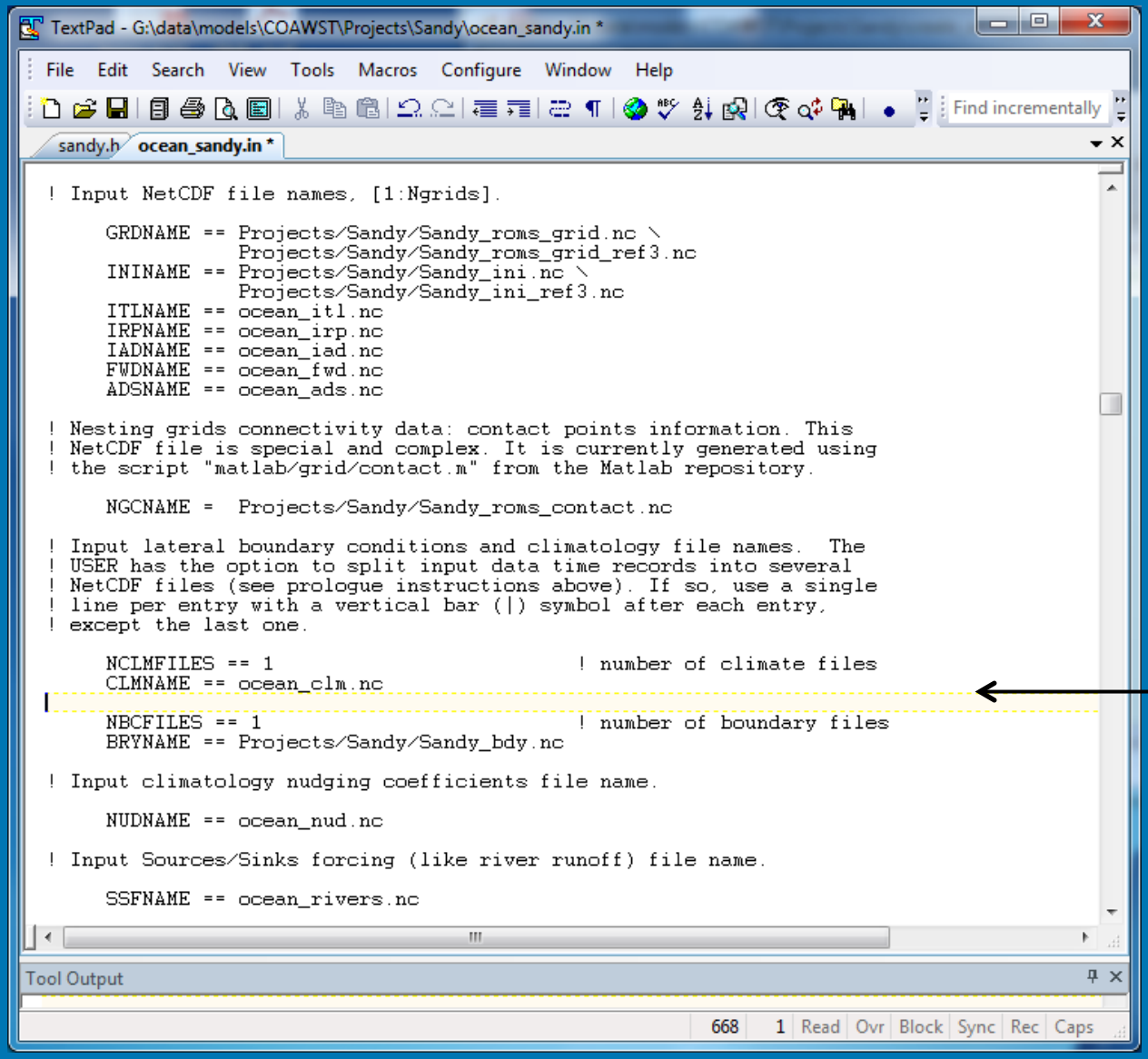


History

Averages

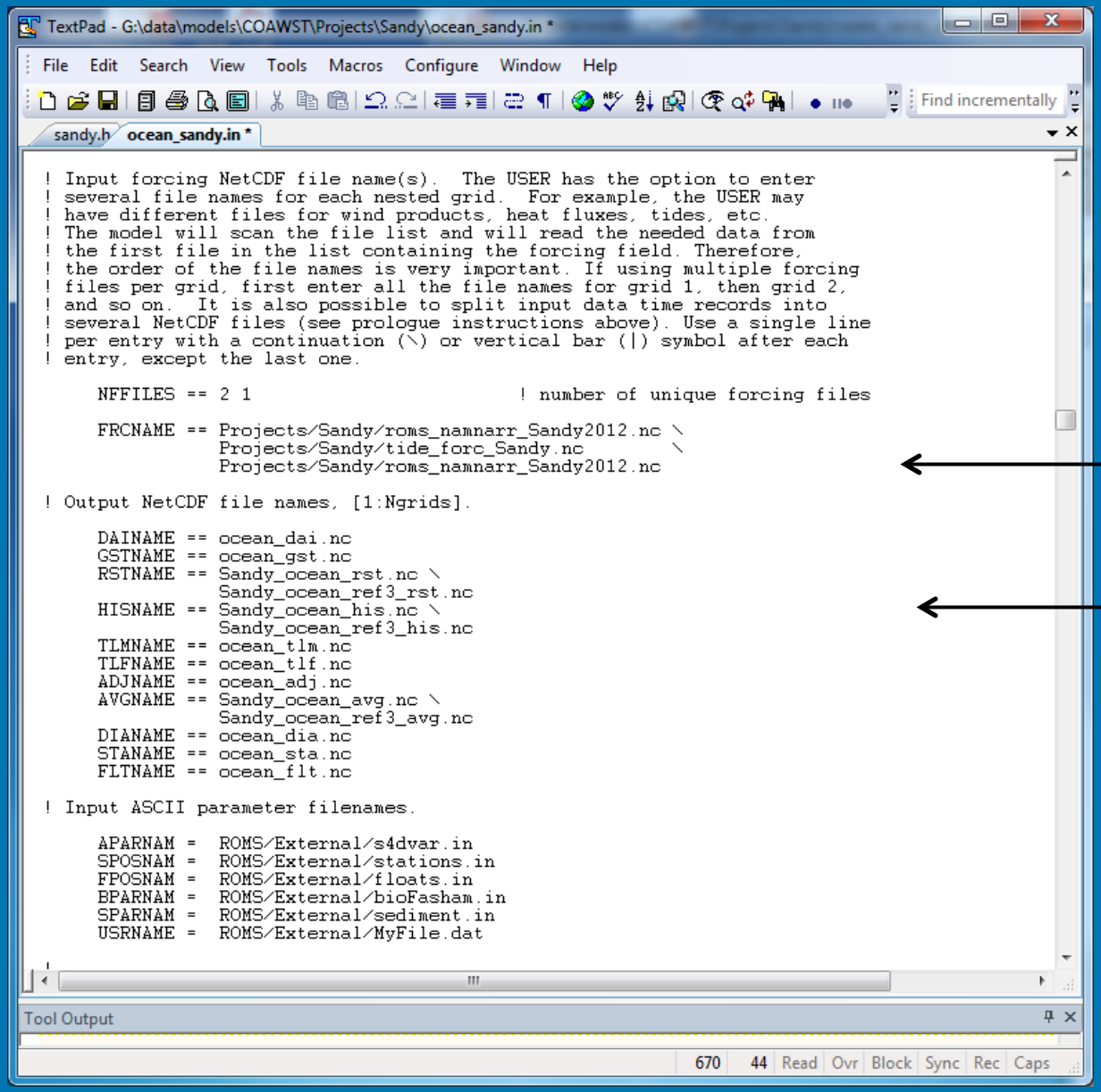
Diagnostics

8) sandy_ocean.in



Different in COAWST

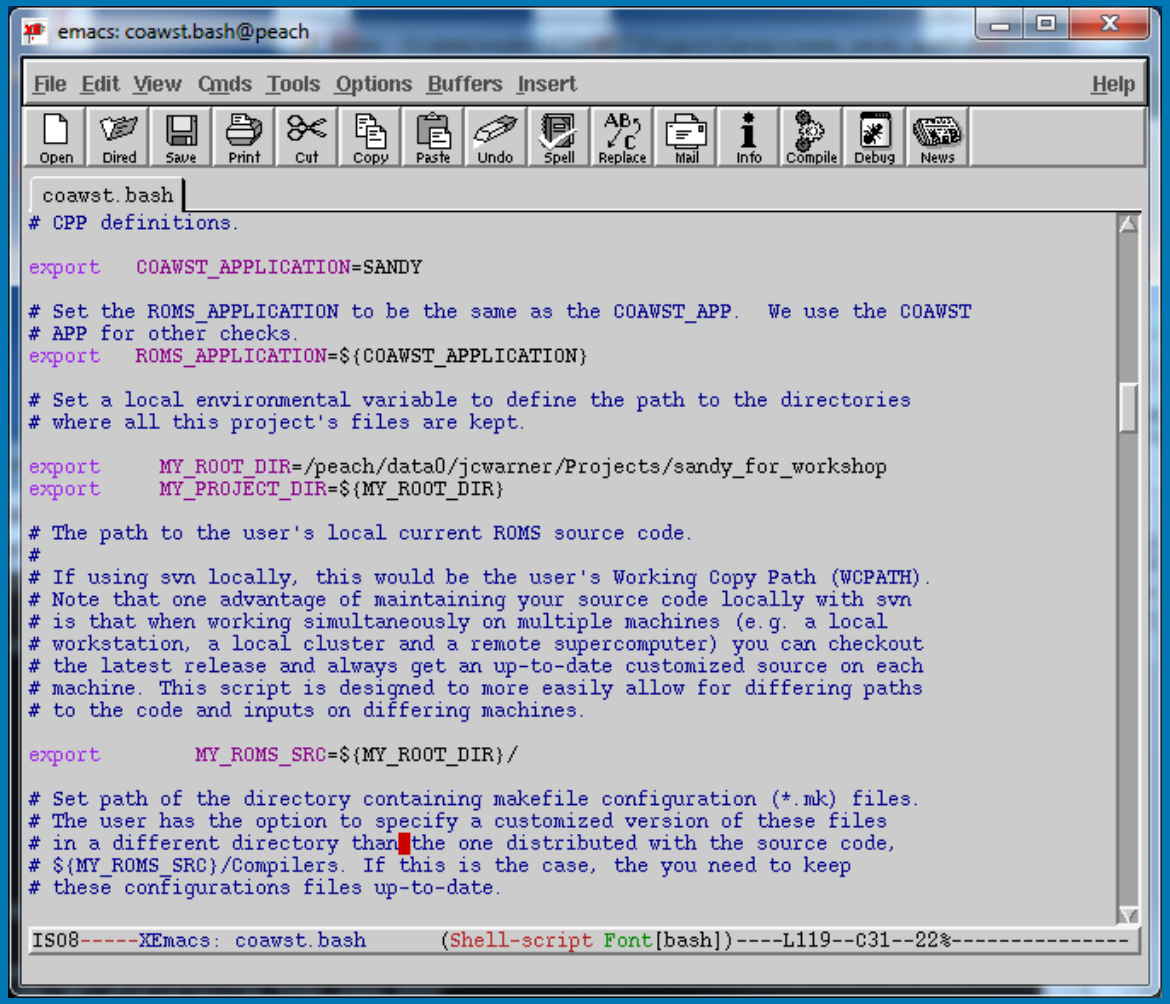
8) sandy_ocean.in



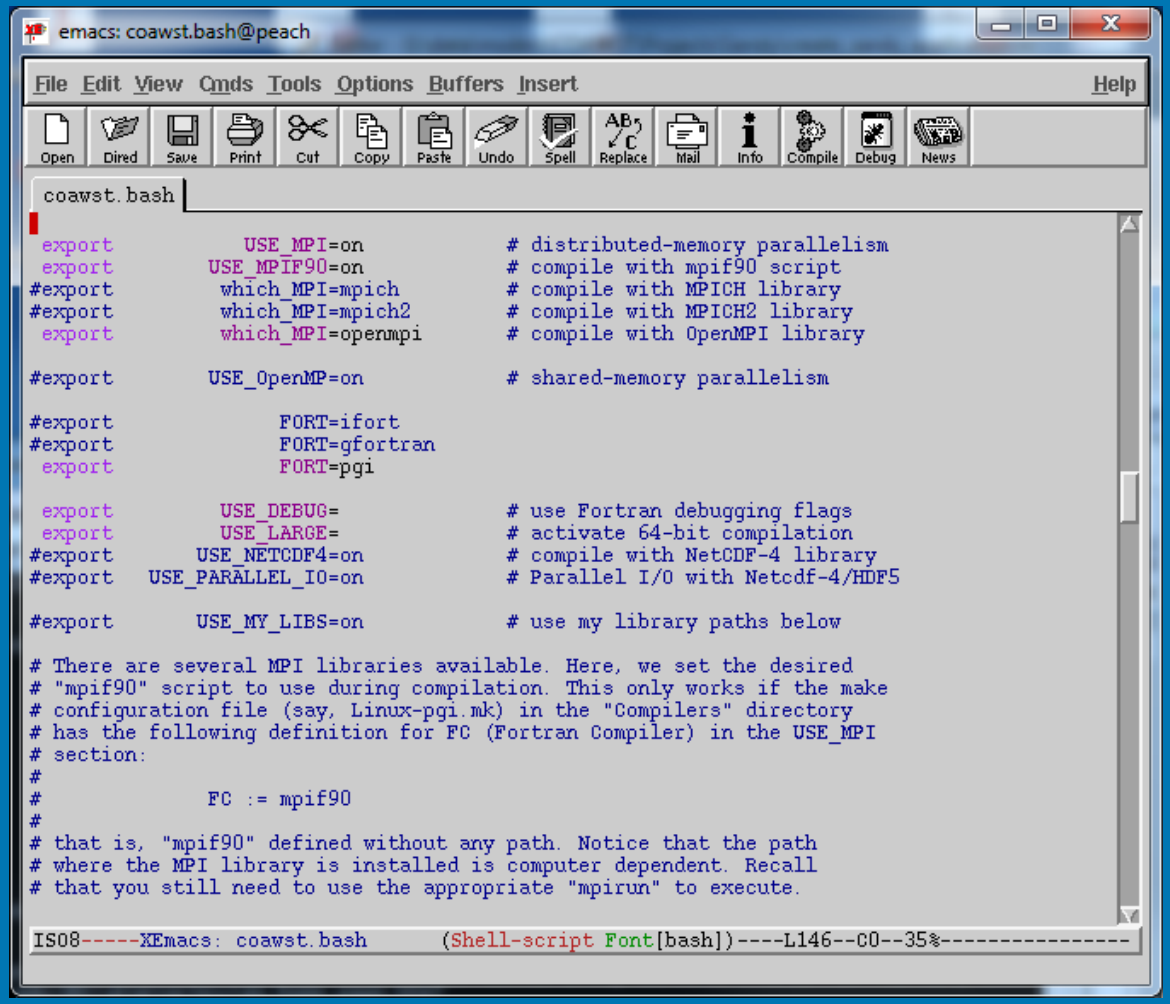
Only list tide forc for parent

his and avg names, etc

9) coawst.bash



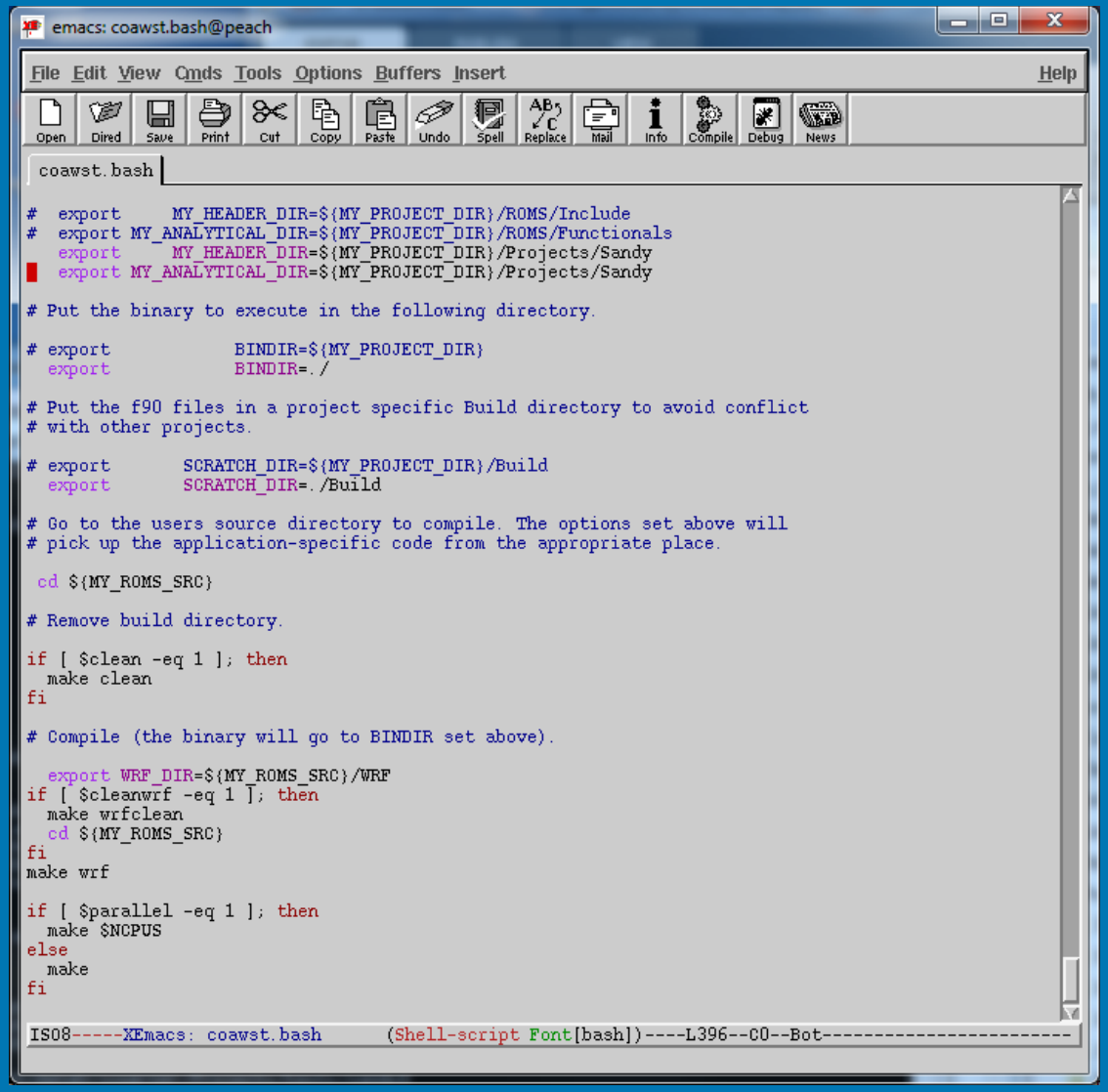
9) coawst.bash



Makefile determines the Compilers file from:
1) uname –s
2) the FORT value

For example, here it will be Compilers/Linux-pgi.mk

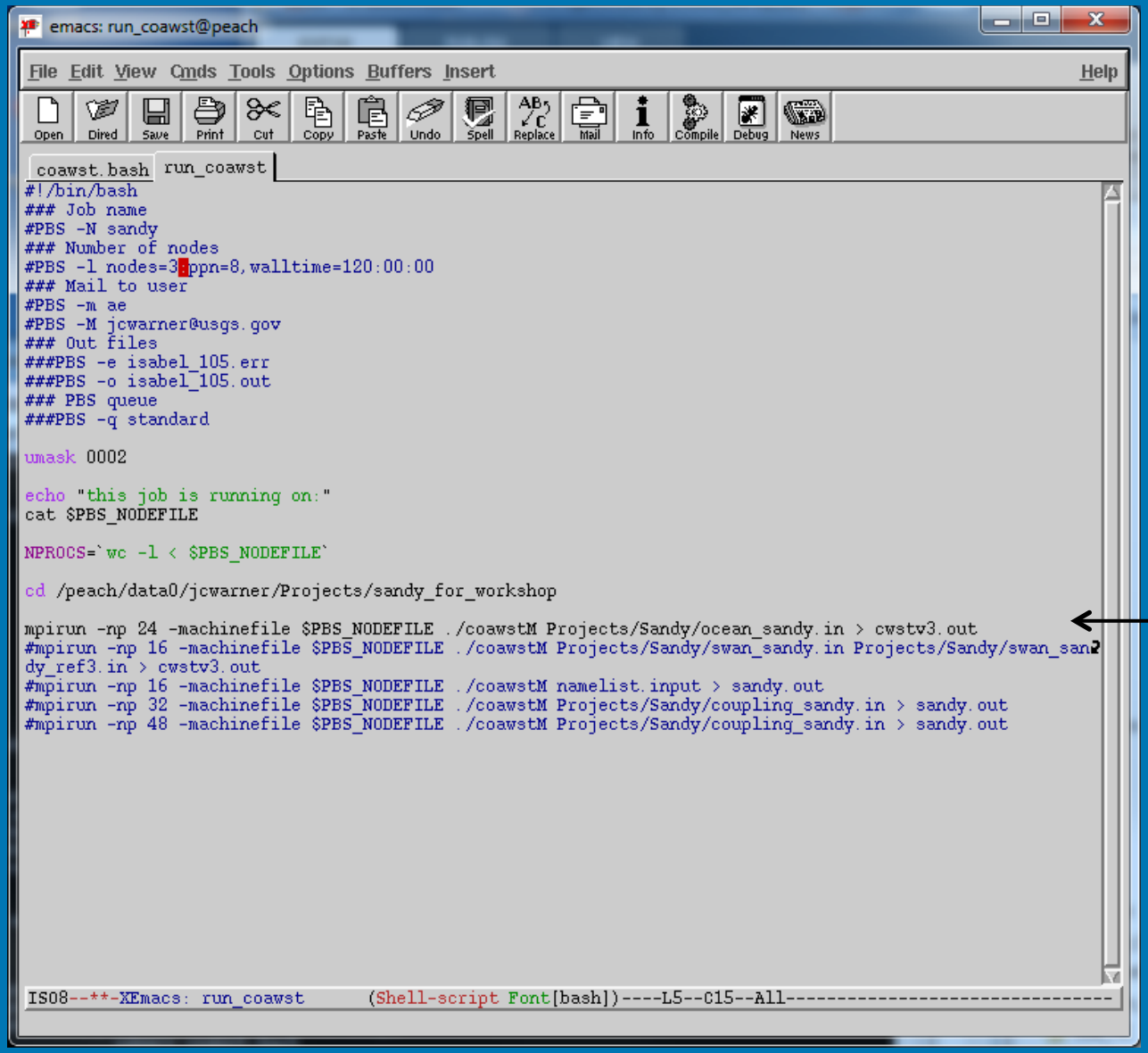
9) coawst.bash



To build the code use
./coawst.bash –j

this should make the
coawstM

10) run_file

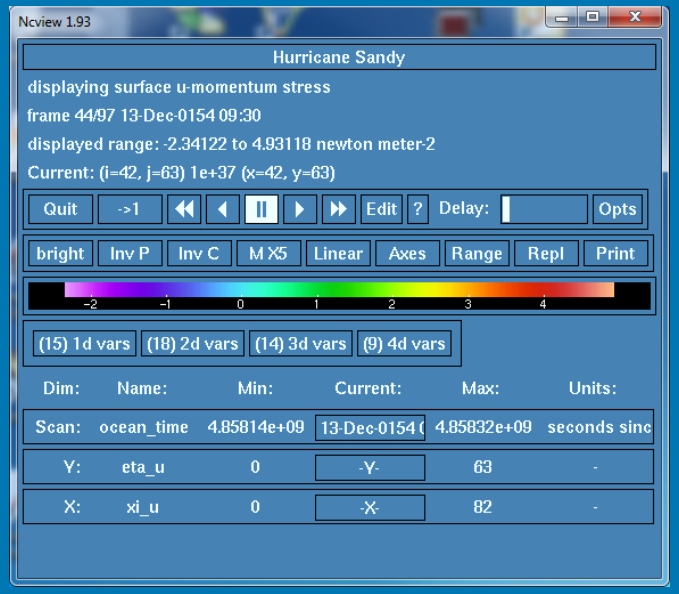


Make sure the –np X number of procs is equal to the NtileI * NtileJ in the ocean.in file.

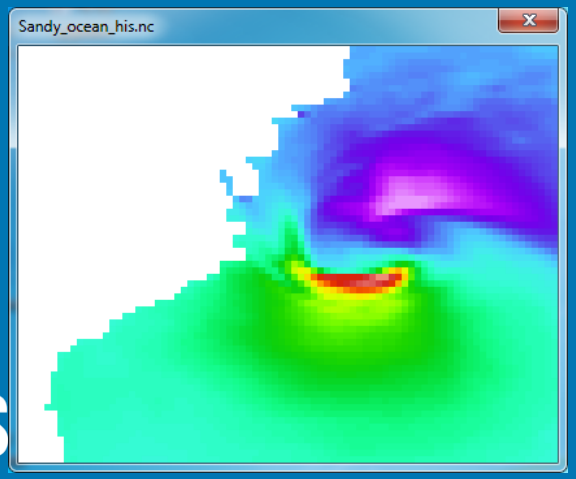
To run just ROMS, the mpirun should point to the ocean.in

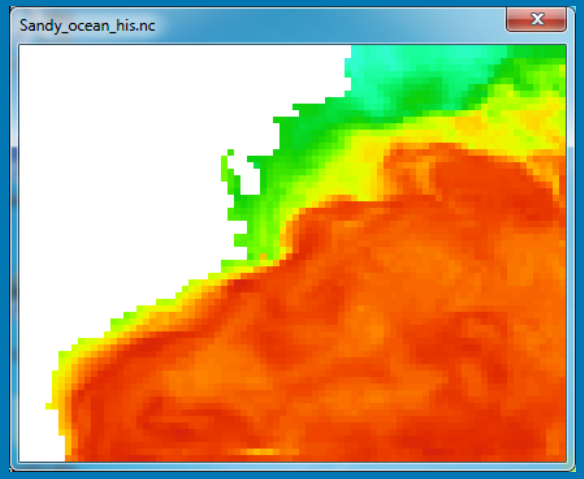
Output

sustr



temp (surface)





USGS

It's that easy. I don't understand why people have problems.