

ROMS Application Tutorial



Projects/Sandy- Hurricane Sandy example



Regional Ocean Modeling System

とばす

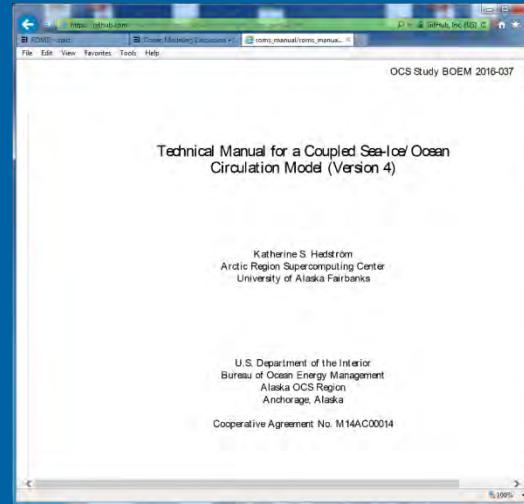
- Free surface, hydrostatic ocean model
- Finite-difference 3D Reynolds-averaged Navier-Stokes equations
- Horizontal orthogonal curvilinear Arakawa C grid
- Vertical stretched terrain-following Sigma coordinates
- Wide range of advection schemes: (e.g. 3rd-order upstream-biased, 4th-order)
- Wide range 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

ROMS information

54'50~



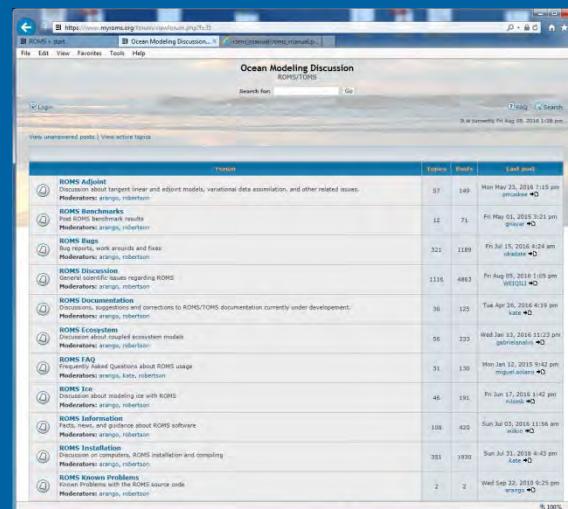
This screenshot shows the WikiROMS Documentation Portal. The left sidebar contains navigation links for Home, Getting Started, Input/Output, Tools, Test Cases, Applications, and Adjoint Algorithms. Below these are Help, Page Formatting, and FAQ sections. The main content area is divided into three columns: Technical Documentation, User Guide, and Examples and Applications. The Technical Documentation column includes links for Introduction, Equations of Motion, Vertical S-coordinate, Ternier/Following Coordinate Transformation, Curvilinear Coordinate Transformation, Time Stepping Schemes Review, Numerical Solution Technique, Nested Grids, Boundary Conditions, Horizontal Mixing, Vertical Mixing Parameterizations, Atmospheric Boundary Layer, Eulerian Boundary Layer, I-sigma-eta Dithers, Balance Term Diagnostics, Biogeographical Models, Sediment Model, Wave Model, Atmospheric Model, Model Coupling, Model Diagnostics, Sea Ice Model, Variational Data Assimilation, and Bibliography. The User Guide column covers Getting Started, Tutorials, Frequently Asked Questions, Source Code, Subversion, Modules, Drivers, Subroutines, Variables, Functionals, Lateral Boundary Conditions, C Preprocessor, Options, cpreps.h, globaldefs.h, Compiling and Linking, makefile, build Script, make, gmake, MakeDepend, External Libraries, Parallelization, Benchmarking, Matlab Scripts, and Visualization. The Examples and Applications column lists Test Cases, Application Set-up, Metadata Design, vardef.dat, Grid Generation, Initialization, Forcing, Atmospheric Forcing, Tidal Forcing, River Runoff, Input Parameter Files, ocean.in, adverb.in, biology.in, sediment.in, floats.in, stations.in, ice.in, snow.in, and Standard Output Files.



This screenshot shows the ROMS Manual (Version 4). The title is "Technical Manual for a Coupled Sea-Ice/ Ocean Circulation Model (Version 4)". It features a photo of Katherine S. Hedstrom from the Arctic Region Supercomputing Center at the University of Alaska Fairbanks. The manual is published by the U.S. Department of the Interior, Bureau of Ocean Energy Management, Alaska OCS Region, Anchorage, Alaska. A Cooperative Agreement Number M14AC00014 is mentioned. The page is 100% zoomed.

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

<https://www.myroms.org/wiki>

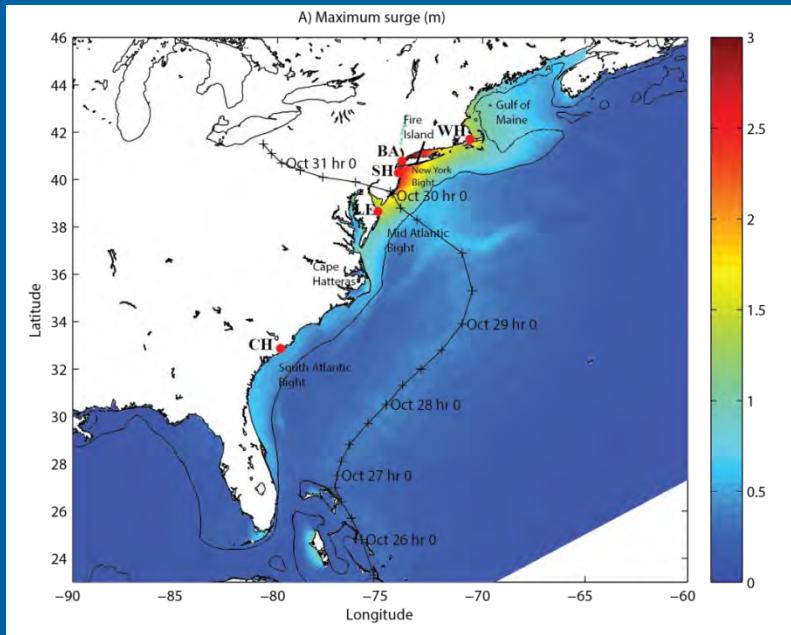


This screenshot shows the Ocean Modeling Discussion forum on the ROMS website. The forum has several active topics: ROMS Adjoint, ROMS Benchmarks, ROMS Bugs, ROMS Discussion, ROMS Documentation, ROMS Ecosystem, ROMS FAQ, ROMS ICE, ROMS Information, ROMS Installation, and ROMS Known Problems. The topics are listed with their respective dates, posters, and replies. For example, the ROMS Adjoint topic was posted on Mon May 25, 2015 7:15 pm by moderator arango, with 149 replies. The ROMS ICE topic was posted on Fri May 01, 2015 3:23 pm by moderator arango, with 71 replies.

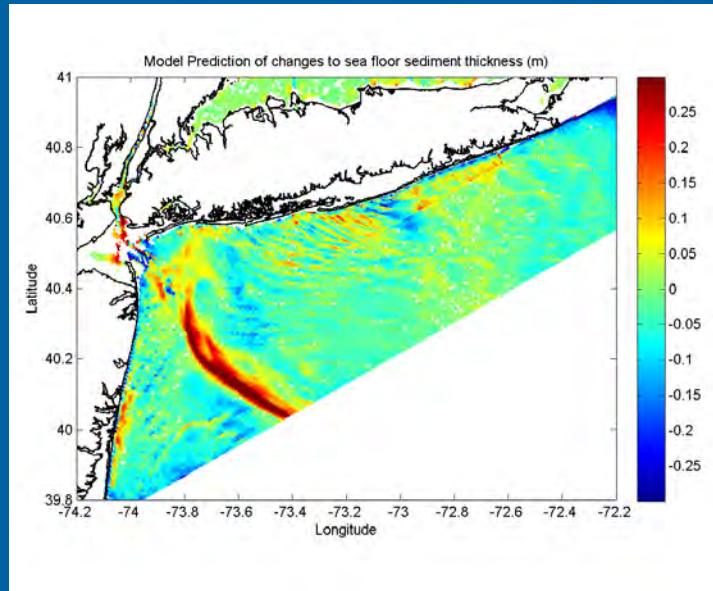
<https://www.myroms.org/forum/index.php>



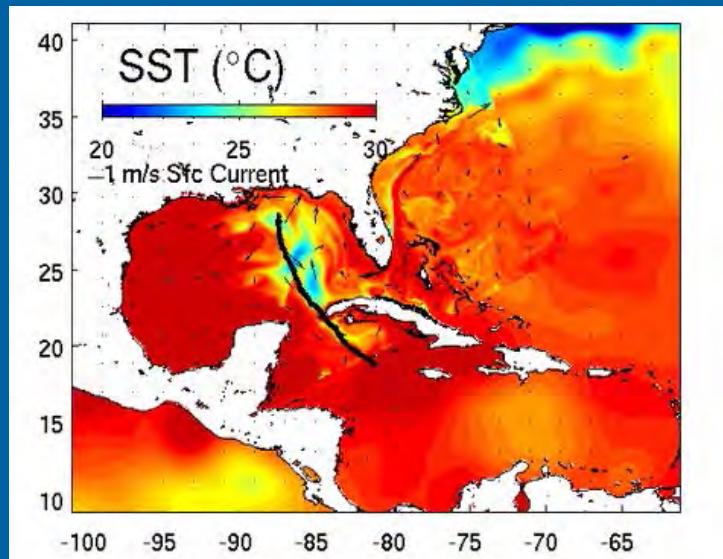
Wide Range of realistic Applications



Storm surge (H. Sandy)



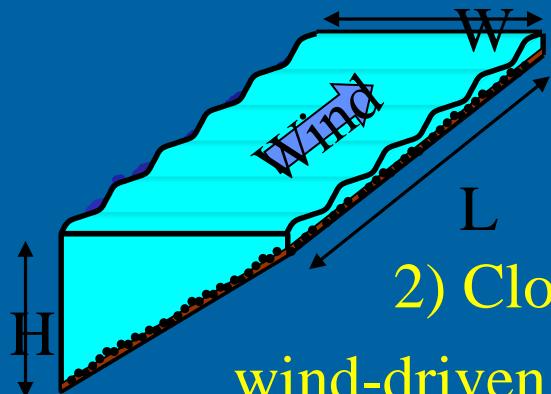
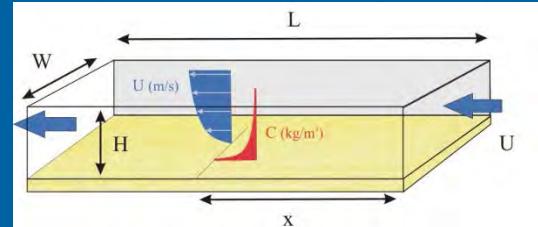
seafloor bed thickness change (H. Sandy)



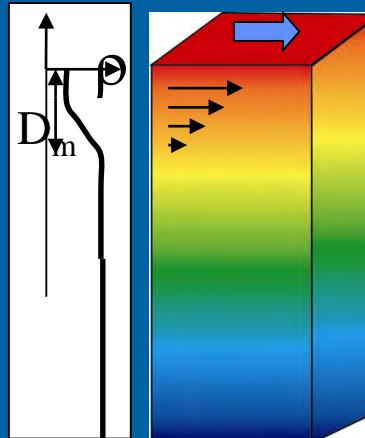
SST (H. Ivan)

Test Cases

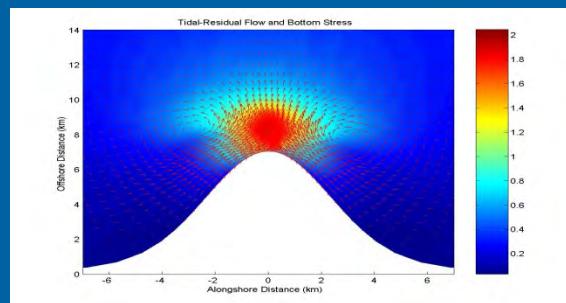
1) Open channel flow



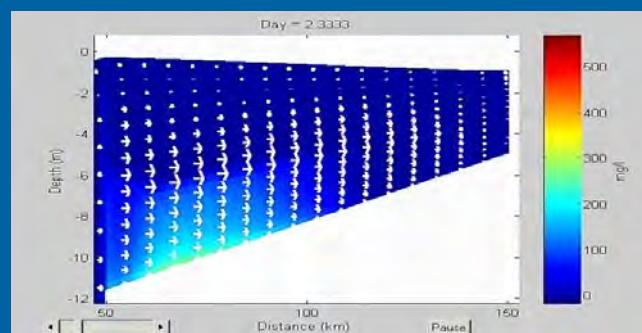
2) Closed basin, wind-driven circulation



3) Mixed layer deepening



4) Tidal flow around **USGS** a headland

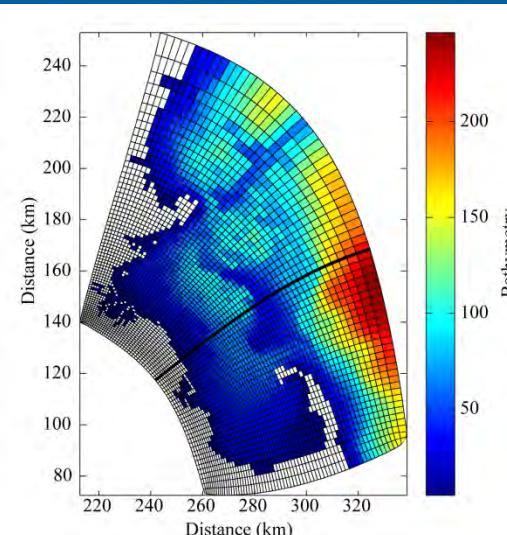
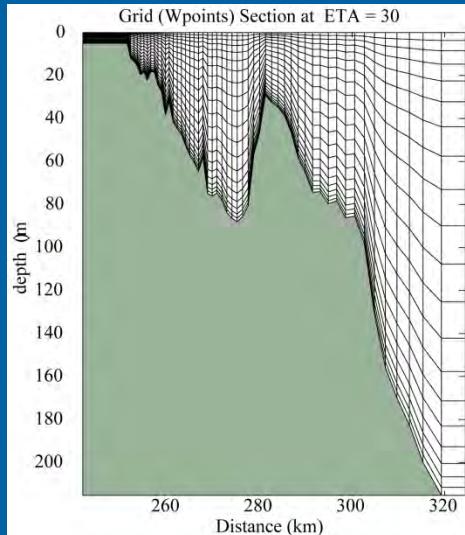


5) Estuarine circulation

ROMS Grid

9/15 45分～

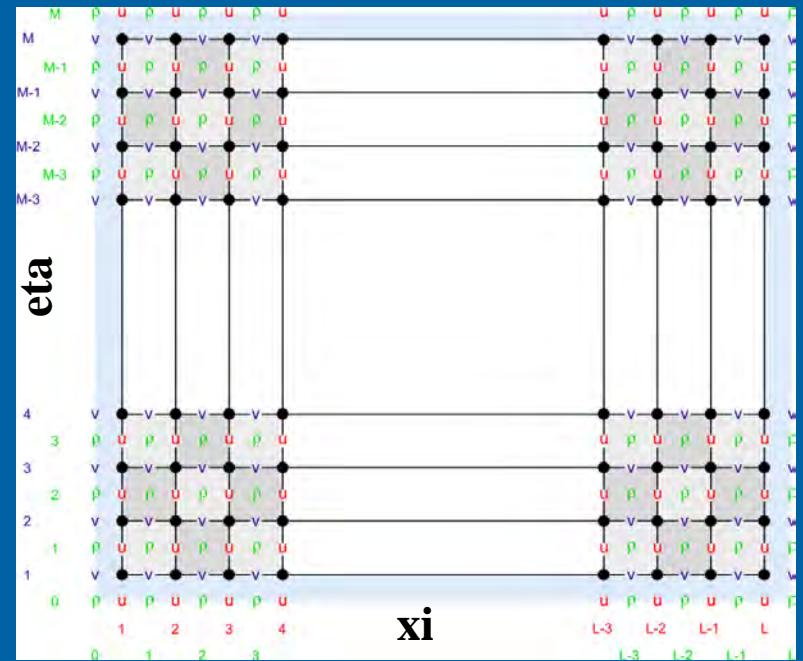
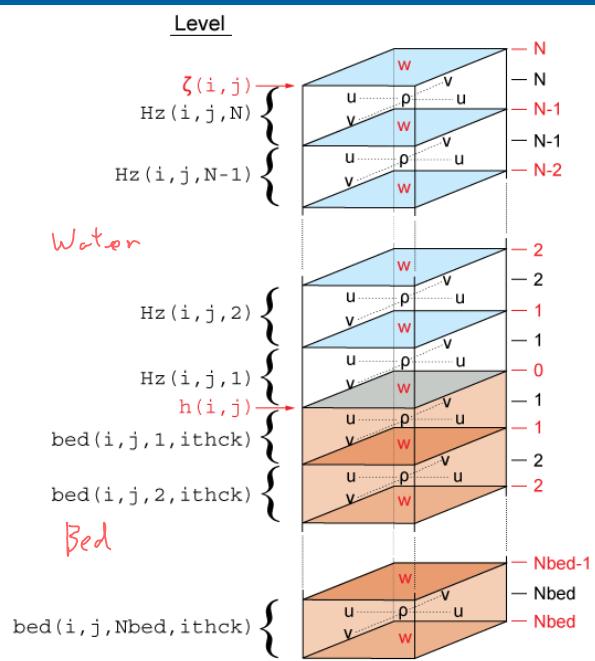
terrain following
coordinates



- masking
- curvature
- stretching

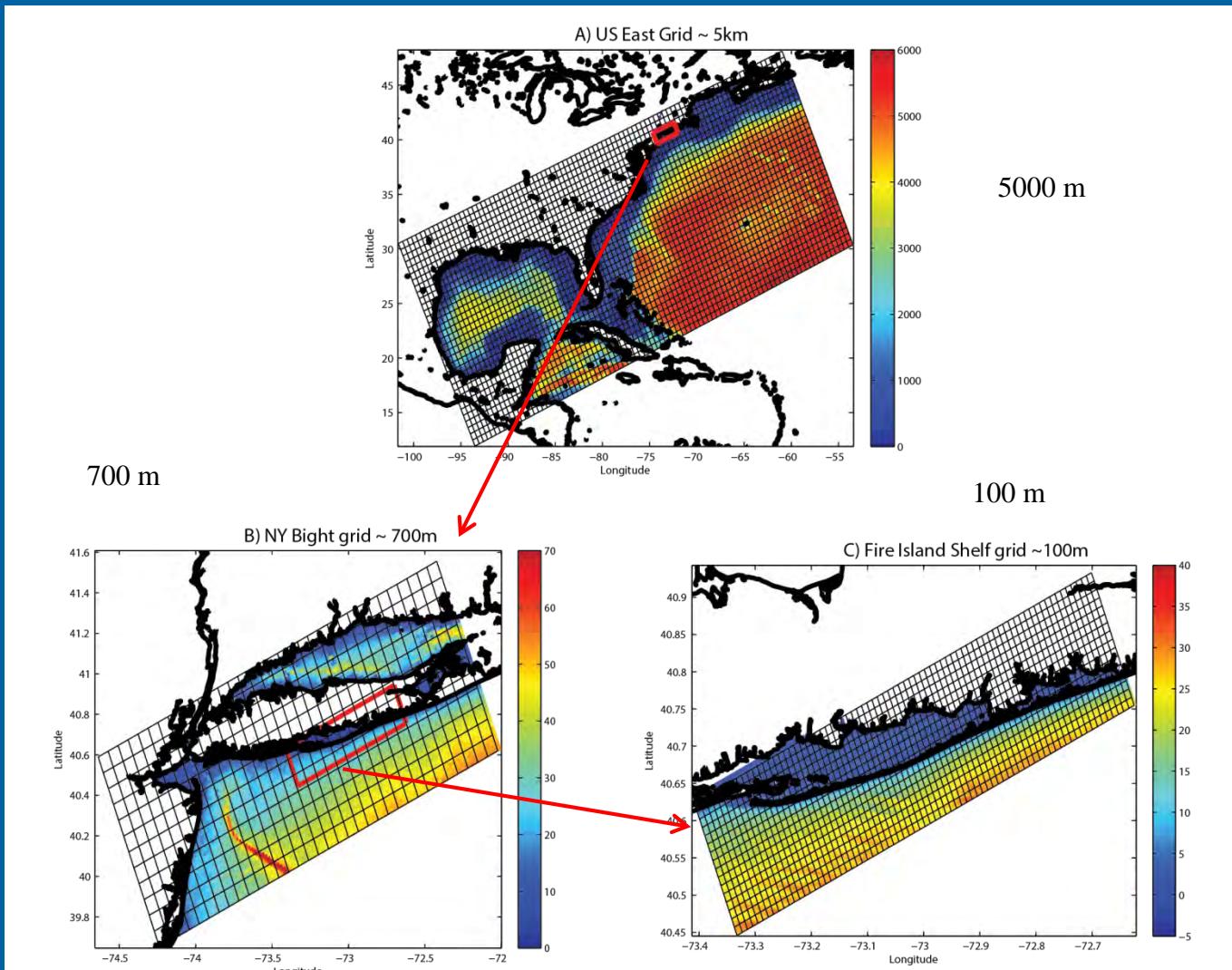
Horizontal
Arakawa "C" grid

Vertical



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 + Vstretching)

Vtransform

Transformation Equations

The following vertical coordinate transformations are available:

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

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

or

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

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

1

★ 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 $-1 \leq \sigma \leq 0$, $C(\sigma)$ is a nondimensional, monotonic, vertical stretching function ranging from $-1 \leq C(\sigma) \leq 0$, 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.



Vstretch (1 of 4)

Available Stretching Functions:

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

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

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.

Vstretch (2 of 4)

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

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

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

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

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

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

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:

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

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

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

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

Vstretch (4 of 4)

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

Surface refinement function:

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

Bottom refinement function:

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

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:

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

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

 Due to its functionality and properties $\text{Vtransform} = 2$ and $\text{Vstretching} = 4$ are now the default values for ROMS.

Equations in Mass Flux form

$$(\mathbf{u}^l, \omega^l) = (\mathbf{u}, \omega) + (\mathbf{u}^{st}, \omega^{st})$$

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

$$\begin{aligned}
 & \frac{\partial}{\partial t} \left(\frac{H_z}{mn} u \right) + \frac{\partial}{\partial \xi} \left(\frac{H_z u}{n} u \right) + \frac{\partial}{\partial \eta} \left(\frac{H_z v}{m} u \right) + u \frac{\partial}{\partial \xi} \left(\frac{H_z u^{st}}{n} \right) + u \frac{\partial}{\partial \eta} \left(\frac{H_z v^{st}}{m} \right) \\
 & \xrightarrow{\text{ACC}} \quad \xrightarrow{\text{HA}} \\
 & + \frac{\partial}{\partial s} \left(\frac{\omega_s}{mn} u \right) + u \frac{\partial}{\partial s} \left(\frac{\omega_s^{st}}{mn} \right) - H_z \left(\frac{fv}{mn} \right) - H_z \left(\frac{fv^{st}}{mn} \right) = - \frac{H_z}{n} \frac{\partial \varphi^c}{\partial \xi} \Big|_z + H_z v^{st} \left(\frac{1}{n} \frac{\partial v}{\partial \xi} - \frac{1}{m} \frac{\partial u}{\partial \eta} \right) - \omega_s^{st} \frac{\partial}{\partial s} \left(\frac{u}{mn} \right) \\
 & \xrightarrow{\text{VA}} \quad \xrightarrow{\text{COR}} \quad \xrightarrow{\text{StCOR}} \quad \xrightarrow{\text{PG}} \quad \xrightarrow{\text{HVF}} \\
 & + \frac{H_z \mathcal{F}^\xi}{mn} + \frac{H_z \mathcal{F}^{w\xi}}{mn} + \frac{H_z D^\xi}{mn} - \frac{\partial}{\partial s} \left(\overline{u w} - \frac{v}{H_z} \frac{\partial u}{\partial s} \right) + \hat{\mathcal{F}}^u \\
 & \xrightarrow{\text{BF}} \quad \xrightarrow{\text{BA+RA+BtSt+SuSt}} \quad \xrightarrow{\text{HM}} \quad \xrightarrow{\text{VM}} \quad \xrightarrow{\text{FCurv}}
 \end{aligned}$$

↑ wave breaking

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

eta-Direction Momentum Balance

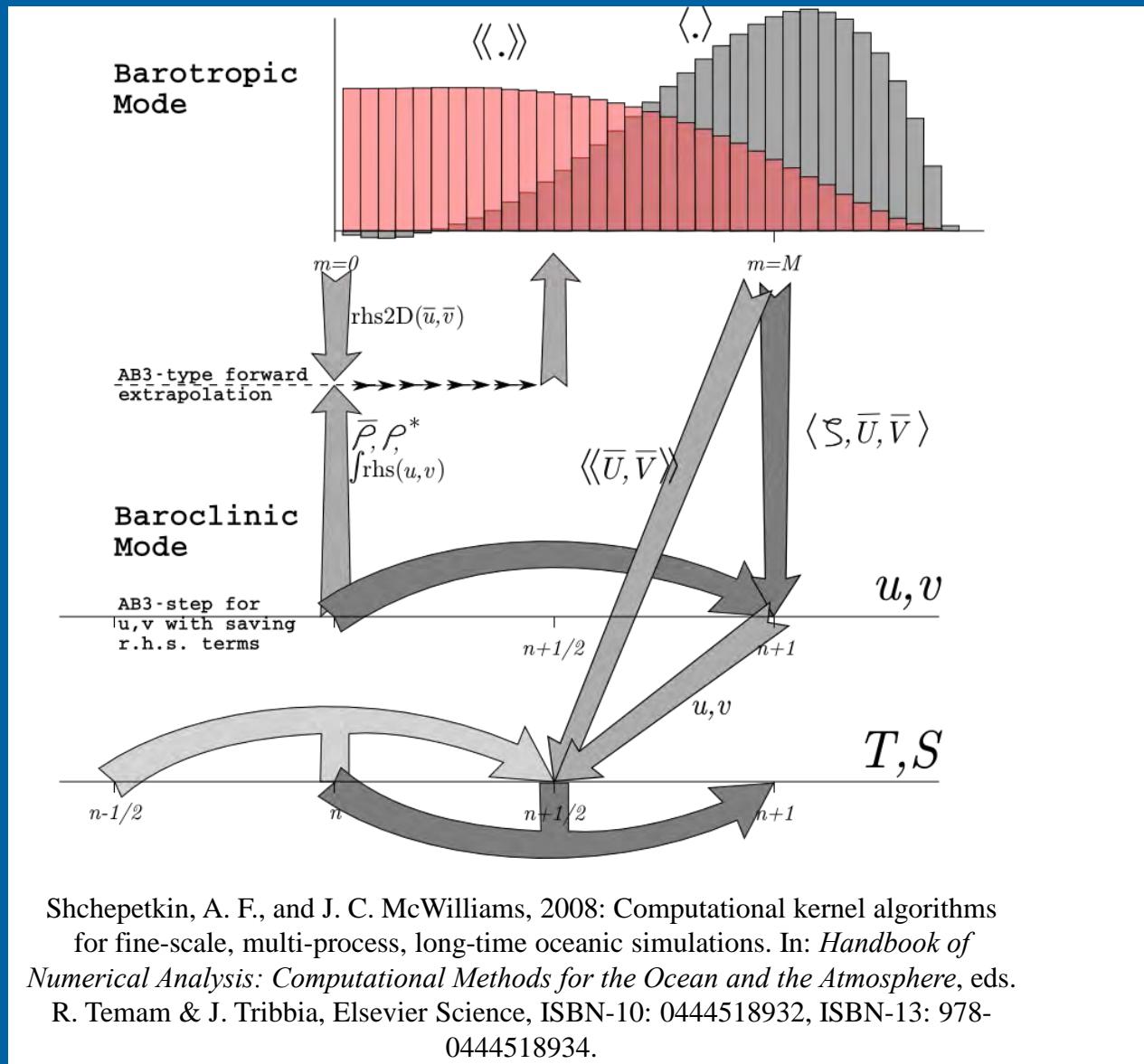
$$\begin{aligned}
 & \frac{\partial}{\partial t} \left(\frac{H_z}{mn} v \right) + \frac{\partial}{\partial \xi} \left(\frac{H_z u}{n} v \right) + \frac{\partial}{\partial \eta} \left(\frac{H_z v}{m} v \right) + v \frac{\partial}{\partial \xi} \left(\frac{H_z u^{St}}{n} \right) + v \frac{\partial}{\partial \eta} \left(\frac{H_z v^{St}}{m} \right) \\
 & \xrightarrow{\text{ACC}} \quad \xrightarrow{\text{HA}} \\
 & + \frac{\partial}{\partial s} \left(\frac{\omega_s}{mn} v \right) + v \frac{\partial}{\partial s} \left(\frac{\omega_s^{St}}{mn} \right) + H_z \left(\frac{fu}{mn} \right) + H_z \left(\frac{fu^{st}}{mn} \right) = - \frac{H_z}{m} \frac{\partial \varphi^c}{\partial \eta} \Big|_z - H_z u^{st} \left(\frac{1}{n} \frac{\partial v}{\partial \xi} - \frac{1}{m} \frac{\partial u}{\partial \eta} \right) - \omega_s^{St} \frac{\partial}{\partial s} \left(\frac{v}{mn} \right) \\
 & \xrightarrow{\text{VA}} \quad \xrightarrow{\text{COR}} \quad \xrightarrow{\text{StCOR}} \quad \xrightarrow{\text{PG}} \quad \xrightarrow{\text{HVF}} \\
 & + \frac{H_z \mathcal{F}^\eta}{mn} + \frac{H_z \mathcal{F}^{w\eta}}{mn} + \frac{H_z D^\eta}{mn} - \frac{\partial}{\partial s} \left(\frac{v}{w} - \frac{v}{H_z} \frac{\partial v}{\partial s} \right) + \hat{\mathcal{F}}^v \\
 & \xrightarrow{\text{BF}} \quad \xrightarrow{\text{BA+RA+BtSt+SuSt}} \quad \xrightarrow{\text{HM}} \quad \xrightarrow{\text{VM}} \quad \xrightarrow{\text{FCurv}}
 \end{aligned}$$

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+BtSt+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.

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



$$F^\xi = \frac{\bar{H}_z^\xi u \bar{C}^\xi}{\bar{n}^\xi}$$
$$F^\eta = \frac{\bar{H}_z^\eta v \bar{C}^\eta}{\bar{m}^\eta}$$
$$F^\sigma = \frac{\bar{H}_z^\sigma \Omega \bar{C}^\sigma}{mn}.$$

many choices, see manual for details.

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 ライブラリから 1つの 実行形式を作成
- for coupling, it makes wrf, roms, and swan as libs, then pull them together for coupling and only produces one exe → coawstM [.exe]

Build errors

無視するエラー

- 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

script コードで 作業を記録する
ファイル名を指定しない場合は typescript が作成

./coawst.bash

exit が記録終了

exit

and that will create a file build.txt

Edit that file and search for the word “error”



COAWST specific cpp options

295 ~

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

Interpolation

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.



COAWST specific cpp options

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

- `#define ATM2OCN_FLUXES`

おすすめ

--- OR ---

- `#define BULK_FLUXES`

atmosphere ocean

provide consistent fluxes between atm and ocn.

WRF send USTRESS,VSTRESS, LH, HFX, GSW, GLW

If you don't use this, then

this will send Uwind, Vwind, Patm, RH, Tair,
cloud, GSW, GLW

Related: $E - P$ (Evaporation - Precipitation)

- `#define EMINUSP`
- `#define CLOUDS`
- `#define ATM_PRESS`

send EVAP and RAIN from WRF to ROMS

send cloud fraction from WRF to ROMS

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



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 radiation stress terms from Mellor 08
- `#define WEC_MELLOR` wave-current stresses from Uchiyama et al.
- `#define WEC_VF` wave dissipation from Thorton/Guza
- `#define WDISS_THORGUZA` wave dissipation from Church/Thorton
- `#define WDISS_CHURTHOR` wave dissipation from a wave model
- `#define WDISS_WAVEMOD` wave dissipation from a InWave model
- `#define WDISS_INWAVE` wave roller based on Svendsen
- `#define ROLLER_SVENDSEN` wave roller for monochromatic waves
- `#define ROLLER_MONO` wave roller based on Reniers
- `#define ROLLER_RENIERS` wave enhanced bottom streaming
- `#define BOTTOM_STREAMING` wave enhanced surface streaming
- `#define SURFACE_STREAMING`



COAWST specific cpp options

38/20 45~

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)



+ ROMS Specific CPP options

ROMS/Include/cppdefs.h

4D ~

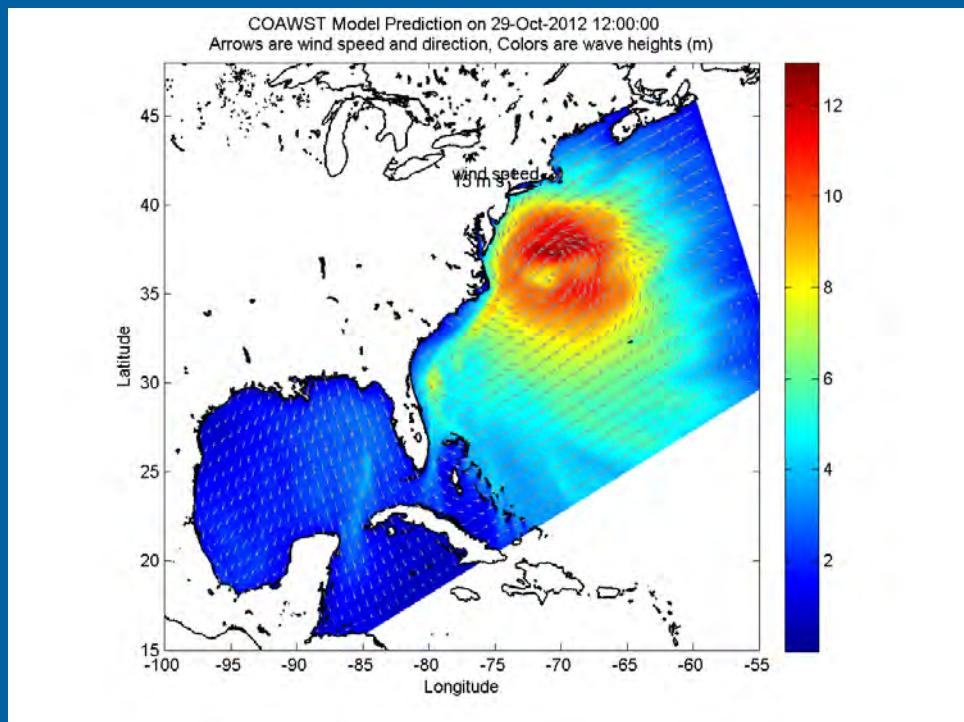
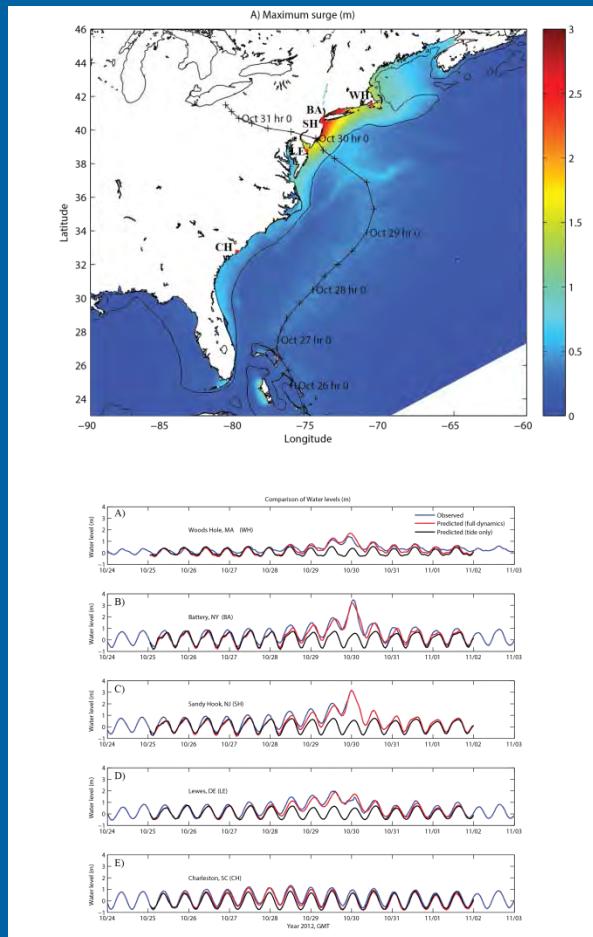
See cppdefs for a complete list of ROMS options

```
TextPad - C:\work\models\COAWST\ROMS\Include\cppdefs.h
File Edit Search View Tools Macros Configure Window Help
Find incrementally
cppdefs.h X
**
** The following is short description of all available CPP options.
**
** OPTIONS associated with momentum equations:
**
**     The default horizontal advection is 3rd-order upstream bias for
**     3D momentum and 4th-order centered for 2D momentum. The default
**     vertical advection is 4th-order centered for 3D momentum. If this
**     is the case, no flags for momentum advection need to be activated.
**
**     The 3rd-order upstream split advection (UV_U3ADV_SPLIT) can be used
**     to correct for the spurious mixing of the advection operator in
**     terrain-following coordinates. If this is the case, the advection
**     operator is split in advective and viscosity components and several
**     internal flags are activated in "globaldefs.h". Notice that
**     horizontal and vertical advection of momentum is 4th-order centered
**     plus biharmonic viscosity to correct for spurious mixing. The total
**     time-dependent horizontal mixing coefficient are computed in
**     "hmixing.F".
**
**     WARNING: Use the splines vertical advection option (UV_SADVECTION)
**             only in idealized, high vertical resolution applications.
**
** UV_ADV           use to turn ON or OFF advection terms
** UV_COR           use to turn ON or OFF Coriolis term
** UV_U3ADV_SPLIT   use if 3rd-order upstream split momentum advection
** UV_C2ADVECTION  use to turn ON or OFF 2nd-order centered advection
** UV_C4ADVECTION  use to turn ON or OFF 4th-order centered advection
** UV_SADVECTION    use to turn ON or OFF splines vertical advection
** UV_VIS2          use to turn ON or OFF harmonic horizontal mixing
** UV_VIS4          use to turn ON or OFF biharmonic horizontal mixing
** UV_SMAGORINSKY  use to turn ON or OFF Smagorinsky-like viscosity
** UV_DRAG_GRID     use if spatially varying bottom friction parameters
** UV_LOGDRAG       use to turn ON or OFF logarithmic bottom friction
** UV_LDRAG         use to turn ON or OFF linear bottom friction
** UV_QDRAG         use to turn ON or OFF quadratic bottom friction
** UV_WAVEDRAG      use to turn ON or OFF extra linear bottom wave drag
** SPLINES_VVISC    use if splines reconstruction of vertical viscosity
**
Tool Output
Search Results Tool Output
11 | 1 | Read | Ovr | Block | Sync | Rec | Caps
```

This list goes on
for many many
more pages

42 ↗ ~

Projects/Sandy Application



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) ROMS input files: `sandy.h` and `ocean_sandy.in`
- 9) Build and run: `coawst.bash`
- 10) Output

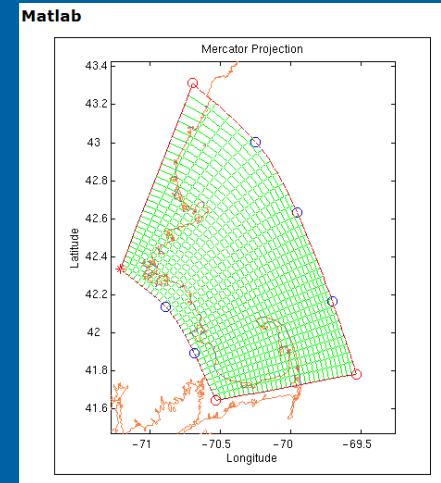


These steps will follow the User Manual Section 9

1) Grid generation tools

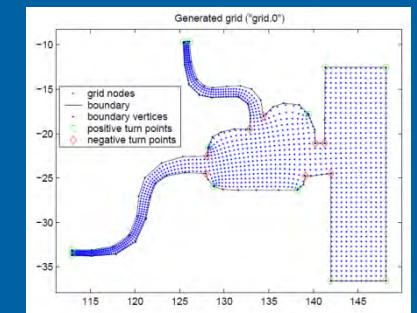
■ Seagrid - matlab

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



■ gridgen - command line

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

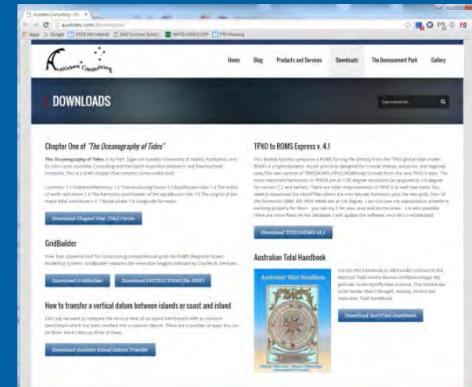


■ EASYGRID

1) Grid generation tools

- Austides - Gridbuilder

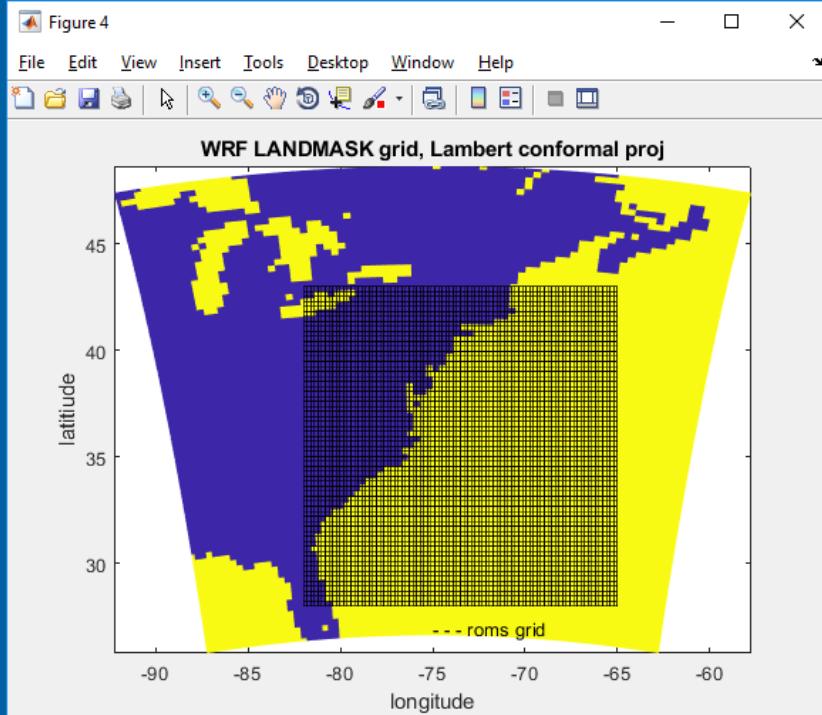
<http://austides.com/downloads/>



- 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



1) Grid generation tools – matlab



select 4 corners

We will use the wrf grid to get the outline, but with a different resolution. These steps use Matlab. Load the wrf grid to get coastline data, cd to Projects/Sandy and use:

```
netcdf_load('wrfinput_d01')
figure
pcolorjw(XLONG,XLAT,double(1-LANDMASK))
hold on
title('WRF LANDMASK grid, Lambert conformal proj')
xlabel('longitude'); ylabel('latitude')
```

Pick 4 corners for roms parent grid

```
xl=-82; xr=-65;
yb= 28; yt= 43;
```

Pick number of points in the grid

```
numx=86; numy=64;
```

Make a matrix of the lons and lats.

```
dx=(xr-xl)/numx; dy=(yt-yb)/numy;
[lon, lat]=meshgrid(xl:dx:xr, yb:dy:yt);
lon=lon.';lat=lat.';
plot(lon,lat,'k-')
plot(lon',lat','k-')
text(-75,27,'--- roms grid')
```

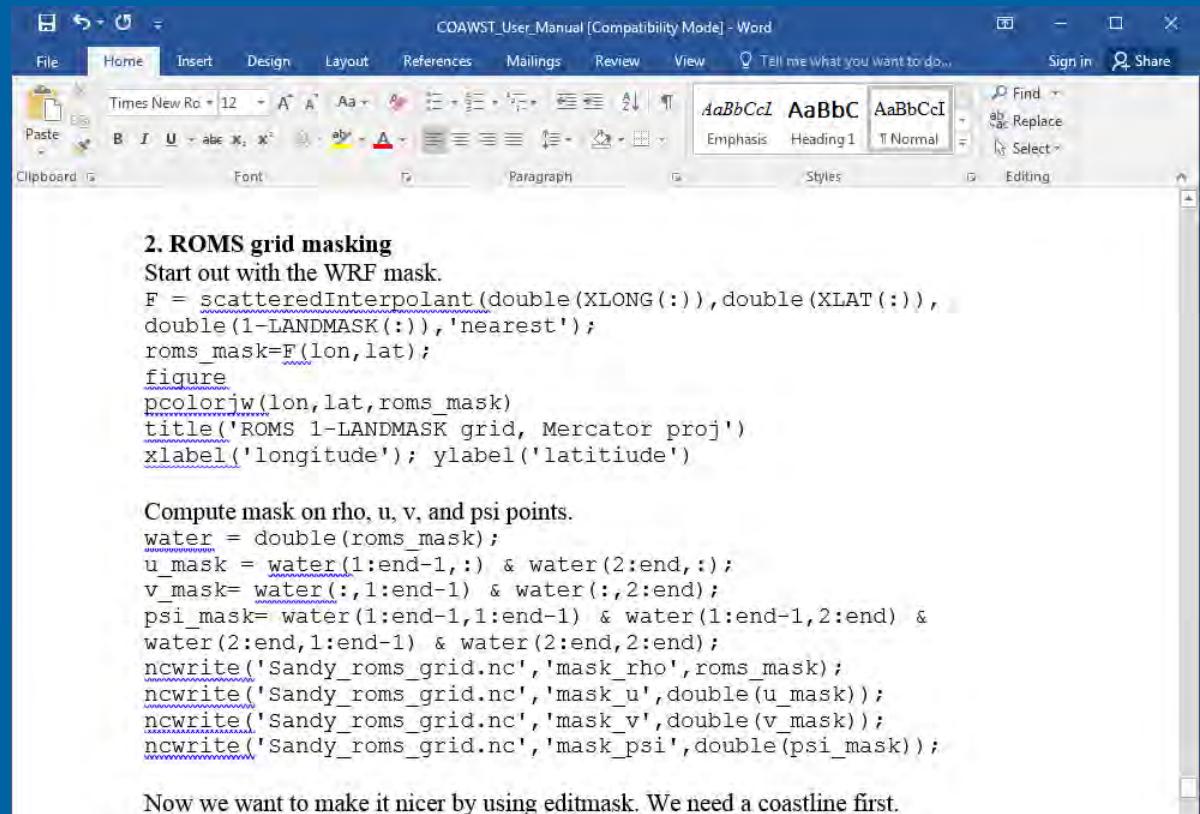
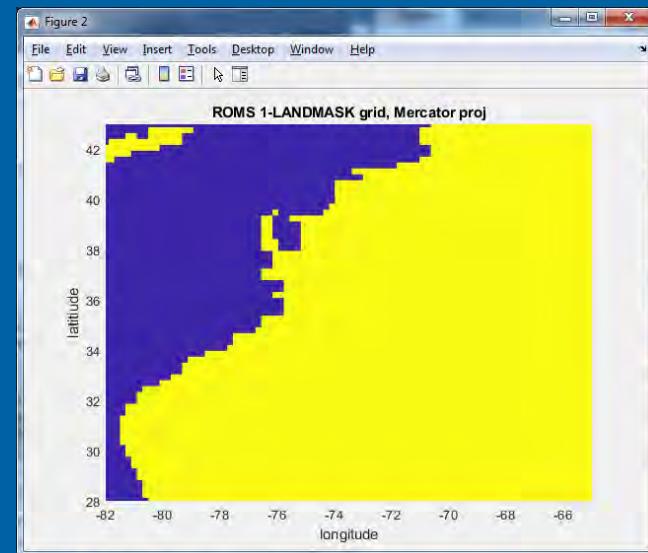
Figure. First steps to show WRF and ROMS grids.

1) Grid generation tools

```
Call generic grid creation.
roms_grid='Sandy_roms_grid.nc';
rho.lat=lat; rho.lon=lon;
rho.depth=zeros(size(rho.lon))+100; % for now just make zeros
rho.mask=zeros(size(rho.lon)); % for now just make zeros
spherical='T';
%projection='lambert conformal conic';
projection='mercator';
save temp_jcw33.mat rho spherical projection
eval(['mat2roms_mw(''temp_jcw33.mat'', '''',roms_grid,'''');'])
!del temp_jcw33.mat
%User needs to edit roms variables
disp(['      '])
disp(['Created roms grid --> ',roms_grid])
disp(['      '])
disp(['You need to edit depth in ',roms_grid])
disp(['      '])
```

This step creates a roms grid file called ‘Sandy_roms_grid.nc’

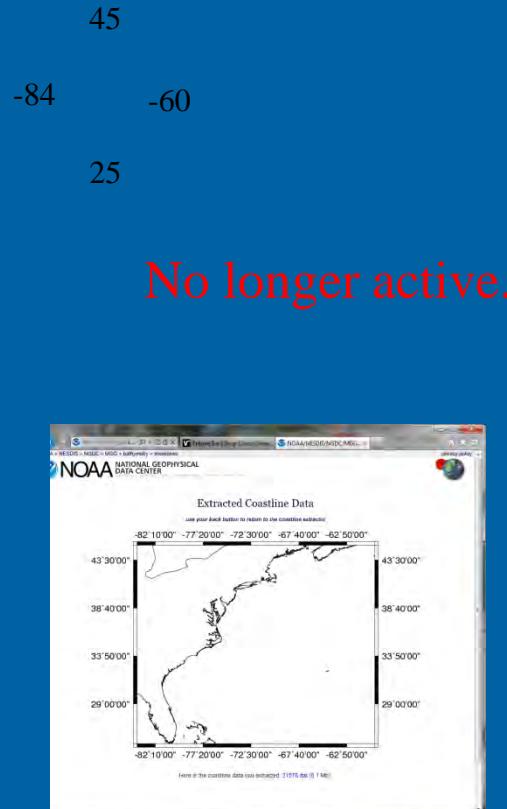
2) masking – first base it on WRF



2) Masking – get coastline

May also need a coastline, can obtain this here:

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



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

The screenshot shows a web browser displaying the NOAA GEODAS-NG Desktop Software page. The URL is <http://www.ngdc.noaa.gov/mgg/geodas/geodas.html>. The page header includes the NOAA logo and links for 'All MGG Data', 'All Marine Geophysics', 'Bathymetry', 'Seismic Reflection', and 'Trackline Geophysical Data Viewer'. The main content area is titled 'GEODAS-NG Desktop Software for use with Downloaded Trackline Geophysical Data' and features a 'Trackline Geophysical Data Viewer' thumbnail. A note states: 'NOEI's GEODAS-NG (Geophysical DAyta System - Next Generation) desktop software tools can be used for working with data downloaded from NGDC's trackline web presence, especially NOEI-developed, downloaded formats. While this software suite is expected to work well for some time, support in the form of updates and other changes may not be available after late 2014. It may be useful to explore other commercial and open-source tools to determine if they meet your needs.' Below this is a section titled 'NGDC (now NCEI) Exchange Formats' with a link to 'GEOODAS-NG Software applications and many non-NCEI sites often use a set of data exchange formats (M2D-2000) developed at NGDC (now NCEI). Formats included are:' followed by a bulleted list: 'Marine Trackline Geophysical Data Exchange Format "M2D77/M2D77T"' and 'Airborne Magnetic Surveys Exchange Format - "MAG88T"'. The bottom section is titled 'Coastlines' with a note: 'NCEI has moved to using the GSHHG High-resolution Geography Database for World Shorelines and Political Boundaries. GEODAS-NG software (see Desktop Windows Software below) works with GSHHG/WDBII. GEODAS application Coastline Extractor allows sub-setting GSHHG/WDBII (and many other shoreline data sets) by area, resolution, etc.'

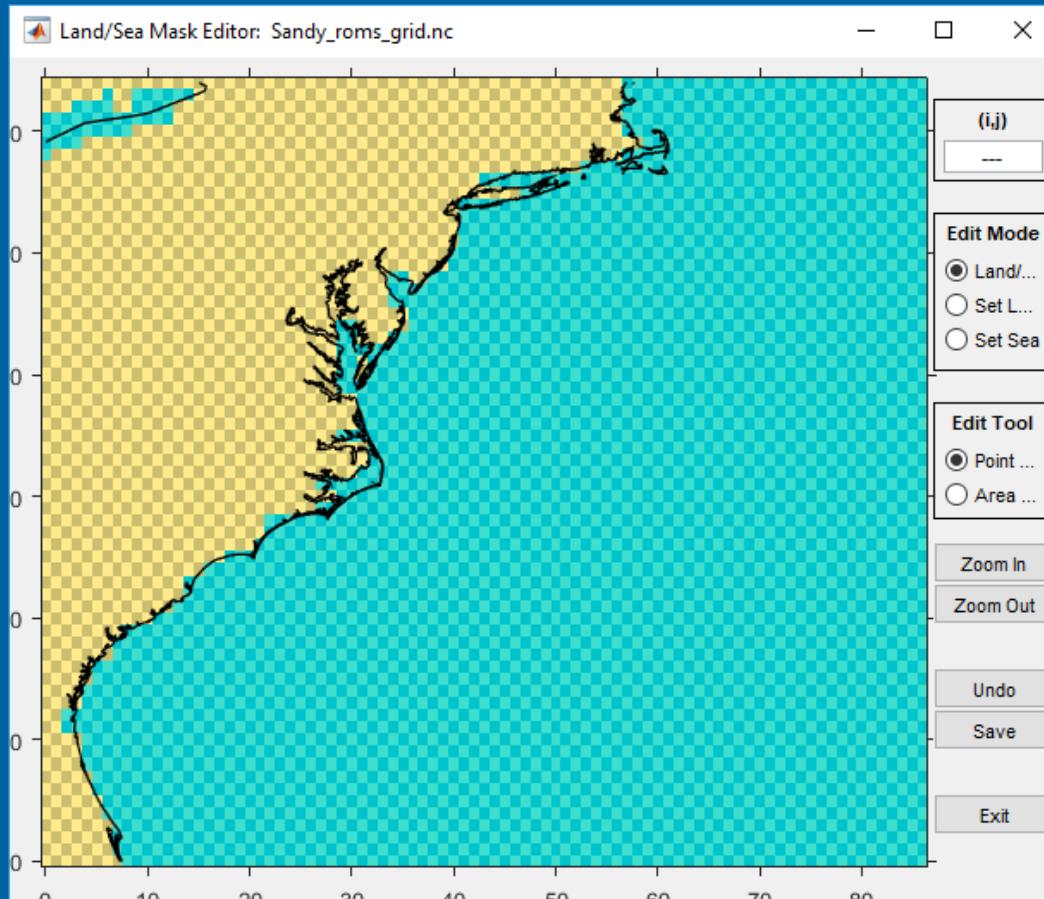
<http://www.ngdc.noaa.gov/mgg/geodas/geodas.html>
shorelines/



2) Masking

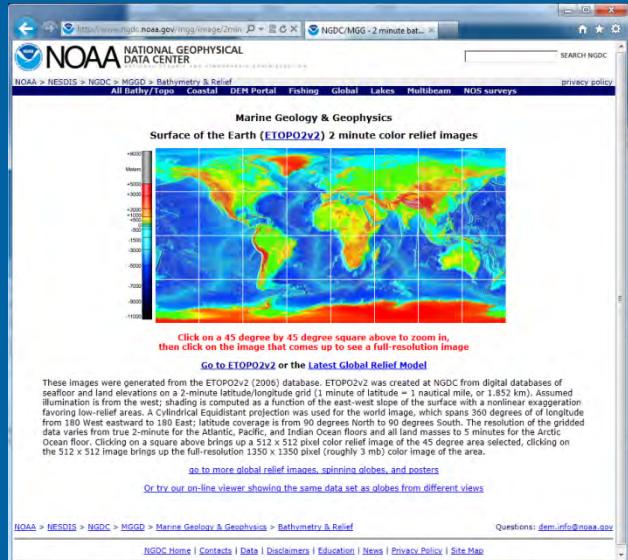
use COAWST/Tools/mfiles/rutgers/landmask/editmask m file

```
editmask('Sandy_roms_grid.nc','coastline.mat')
```

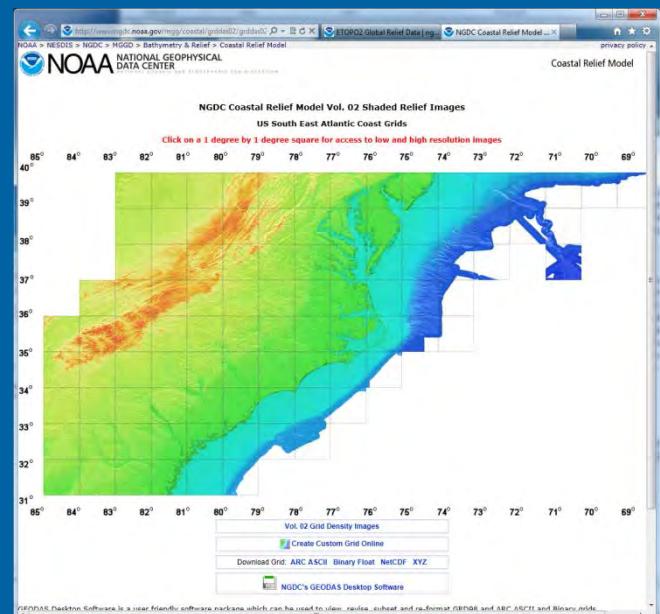


3) bathymetry

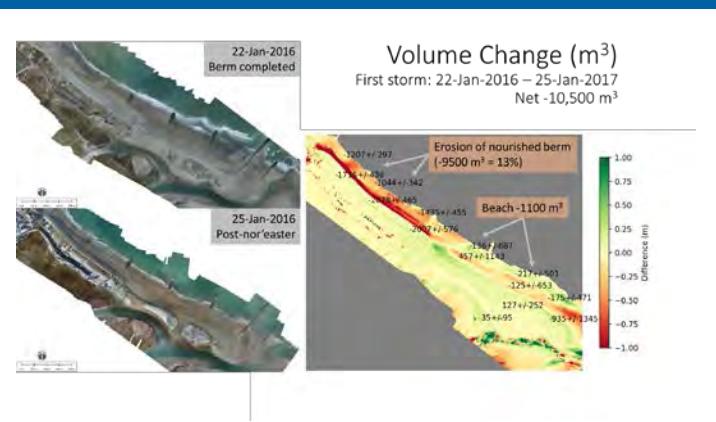
many sources



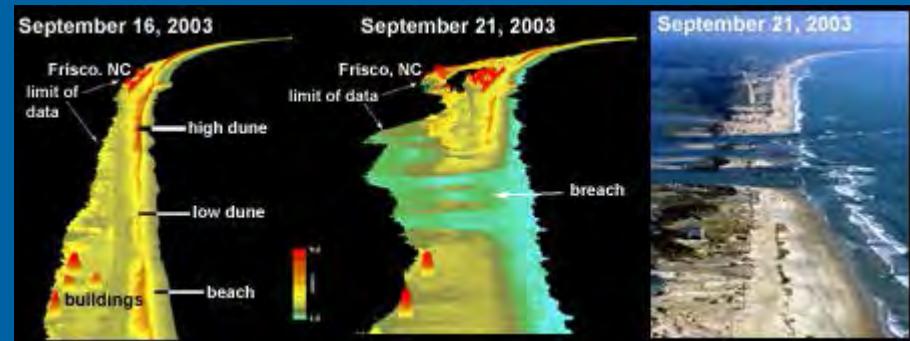
ETOPO2



Coastal Relief Model

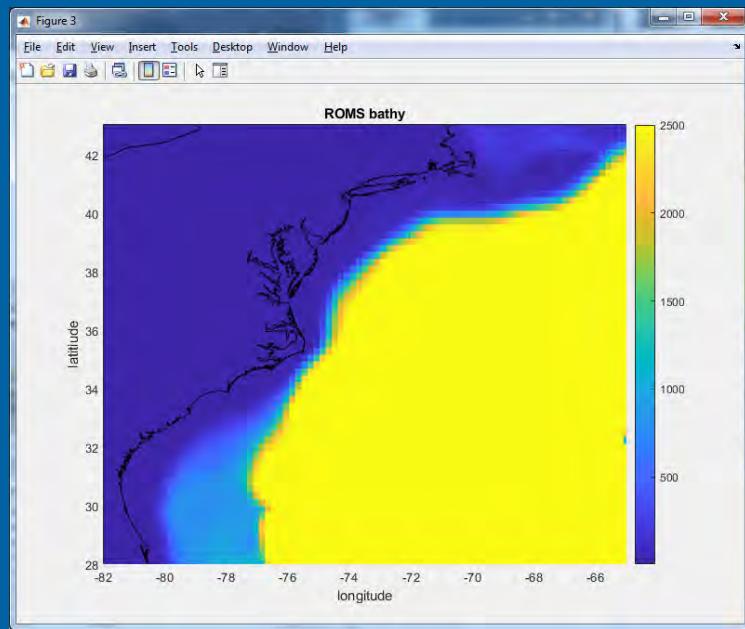


SFM structure from motion



LIDAR

3) bathymetry



COAWST_User_Manual [Compatibility Mode] - Word

File Home Insert Design Layout References Mailings Review View Tell me what you want to do... Sign in Share

Font Paragraph Styles Editing

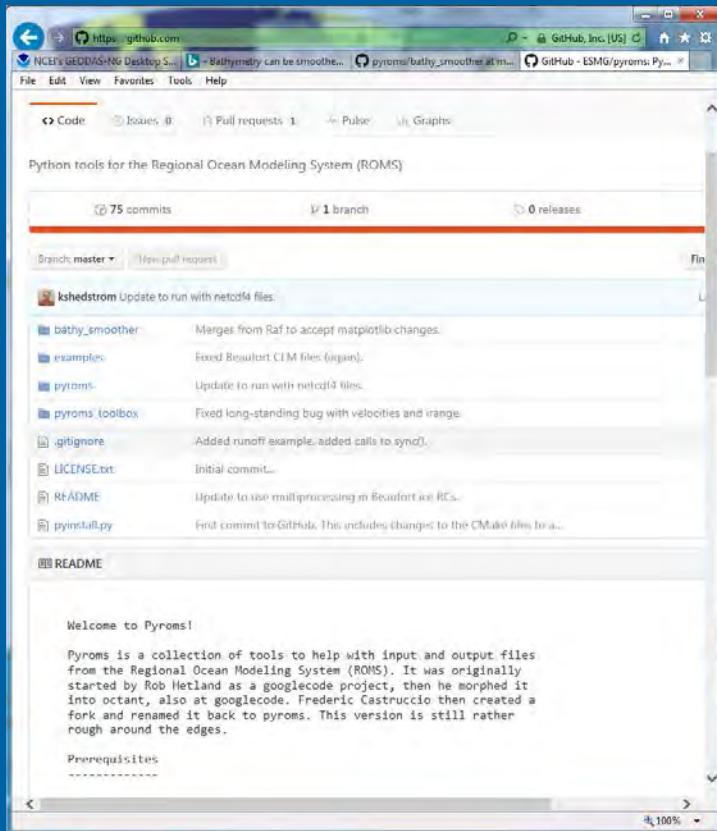
3. Bathymetry

Bathymetry can come from many different places. You can use a source like this:
<http://www.ngdc.noaa.gov/mgg/global/global.html>

For this application, we use data from a local file

```
load USEast_bathy.mat
netcdf_load('Sandy_roms_grid.nc')
h=griddata(h_lon,h_lat,h_USeast,lon_rho,lat_rho);
h(isnan(h))=5;
%smooth h a little
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)+h(2:end-1,1:end-2)+h(2:end-1,3:end));
figure
pcolorjw(lon_rho,lat_rho,h)
hold on
load coastline.mat
plot(lon,lat,'k')
caxis([5 2500]);
colorbar
title('ROMS bathy')
xlabel('longitude'); ylabel('latitiude')
ncwrite('Sandy_roms_grid.nc','h',h);
```

3) bathymetry



Bathy smoothing:
<https://github.com/ESMG/pyroms>



Manual of the matlab scripts of LP Bathymetry

Mathieu Dutout Sikirić

April 26, 2011

When one uses the ROMS model, one needs to smooth the bathymetry in order to get realistic results. Two roughness factors are involved: the rx_0 factor of Beckman and Haidvogel:

$$rx_0 = \max_{e \in e'} \frac{|h(e) - h(e')|}{h(e) + h(e')}$$

which should not go above 0.2 and the rx_1 factor of Haney which should not be above 6 [1]. (both rx_0 and rx_1 are shown up at the beginning of a ROMS run).

The original physical bathymetry as computed by interpolation and sampling is often too rough for the models and a smoothing operation is needed. The programs exposed here try given a roughness factor to find the bathymetry that is nearest to the real one. More details are given in [2].

The factor that matters is actually the rx_1 number which is required to be small. The problem is that it is quite difficult to optimize with respect to rx_1 . The idea is to assume that there is a multiplying factor between rx_0 and rx_1 , i.e. $rx_1 = Crx_0$ and to optimize rx_0 instead of rx_1 . This works quite well for $Vtransform=1$ but not for the other transformations that were introduced later. Then a possible solution is to optimize with respect to a varying factor rx_0 . The appropriate functions are provided.

1 Availability

The source of the program is available from <http://www.liga.ens.fr/~dutour/Bathymetry/index.html>

The linear programs are solved by the program `lpsolve` (see [6] for the installation). Note that we do not use the mex facility but the standalone program. The scripts are `matlab®` scripts and so you need to have `matlab®` installed.

LP Bathy

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

COAWST_User_Manual [Compatibility Mode] - Word

4. ROMS child grid

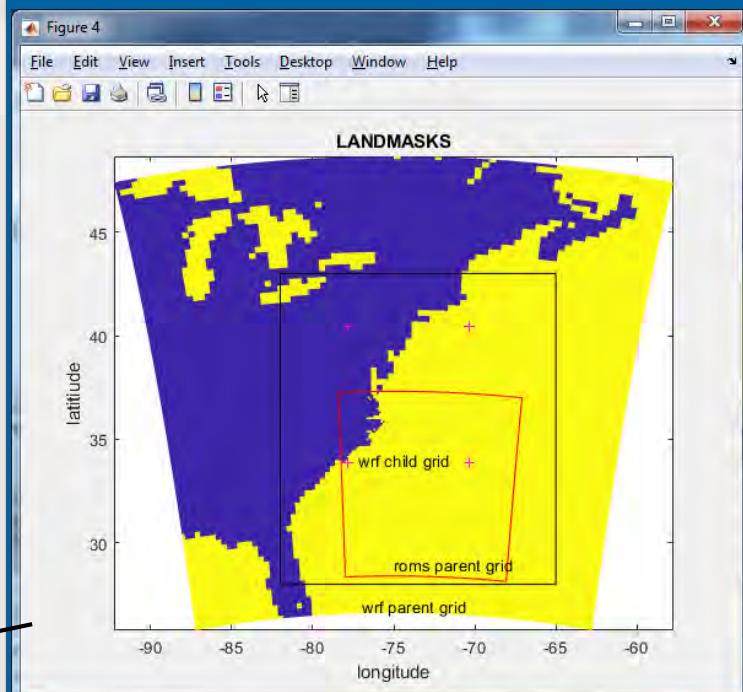
To create a nested child grid, we want to see how the 2 WRF and 2 ROMS grids will overlay.

```
netcdf_load('wrfinput_d01')
figure
pcolorjw(XLONG,XLAT,double(1-LANDMASK))
hold on
netcdf_load('wrfinput_d02')
pcolorjw(XLONG,XLAT,double(1-LANDMASK))
plot(XLONG(:,1),XLAT(:,1),'r')
plot(XLONG(end,:),XLAT(end,:),'r')
plot(XLONG(:,1),XLAT(:,1),'r')
plot(XLONG(:,end),XLAT(:,end),'r')
% plot roms parent grid
netcdf_load('Sandy_roms_grid.nc');
plot(lon_rho(:,1),lat_rho(:,1),'k')
plot(lon_rho(end,:),lat_rho(end,:),'k')
plot(lon_rho(:,1),lat_rho(:,1),'k')
plot(lon_rho(:,end),lat_rho(:,end),'k')
text(-75,29,'roms parent grid')
text(-77,27,'wrf parent grid')
text(-77.2,34,'wrf child grid')
title('LANDMASKS')
xlabel('longitude'); ylabel('latitude')
```

Select child indices and plot location of roms child grid.

```
Istr=22; Iend=60; Jstr=26; Jend=54;
plot(lon_rho(Istr,Jstr),lat_rho(Istr,Jstr),'m+')
plot(lon_rho(Istr,Jend),lat_rho(Istr,Jend),'m+')
plot(lon_rho(Iend,Jstr),lat_rho(Iend,Jstr),'m+')
plot(lon_rho(Iend,Jend),lat_rho(Iend,Jend),'m+')
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');
```

Page 55 of 27 22362 words 100% 100%



4) Child grid

COAWST_User_Manual [Compatibility Mode] - Word

File Home Insert Design Layout References Mailings Review View Tell me what you want to do... Sign in Share

Clipboard

Font Paragraph Styles Editing

4. ROMS child grid

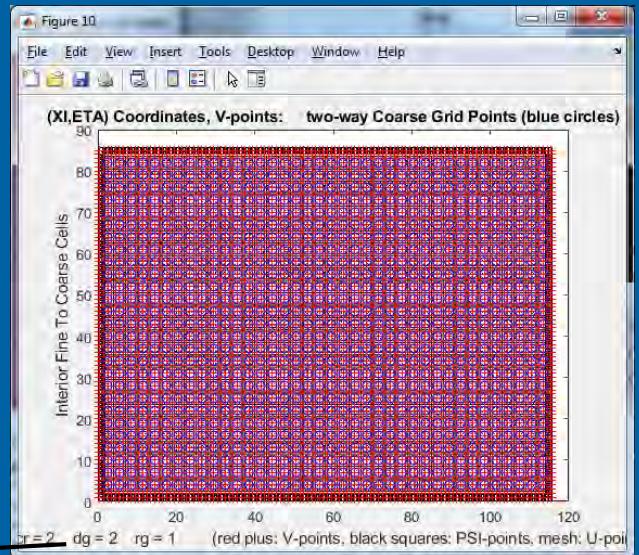
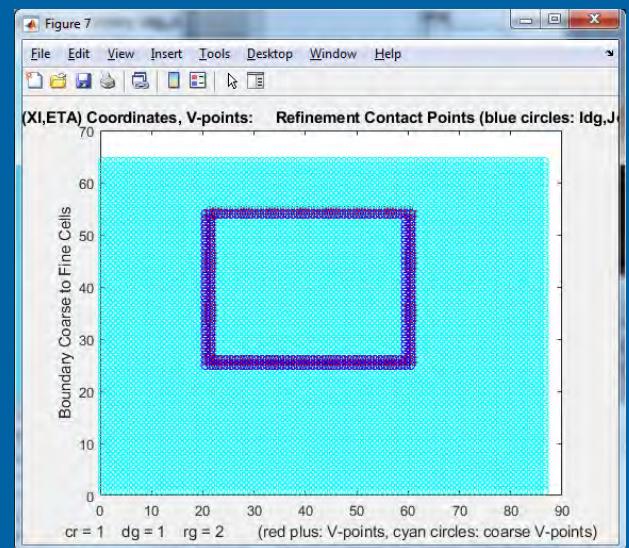
To create a nested child grid, we want to see how the 2 WRF and 2 ROMS grids will overlay.

```
netcdf_load('wrfinput_d01')
figure
pcolorjw(XLONG,XLAT,double(1-LANDMASK))
hold on
netcdf_load('wrfinput_d02')
pcolorjw(XLONG,XLAT,double(1-LANDMASK))
plot(XLONG(:,1),XLAT(:,1),'r');
plot(XLONG(end,:),XLAT(end,:),'r')
plot(XLONG(:,1),XLAT(:,1),'r');
plot(XLONG(:,end),XLAT(:,end),'r')
% plot roms parent grid
netcdf_load('Sandy_roms_grid.nc');
plot(lon_rho(1,:),lat_rho(1,:),'k');
plot(lon_rho(end,:),lat_rho(end,:),'k')
plot(lon_rho(:,1),lat_rho(:,1),'k');
plot(lon_rho(:,end),lat_rho(:,end),'k')
text(-75,29,'roms parent grid')
text(-77,27,'wrf parent grid')
text(-77.2,34,'wrf child grid')
title('LANDMASKS')
xlabel('longitude'); ylabel('latitude')
```

Select child indices and plot location of roms child grid.

```
Istr=22; Iend=60; Jstr=26; Jend=54;
plot(lon_rho(Istr,Jstr),lat_rho(Istr,Jstr),'m+')
plot(lon_rho(Istr,Jend),lat_rho(Istr,Jend),'m+')
plot(lon_rho(Iend,Jstr),lat_rho(Iend,Jstr),'m+')
plot(lon_rho(Iend,Jend),lat_rho(Iend,Jend),'m+')
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');
```

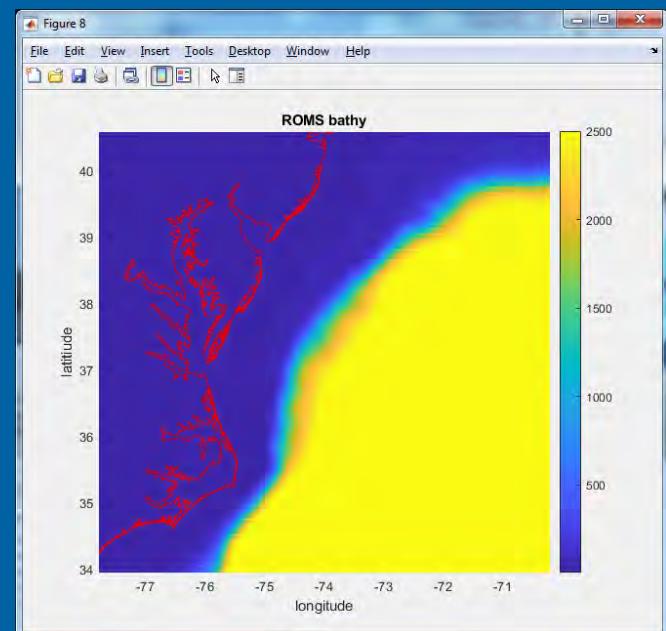
Page 55 of 27 22362 words 100% 150%



4) Child grid- bathy

Compute bathy for the child.

```
netcdf_load('Sandy_roms_grid_ref3.nc')
load USEast_bathy.mat
h=griddata(h_lon,h_lat,h_USeast,lon_rho,lat_rho);
h(isnan(h))=5;
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)+...
    h(2:end-1,1:end-2)+h(2:end-1,3:end));
figure
pcolorjw(lon_rho,lat_rho,h)
hold on
load coastline.mat
plot(lon,lat,'r')
caxis([5 2500]); colorbar
title('ROMS bathy')
xlabel('longitude'); ylabel('latitude')
ncwrite('Sandy_roms_grid_ref3.nc','h',h);
```



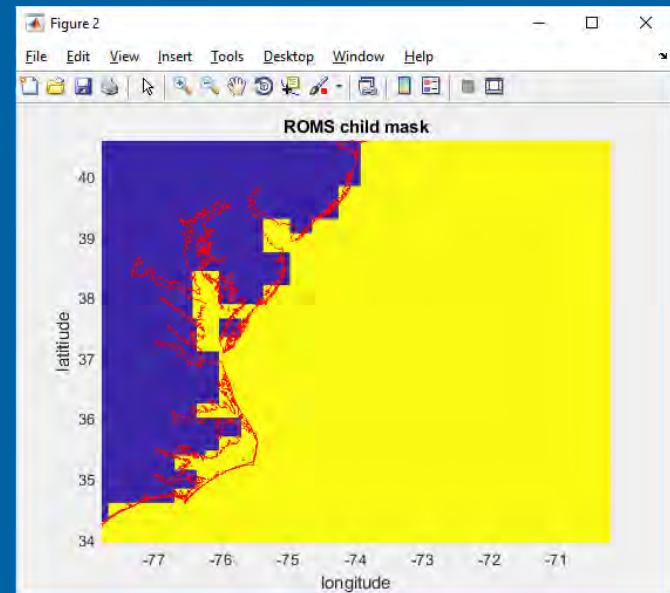
4) Child grid- masking

File Home Insert Design Layout References Mailings Review View Tell me what you want to do... Sign in Share

Clipboard Font Paragraph Styles Editing

```
Recompute child mask based on WRF mask
netcdf load('wrfinput_d01');
F = TriScatteredInterp(double(XLONG(:)),double(XLAT(:)), ...
    double(1-LANDMASK(:)),'nearest');
roms_mask=F(lon_rho,lat_rho);
figure
pcolorjw(lon_rho,lat_rho,roms_mask)
title('ROMS child mask')
xlabel('longitude'); ylabel('latitude')
hold on
plot(lon,lat,'r')
water = double(roms_mask);
u_mask = water(1:end-1,:) & water(2:end,:);
v_mask= water(:,1:end-1) & water(:,2:end);
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);
ncwrite('Sandy_roms_grid_ref3.nc','mask_rho',roms_mask);
ncwrite('Sandy_roms_grid_ref3.nc','mask_u',double(u_mask));
ncwrite('Sandy_roms_grid_ref3.nc','mask_v',double(v_mask));
ncwrite('Sandy_roms_grid_ref3.nc','mask_psi',double(psi_mask));
```

Page 56 of 88 22 of 22362 words 130%



You can use editmask again to make this nicer!

5) 3D BC's (u,v,temp,salt), init, clim

This step creates roms init conditions, bc's, and nudging files.

There are several tools out there. Here we will use

Tools/mfiles/mtools/roms_master_climatology_coawst_mw

This m file will create BC, IC, and Climatology for the parent grid.

Editor - C:\work\models\COAWST\Tools\mfiles\roms_clm\roms_master_climatology_coawst_mw.m

HYCOM

Inactive

```
% ocean_time=T1-datenum(1858,11,17,0,0,0)
% Updates from Christie Hegermiller, Feb 2019
%
% (1) Enter start date (T1) and number of days to get climatology data
T1 = datenum(2012,10,28,12,0,0); %start date
%number of days and frequency to create climatology files for
numdays = 5;
dayFrequency = 1;

% (2) Enter URL of the HYCOM catalog for the requested time, T1
% see http://tds.hycom.org/thredds/catalog.html
url = 'http://tds.hycom.org/thredds/dodsC/GLBa0.08/expt_90.9'; % 2011-01 to 2013-08

% (3) Enter working directory (wdr)
wdr = 'F:\data\models\COAWST_tests\coawstv3.4_update\coawst_v3.4_tests\sandy\Projects\Sandy';

% (4) Enter path and name of the ROMS grid
modelgrid = 'Sandy_roms_grid.nc'

% (5) Enter grid vertical coordinate parameters --These need to be consistent with the ROMS setup.
theta_s = 5.0;
theta_b = 0.4;
Tcline = 50.0;
N = 16;
Vtransform = 2;
Vstretching = 4;

%%%%%%%%%%%% END OF USER INPUT %%%%%%
eval(['cd ',wdr])
tic
```

5) 3D BC's (u,v,temp,salt), init, clim

Catalog <http://tds.hycom.org/thredds/catalog.html>

Dataset

- * Unaggregated *
- All_Data/
- GOFS 3.1: 41-layer HYCOM + NCODA Global 1/12° Analysis (NRL)
 - GLBy0.08/latest (Present + FORECASTS)/
 - GLBy0.08/expt_93.0 (Dec-04-2018 to Present + FORECASTS + ice + sur)/
 - GLBv0.08/latest (Present + FORECASTS)/
 - GLBv0.08/expt_93.0 (Jan-01-2018 to Present + FORECASTS)/
- INACTIVE-----
 - GLBu0.08/expt_93.0 (Sep-19-2018 to Dec-08-2018)/
 - GLBV0.08/expt_92.9 (Oct-01-2017 to Dec-31-2017)/
 - GLBv0.08/expt_57.7 (Jun-01-2017 to Sep-30-2017)/
 - GLBv0.08/expt_92.8 (Feb-01-2017 to May-31-2017)/
 - GLBv0.08/expt_57.2 (May-01-2016 to Jan-31-2017)/
 - GLBv0.08/expt_56.3 (Jul-01-2014 to Apr-30-2016)/
- GOFS 3.1: 41-layer HYCOM + NCODA Global 1/12° Reanalysis (NRL)
 - GLBy0.08/expt_53.X (Jan-01-1994 to Dec-30-2015)/
- HYCOM + NCODA Southeast United States 1/25° Analysis (HYCOM.org)
 - GOM10.04/expt_32.5 (2014-04 to Present) + FORECASTS/

Active



Catalog <http://tds.hycom.org/thredds/catalog.html>

-----INACTIVE-----

- GOM10.04/expt_31.0 (2009-04 to 2014-07)/
- HYCOM + NCODA Gulf of Mexico 1/25° Reanalysis (NRL)
- GOM10.04/expt_50.1 (1993-01-01 to 2012-12-31)/
- HYCOM + CFSR Gulf of Mexico 1/25° 54-year Experiment (COAPS)
- GOM10.04/expt_02.2 (1992 to 2009)/
- GOFS 3.0: HYCOM + NCODA Global 1/12° Analysis (HYCOM.org)
- INACTIVE-----
- GLBa0.08/expt_91.2 (2016-04 to 2018-11)/
- GLBa0.08/expt_91.1 (2014-04 to 2016-04)/
- GLBa0.08/expt_91.0 (2013-08 to 2014-04)/
- GLBa0.08/expt_90.9 (2011-01 to 2013-08)/
- GLBa0.08/expt_90.8 (2009-05 to 2011-01)/
- GLBa0.08/expt_90.6 (2008-09 to 2009-05)/
- GOFS 3.0: HYCOM + NCODA Global 1/12° Analysis (NRL)
- INACTIVE-----
- GLBu0.08/expt_91.2 (2016-04 to 2018-11)/
- GLBu0.08/expt_91.1 (2014-04 to 2016-04)/
- GLBu0.08/expt_91.0 (2013-08 to 2014-04)/
- GLBu0.08/expt_90.9 (2012-05 to 2013-08)/
- GOFS 3.0: HYCOM + NCODA Global 1/12° Reanalysis (NRL)
 - GLBu0.08/reanalysis_(expt_19.0_and_expt_19.1_combined)/
 - GLBu0.08/expt_19.0 (1992-10-02 to 1995-07-31)/
 - GLBu0.08/expt_19.1 (1995-08-01 to 2012-12-31)/

Inactive

5) 3D BC's (u,v,temp,salt), init, clim

The screenshot shows a Microsoft Word document window titled "COAWST_User_Manual [Compatibility Mode] - Word". The ribbon tabs visible are File, Home, Insert, Design, Layout, References, Mailings, Review, and View. The "Home" tab is selected. The main content area contains the following text:

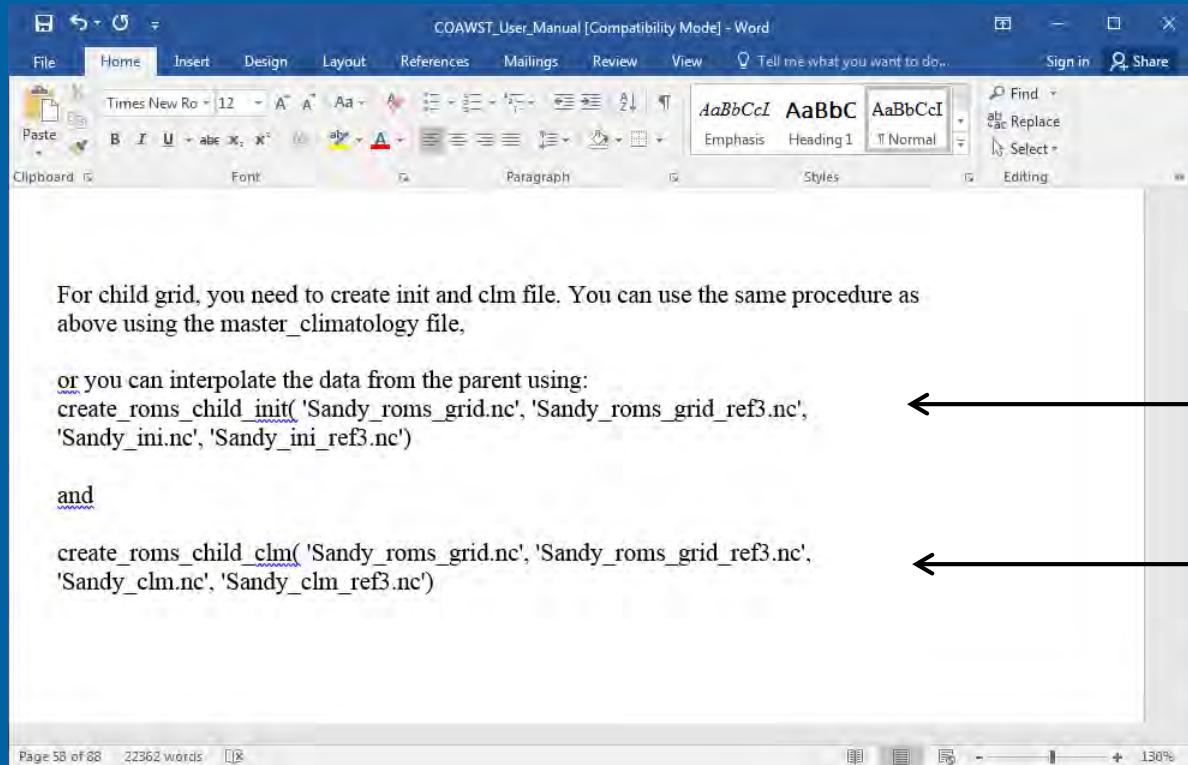
When you run this m file, it will create multiple files:

- Initial conditions: coawst_ini.nc
- Boundary conditions: coawst_bdy_20121028.nc, 29, 30, 31, 01 (5 files) and a merged_coawst_bdy.nc (all 5 of those merged into 1 file).
- Climatology: coawst_clm_20121028.nc, 29, 30, 31, 01 (5 files) and a merged_coawst_clm.nc (all 5 of those merged into 1 file).

I renamed:
coawst_ini.nc to Sandy_ini.nc
merged_coawst_bdy.nc to Sandy_bdy.nc
merged_coawst_clm.nc to Sandy_clm.nc

Page 50 of 88 22362 words 130%

5) For child grid need init and clm.



```
Editor > C:\work\models\COAWST\Tools\file\mdpool\create_roms_child_init.m  
FILE EDITOR PUBLISH VIEW  
New Open Save Compare Go To Comment Insert fx Breakpoints Run Run and Advance Run and Time  
FILE  
updateinit_cowst.mnw | roms_master_climatology_cowst.mnw | create_roms_child_init.m  
1 % Create a netcdf file that contains initialization data for a ROMS  
2 % child grid. It initializes temp, salt, u, v, ubar, vbar, and  
3 % all sediment parameters.  
4 %  
5 % parent_grid - input parent grid  
6 % child_grid - input child grid  
7 % parent_ini - input parent init file  
8 % child_ini - output child init file  
9 %  
10 % jow 5-25-2005  
11 % jow 3-7-07 add L and M for get_grid  
12 % jowarner adapt from Brandy.m file, 12Aug2014  
13 % jowarner 7/27/18 add time index for init file  
14 %  
15 % W-level RHO-level  
16 %  
17 %  
18 %  
19 %  
20 %  
21 %  
22 %  
23 %  
24 %  
25 %  
26 %  
27 %  
28 %  
29 %  
30 %  
31 %  
32 %  
33 %  
34 %
```

```
Editor > C:\work\models\COAWST\Tools\file\mtools\create_roms_child_clm.m  
FILE EDITOR PUBLISH VIEW  
New Open Save Compare Go To Comment Insert fx Breakpoints Run Run and Advance Run and Time  
FILE  
updateclm_cowst.mnw | updateinit_cowst.mnw | roms_master_climatology_cowst.mnw | create_roms_child_clm.m  
1 % Create a netcdf file that contains climatology data for a ROMS  
2 % child grid. Reads temp, salt, u, v, ubar, vbar, and  
3 % all sediment parameters.  
4 %  
5 % parent_grid - input parent grid  
6 % child_grid - input child grid  
7 % parent_clm - input parent climatology file  
8 % child_clm - output child climatology file  
9 %  
10 % jow 5-25-2005  
11 % jow 3-7-07 add L and M for get_grid  
12 % jowarner adapt from Brandy.m file, 12Aug2014  
13 %  
14 % W-level RHO-level  
15 %  
16 %  
17 %  
18 %  
19 %  
20 %  
21 %  
22 %  
23 %  
24 %  
25 %  
26 %  
27 %  
28 %  
29 %  
30 %  
31 %  
32 %  
33 %  
34 %
```

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

https://www.myroms.org/wiki/index.php/Tidal_Forcing

Description of Tidal Forcing in ROMS

Contents [hide]

1 Tidal Forcing Files in ROMS
2 OSU Tidal Prediction Software Example (Matlab)
3 ADCIRC Tidal Database Example (Matlab)
4 Comments on the 18.6 year tides

Tidal Forcing Files in ROMS

Once the appropriate CPP options have been set (e.g. `SSH_TIDES`, `UV_TIDES`, `RAMP_TIDES`), a netcdf file of tidal constituents must be generated.

1. **Download the desired tidal constituent database and associated software.** There are many tidal constituent databases available for download and the choice of databases depends on the desired constituents and location of the ROMS grid. Two possibilities which have broad geographic range and generally the dominant constituents are the [OSU Tidal Data Inversion Software](#) (OTIS) and the [ADCIRC tidal database](#). Remember to download the associated software with each data base as there are typically routines which facilitate the extraction and interpolation of the database to the ROMS grid.
2. **Interpolate the tidal constituent database to the desired ROMS grid.** See the above comment. While it is possible to write code (e.g. MATLAB, FORTRAN) to perform this task, it is typically easier to use the provided packages.
3. **Verify all open boundary grid cells contain valid data.** During the interpolation process, depending on the land mask for the ROMS grid and the tidal database grid, it is possible to have grid cells near land points which the ROMS grid may define as water, but the tidal grid defines as land. Should this occur a 180 degree phase shift along the open boundary near land points is possible. As ROMS is tidally forced on the open boundary, this could be problematic.
4. **The phase of the u/v/zeta components of the tidal constituents should be shifted to the appropriate reference time(t=to)** It is typical for tidal constituent databases to be stored with the phase shifted by the equilibrium tidal argument. Consequently the reference time for the tide is not the desired time, as set in `ocean.in` using the variable `TIDE_START`. In addition, nodal corrections need to be made due to long period tides. The equilibrium argument and nodal corrections can be calculated using the tidal database software or Rich Pawlowicz's [T_TIDE](#) MATLAB package. Also, see below for more thoughts on the 18.6 year business.
5. **Convert the amplitude/phase information to tidal ellipse parameters, if necessary.** ROMS requires tidal information to be stored as ellipse parameters for use. One tidal ellipse package is [ap2ep.m](#) from Zhigang Xu. MATLAB tidal ellipse code is available.
6. **Export tidal ellipse parameters to a ROMS netcdf forcing file.**

Two examples of this process are described below.

⚠ Newer Matlab versions are incompatible with older versions of `t_tide` and will cause errors. In particular, the "finite" function has been replaced with "isfinite". Rich Pawlowicz has provided an updated version of `t_tide` on his website http://www.eos.ubc.ca/~rich/#T_Tide. I have also provided a link here [\[1\]](#).

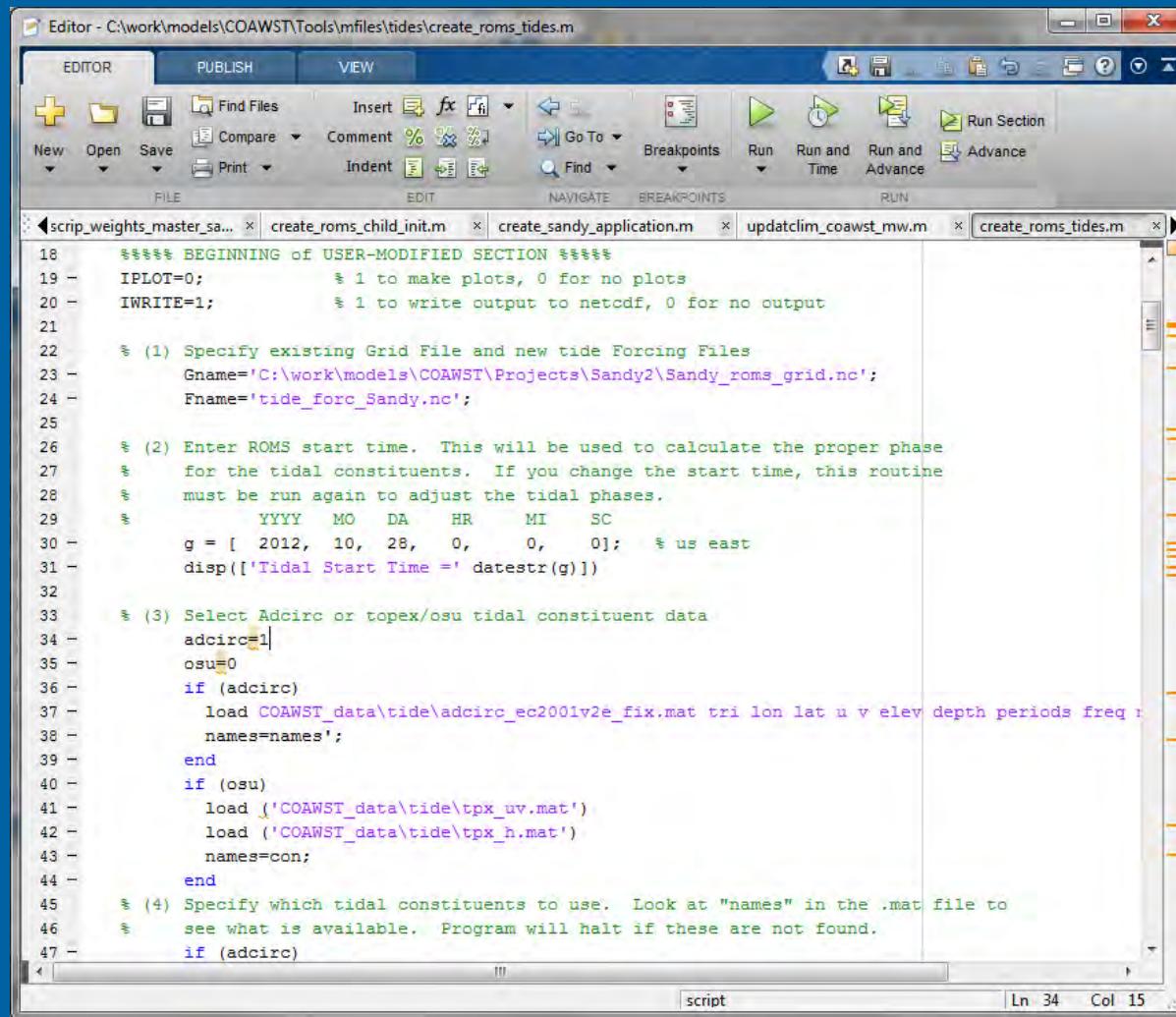
OSU Tidal Prediction Software Example (Matlab)

The processing in this example was carried out in MATLAB using routines found in <http://marine.rutgers.edu/~hunter/roms/tides/otps/>. It has been

last modified 14:48, 1 August 2011. accessed 12,558 times. Privacy policy About WikiROMS Powered By MediaWiki

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

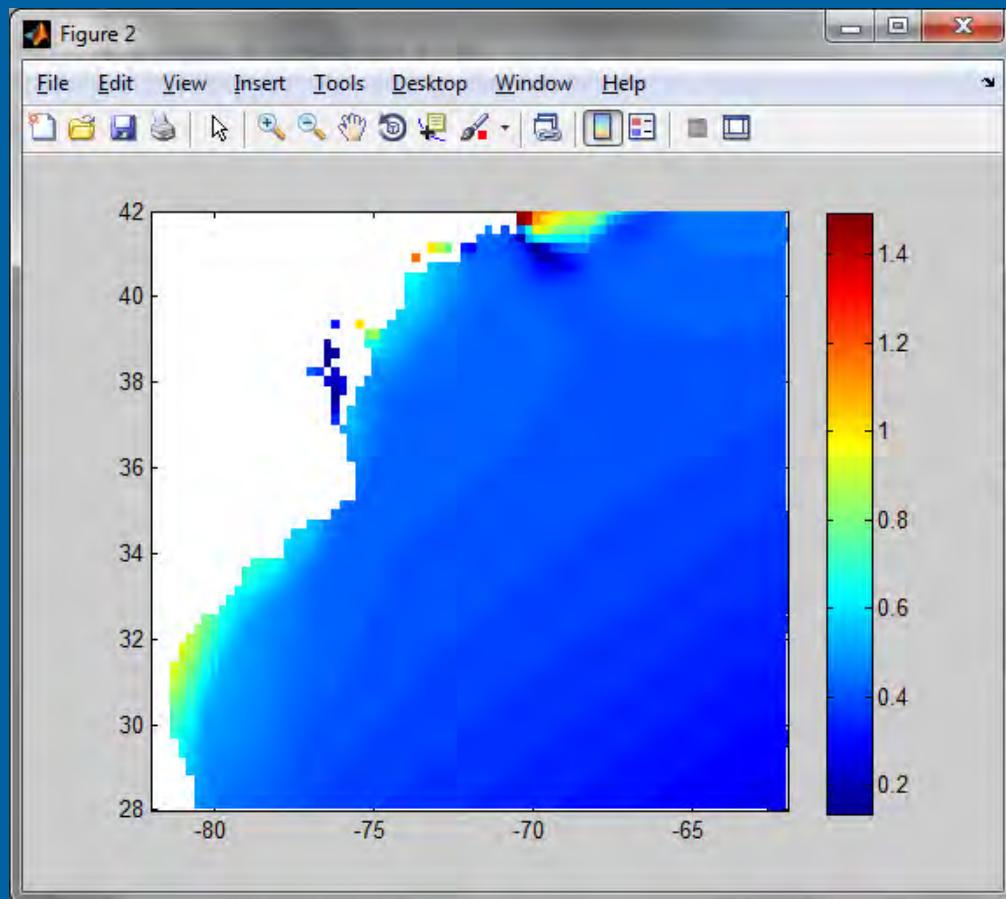


The screenshot shows a MATLAB editor window titled "Editor - C:\work\models\COAWST\Tools\mfiles\tides\create_roms_tides.m". The window has tabs for "EDITOR", "PUBLISH", and "VIEW". The toolbar includes buttons for New, Open, Save, Insert, Comment, Indent, Go To, Breakpoints, Run, Run and Time, Run and Advance, and Advance. The code in the editor is as follows:

```
18 %%%%% BEGINNING of USER-MODIFIED SECTION %%%%
19 - IPLOT=0; % 1 to make plots, 0 for no plots
20 - IWRITE=1; % 1 to write output to netcdf, 0 for no output
21
22 % (1) Specify existing Grid File and new tide Forcing Files
23 - Gname='C:\work\models\COAWST\Projects\Sandy2\Sandy_roms_grid.nc';
24 - Fname='tide_forc_Sandy.nc';
25
26 % (2) Enter ROMS start time. This will be used to calculate the proper phase
27 % for the tidal constituents. If you change the start time, this routine
28 % must be run again to adjust the tidal phases.
29 % YYYY MO DA HR MI SC
30 - g = [ 2012, 10, 28, 0, 0, 0]; % us east
31 - disp(['Tidal Start Time=' datestr(g)])
32
33 % (3) Select Adcirc or topex/osu tidal constituent data
34 - adcirc=1;
35 - osu=0;
36 - if (adcirc)
37 -     load COAWST_data\tide\adcirc_ec2001v2e_fix.mat tri lon lat u v elev depth periods freq r
38 -     names=names';
39 - end
40 - if (osu)
41 -     load ('COAWST_data\tide\tpx_uv.mat')
42 -     load ('COAWST_data\tide\tpx_h.mat')
43 -     names=con;
44 - end
45 % (4) Specify which tidal constituents to use. Look at "names" in the .mat file to
46 % see what is available. Program will halt if these are not found.
47 - if (adcirc)
```

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

1時限 9分30秒～

7. Surface forcings.

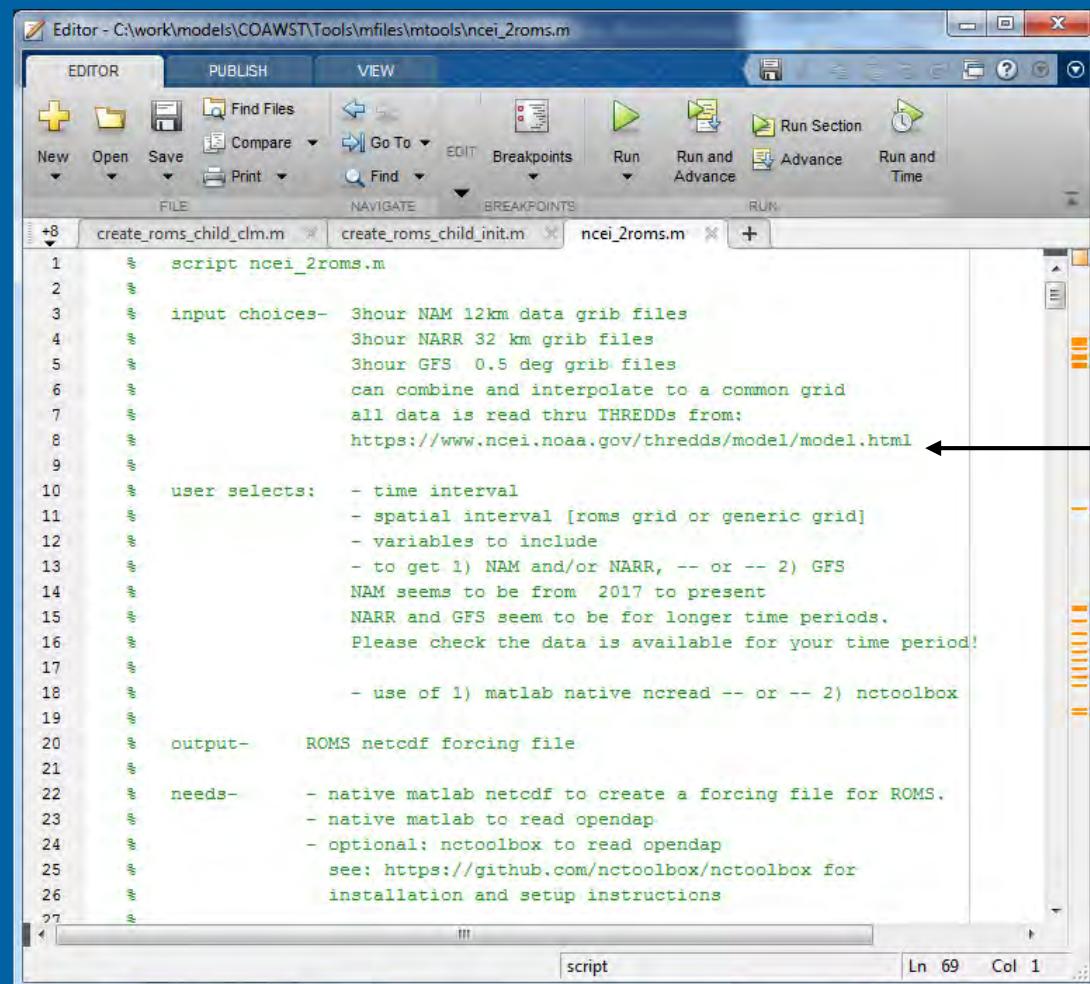
To create ROMS surface forcing files, there are several options such as:

- `create_roms_forcings.m`: converts data from matlab workspace to a netcdf forcing file.
- `narrnc2roms.m` : converts nc files from <ftp://cdc.noaa.gov/Datasets/NARR/monolevel> to netcdf.
- `ncei_2roms.m` : uses THREDDS to get data and create a forcing file.

We used `ncei_2roms` to create the `romsforc_NARR_Sandy2012.nc` for Sandy.

Many options. We will use `ncei_2roms.m`

7) Surface forcings – ncei_2roms.m

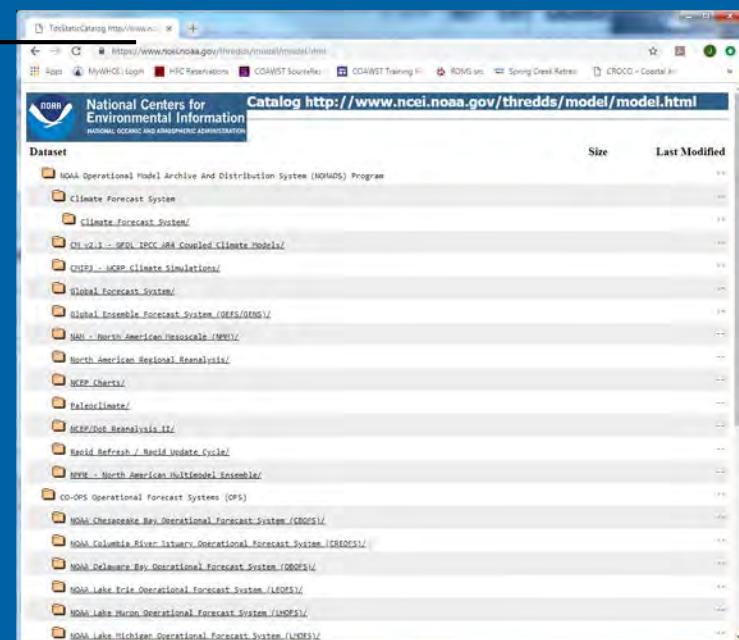


```
Editor - C:\work\models\COAWST\Tools\mfiles\mtools\ncei_2roms.m

EDITOR PUBLISH VIEW
FILE NAVIGATE BREAKPOINTS RUN
New Open Save Compare Go To EDIT Breakpoints Run Run and Advance Run and Time
Print Find +8 create_roms_child_clm.m create_roms_child_init.m ncei_2roms.m +
1 %> script ncei_2roms.m
2 %
3 %> input choices- 3hour NAM 12km data grib files
4 %> 3hour NARR 32 km grib files
5 %> 3hour GFS 0.5 deg grib files
6 %> can combine and interpolate to a common grid
7 %> all data is read thru THREDDs from:
8 %> https://www.ncei.noaa.gov/thredds/model/model.html
9 %
10 %> user selects:
11 %> - time interval
12 %> - spatial interval [roms grid or generic grid]
13 %> - variables to include
14 %> - to get 1) NAM and/or NARR, -- or -- 2) GFS
15 %> NAM seems to be from 2017 to present
16 %> NARR and GFS seem to be for longer time periods.
17 %> Please check the data is available for your time period!
18 %
19 %> - use of 1) matlab native ncread -- or -- 2) nettoolbox
20 %
21 %> output- ROMS netcdf forcing file
22 %
23 %> needs-
24 %> - native matlab netcdf to create a forcing file for ROMS.
25 %> - native matlab to read opendap
26 %> - optional: nettoolbox to read opendap
27 %> see: https://github.com/nctoolbox/nctoolbox for
%> installation and setup instructions
```

COAWST/Tools/mfiles/mtools/
ncei_2roms

creates netcdf forcing files for
ROMS.



7) Surface forcings – ncei_2roms.m

```
Editor - C:\work\models\COAWST\Tools\mfiles\mtools\ncei_2roms.m

EDITOR PUBLISH VIEW
FILE NAVIGATE EDIT BREAKPOINTS RUN
New Open Save Compare Go To Comment Breakpoints Run Run and Advance Run and Time
Find Print Indent Advanced Find
+6 updatinit_coawst_mw.m roms_master_climatology_coawst_mw.m create_roms_child_clm.m create_roms_child_init.m ncei_2roms.m +
```

```
51
52 %%%%%% START OF USER INPUT %%%%%%
53
54 % (1) Select which variables to include in this netcdf forcing file.
55 % put a '1' if you want to include it, '0' otherwise.
56 % 1/0 Var description (Units)
57 get_lwrad = 1; % gets downward and upward longwave and computes net flux of down-up (W/m2)
58 % this will also store lwrad_down so you can use LONGWAVE option.
59 get_swrad = 1; % gets downward and upward shortwave and computes net down-up flux (W/m2)
60 get_rain = 1; % precipitation rate (kg/m2/s) at surface
61 get_Tair = 1; % surface air temperature (C) at 2m
62 get_Pair = 1; % pressure reduce to MSL (Pa)
63 get_Qair = 1; % relative_humidity (percent) at 2m
64 get_Wind = 1; % surface u- and v- winds (m/s) at 10m
65
66 % (2) Enter name of output ROMS forcing file
67 %ROMS_force_name = 'romsforc_GFS_Sandy2012.nc';
68 ROMS_force_name = 'romsforc_NARR_Sandy2012.nc';
69
70 % (3) Enter start and end dates
71 time_start = datenum('28-Oct-2012');
72 time_end = datenum('31-Oct-2012');
73
74 % (4) Select Which data to obtain: NAM, NARR, both NAM+NARR -- or -- GFS.
75 get_NARR = 1; % NARR-A grid 221 32km data, available 1979-2014
76 get_NAM = 0; % NAM grid 218 12km data
77 % --- or ---
78 get_GFS = 0; % GFS 0.5 degree
79 % GFS is 6 hr and NAM/NARR is 3 hr. I dont have time interpolation
80 % added in so you have to pick NAM/NARR or GFS.
81
```

1) Select variables to be downloaded.

2) Forcing file name

3) Times

4) Data sets

7) Surface forcings – ncei_2roms.m

```
Editor - F:\data\models\COAWST\Tools\mfiles\mtools\ncei_2roms.m

EDITOR PUBLISH VIEW
FILE NAVIGATE EDIT BREAKPOINTS RUN
New Open Save Print Find Indent Breakpoints Run Run and Advance Run and Time
+4 create_inwave_zeta_bdy_working.m timeseries_plots.m compute_coawst_flux.m ncei_2roms.m +
73
74 % (4) Select which data to obtain: NAM, NARR, both NAM+NARR -- or -- GFS.
75 get_NARR = 1; % NARR-A grid 221 32km data, available 1979-2014
76 get_NAM = 0; % NAM grid 218 12km data
77 % --- or ---
78 get_GFS = 0; % GFS 0.5 degree
79 % GFS is 6 hr and NAM/NARR is 3 hr. I dont have time interpolation
80 % added in so you have to pick NAM/NARR or GFS.
81
82 % (5) Select to interpolate to a roms grid or a user defined grid.
83 % Set one of these to a 1, the other to a 0.
84 interpto_roms_grid = 0;
85 interpto_user_grid = 1; ← user_grid を選ぶ → nesting grid を利用
86
87 % The code has been written so that longitude is in -deg E.
88 if interpto_roms_grid
89     model_grid = 'C:\models\grid\USEast_grd31.nc';
90 elseif interpto_user_grid
91     % Provide lon_rho, lat_rho, and angle_rho.
92     % NAM / NARR grids are centered at~ -100 deg lon; GFS = 0:360 lon
93     if (get_NARR); offset=-360; end
94     if (get_NAM); offset=-360; end
95     if (get_GFS); offset=0; end
96     % You Probably want to make a finer resolution from 0.2 to 0.1.
97     lon_rho = [255:0.2:310]+offset; } 設定行
98     lat_rho = [ 10:0.2:50 ];
99     lon_rho = repmat(lon_rho,size(lat_rho,2),1)';
100    lat_rho = repmat(lat_rho',1,size(lon_rho,1))';
101    angle_rho = lon_rho*0; ← 設定行
102 else
103     disp('pick a grid')
104 end
105
106 % (6) Select which method to use: ncread or nctoolbox
107 use_matlab = 1;
108 use_nctoolbox = 0;
109
110 %%%%%%%%%%%%%% END OF USER INPUT %%%%%%%%%%%%%%
```

2 usages of "interpto_user_grid" found

script

Ln 90 Col 26

コード中の url の修正要
コード中の変数定義の修正要

5) User or model grid

6) Native tools or outside toolbox

8) ROMS input files: 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_OC2VV
|
#if defined WRF_MODEL && defined SWAN_MODEL
#define DRAGLIM_DAVIS
#define COARE_TAYLOR_YELLAND
#endif

#ifndef ROMS_MODEL
/* Physics + numerics */
#define UW_ADV
#define UV_COR
#define UV_VIS2
#define MIX_S_UV
#define TS_FIXED
#define TS_U3HADVECTION
#define TS_C4VADVECTION
#define TS_MPDATA
#define 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 */
#ifndef WRF_MODEL
#define BULK_FLUXES
#define ATM2CN_FLUXES
#define ANA_SSFLUX
#define LONGWAVE_OUT
#else
#define BULK_FLUXES
#endif
#define ATM_PRESS
#define ANA_BTFLUX
#define ANA_BSFLUX
#define ANA_BPFLUX
#define ANA_SFPLUX
#define ANA_SRFLUX
#define EMINUSP
#define SOLAR_SOURCE

/* Turbulence closure */
#define GLS_MIXING
#define 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 */
#define DIAGNOSTICS_UV
#define DIAGNOSTICS_TS
```

Tool Output

46 21 Read Ovr Block Sync Rec Caps

8) ROMS input files: sandy_ocean.in

```
! Application title.  
TITLE = Hurricane Sandy  
!  
! C-preprocessing Flag.  
MyAppCPP = SANDY  
!  
! Input variable information file name. This file needs to be processed  
! first so all information arrays can be initialized properly.  
VARNAME = ROMS/External/varinfo.dat  
!  
! Number of nested grids.  
Ngrids = 2  
!  
! Number of grid nesting layers. This parameter is used to allow refinement  
! and composite grid combinations.  
NestLayers = 2  
!  
! Number of grids in each nesting layer [1:NestLayers].  
GridsInLayer = 1 1  
!  
! Grid dimension parameters. See notes below in the Glossary for how to set  
! these parameters correctly.  
Lm == 85 114      ! Number of I-direction INTERIOR RHO-points  
Mm == 63 84      ! Number of J-direction INTERIOR RHO-points  
N == 16 16       ! Number of vertical levels  
ND == 0          ! Number of wave directional bins  
Nbed = 0          ! Number of sediment bed layers  
NAT = 2          ! Number of active tracers (usually, 2)  
NPT = 0          ! Number of inactive passive tracers  
NCS = 0          ! Number of cohesive (mud) sediment tracers  
NNS = 0          ! Number of non-cohesive (sand) sediment tracers  
!  
! Domain decomposition parameters for serial, distributed-memory or  
! shared-memory configurations used to determine tile horizontal range  
! indices (Istr,Iend) and (Jstr,Jend). [1:Ngrids].  
NtileI == 1        ! I-direction partition  
NtileJ == 1        ! J-direction partition  
!  
! Set lateral boundary conditions keyword. Notice that a value is expected  
! for each boundary segment per nested grid for each state variable.
```

Nesting params

Grid sizes

NtileI NtileJ

8) ROMS input files: 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

| Keyword Lateral Boundary Condition Type
| Cha Chapman_implicit (free-surface)
| Che Chapman_explicit (free-surface)
| Cla Clamped
| Clo Closed
| Fla Flather (2D momentum)
| Gra Gradient
| Nes Nested (refinement)
| Nud Nudging
| Per Periodic
| Rad Radiation
| Red Reduced Physics (2D momentum)
| Shc Shchepetkin (2D momentum)

| !  

|      W      S      E      N  

|      e      o      a      o  

|      s      u      s      r  

|      t      t      t      t  

|      h           h      h  

|      1      2      3      4  

| !  

|      N  

|      4  

|      j=Mm  

|      1      W      E      3  

|      i=1           i=Lm      j=1  

| !  

|      S  

|      2  

|      i=1           j=1  

| !  

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

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

|      Nes      Nes      Nes  

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


```

BC's

8) ROMS input files: sandy_ocean.in

The screenshot shows a TPad window displaying the contents of the `sandy_ocean.in` file. The file contains several sections of parameter assignments, each preceded by a ! symbol. Two annotations with arrows point to specific sections: one arrow points to the 'Time stepping' section, and another points to the 'nhis, navg, etc' section.

```
! Time-Stepping parameters.  
NTIMES == 5760    11520  
DT == 30.0d0 15.0d0  
NDTFAST == 28     28  
  
! Model iteration loops parameters.  
ERstr = 1  
ERend = 1  
Nouter = 1  
Ninner = 1  
Nintervals = 1  
  
! Number of eigenvalues (NEV) and eigenvectors (NCV) to compute for the  
! Lanczos/Arnoldi problem in the Generalized Stability Theory (GST)  
! analysis. NCV must be greater than NEV (see documentation below).  
NEV = 2           ! Number of eigenvalues  
NCV = 10          ! Number of eigenvectors  
  
! Input/Output parameters.  
NRREC == 0      0  
LcycleRST == T   T  
NRST == 60     120  
NSTA == 1       1  
NFLT == 1       1  
NINFO == 1      1  
  
! Output history, average, diagnostic files parameters.  
LDEFOUT == T    T  
NHIS == 60    120  
NDEFHIS == 0    0  
NTSAVG == 1    1  
NAVG == 432   432  
NDEFAVG == 0    0  
NTSDIA == 1    1  
NDIA == 432   432  
NDEFDIA == 0    0
```

8) ROMS input files: sandy_ocean.in

```
! Harmonic/biharmonic horizontal diffusion of tracer for nonlinear model
! and adjoint-based algorithms: [1:NAT+NPT,Ngrids].
    TN2 == 0.2d0  0.2d0  0.2d0  0.2d0      ! m2/s
    TN4 == 0.0d0  0.0d0                  ! m4/s
    ad_TN2 == 0.0d0  0.0d0                  ! m2/s
    ad_TN4 == 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                  ! 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

! Upper threshold values to limit vertical mixing coefficients computed
! from vertical mixing parameterizations. Although this is an engineering
! fix, the vertical mixing values inferred from ocean observations are
! rarely higher than this upper limit value.
    AKT_LIMIT == 1.0d-3 1.0d-3          ! m2/s
    AKV_LIMIT == 1.0d-3 1.0d-3          ! m2/s

! 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
```

Limiters for mixing terms

8) ROMS input files: sandy_ocean.in

```
TextPad - F:\data\models\COAWST\Projects\Sandy\ocean_sandy.in
File Edit Search View Tools Macros Configure Window Help
ocean_sandy.in
Find incrementally
 ocean_sandy.in

! Generic length-scale turbulence closure parameters.
GIS_P == 3.0d0          ! K-epsilon
GIS_M == 1.5d0
GIS_N == -1.0d0
GIS_Kmin == 7.6d-6
GIS_Pmin == 1.0d-12

GIS_CMU0 == 0.5477d0
GIS_C1 == 1.44d0
GIS_C2 == 1.92d0
GIS_C3M == -0.4d0
GIS_C3P == 1.0d0
GIS_SIGK == 1.0d0
GIS_SIGP == 1.30d0

! Constants used in surface turbulent kinetic energy flux computation.
CHARNOK_ALPHA == 1400.0d0      ! Charnok surface roughness
ZOS_HSIG_ALPHA == 0.5d0        ! roughness from wave amplitude
SZ_ALPHA == 0.25d0             ! roughness from wave dissipation
CRGBAN_CW == 100.0d0           ! Craig and Banner wave breaking
WEC_ALPHA == 0.0d0              ! 0: all wave dissip goes to break and
                                ! 1: all wave dissip goes to roller and

! Constants used in momentum stress computation.
RDRG == 3.0d-04               ! m/s
RDRG2 == 0.025d0                ! nondimensional
Zob == 0.02d0                  ! m
Zos == 0.02d0                  ! m

! Height (m) of atmospheric measurements for Bulk fluxes parameterization.
BLK_ZQ == 2.0d0                ! air humidity
BLK_ZT == 2.0d0                ! air temperature
BLK_ZW == 10.0d0                ! winds

! Minimum depth for wetting and drying.
DCRIT == 0.10d0                 ! m

! Various parameters.
WTYPE == 1
LEVSFRC == 15
IEVBFR == 1

! Set vertical, terrain-following coordinates transformation equation and
! stretching function (see below for details), [1:Ngrids].
Vtransform == 2                  ! transformation equation
Vstretching == 4                 ! stretching function

! Vertical S-coordinates parameters (see below for details), [1:Ngrids].
THETA_S == 5.0d0                 ! surface stretching parameter
THETA_B == 0.4d0                 ! bottom stretching parameter
TCLINE == 50.0d0                 ! critical depth (m)

333 | 12 | Read | Ovr | Block | Sync | Rec | Caps | ...
```

← GLS params

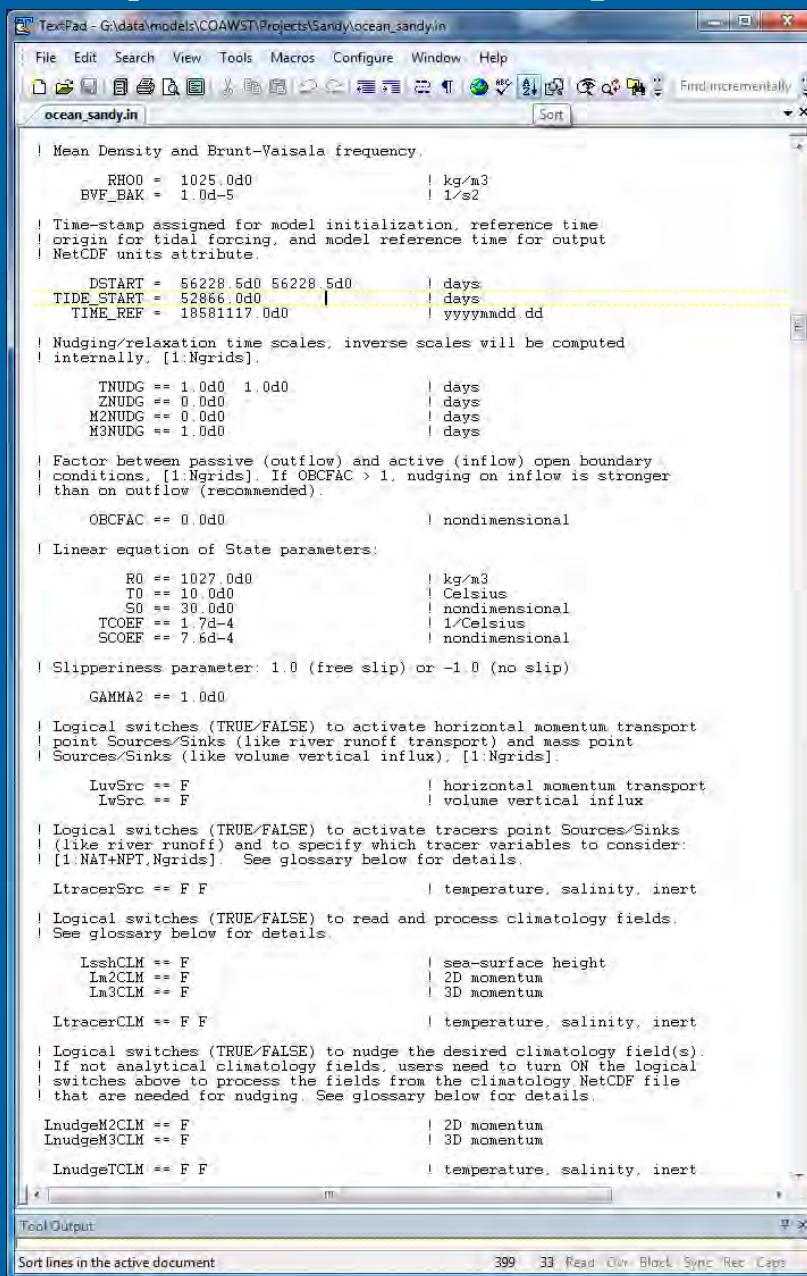
← Wave breaking params

← Bottom roughness

← Bulk flux heights

← Grid stretching params

8) ROMS input files: sandy_ocean.in



```
! Mean Density and Brunt-Vaisala frequency.
RHOO = 1025.0d0 ! kg/m3
BVF_BAK = 1.0d-5 ! 1/s2

! Time-stamp assigned for model initialization, reference time
! origin for tidal forcing, and model reference time for output
! NetCDF units attribute.
DSTART = 56228.5d0 56228.5d0 ! days
TIDE_START = 52866.0d0 ! days
TIME_REF = 18581117.0d0 ! yyyyymmdd dd

! Nudging/relaxation time scales, inverse scales will be computed
! internally, [1:Ngrids].
THUDG == 1.0d0 1.0d0 ! days
ZNUDG == 0.0d0 ! days
M2NUDG == 0.0d0 ! days
M3NUDG == 1.0d0 ! days

! Factor between passive (outflow) and active (inflow) open boundary
! conditions, [1:Ngrids]. If OBCFAC > 1, nudging on inflow is stronger
! than on outflow (recommended).
OBCFAC == 0.0d0 ! nondimensional

! Linear equation of State parameters:
R0 == 1027.0d0 ! kg/m3
T0 == 10.0d0 ! Celsius
S0 == 30.0d0 ! nondimensional
TCOEF == 1.7d-4 !/Celsius
SCOEF == 7.6d-4 ! nondimensional

! Slipperiness parameter: 1.0 (free slip) or -1.0 (no slip)
GAMMA2 == 1.0d0

! Logical switches (TRUE/FALSE) to activate horizontal momentum transport
! point Sources/Sinks (like river runoff transport) and mass point
! Sources/Sinks (like volume vertical influx), [1:Ngrids].
LuvSrc == F ! horizontal momentum transport
IvSrc == F ! volume vertical influx

! Logical switches (TRUE/FALSE) to activate tracers point Sources/Sinks
! (like river runoff) and to specify which tracer variables to consider:
! [1:NAT+NPT,Ngrids]. See glossary below for details.
ItracerSrc == F F ! temperature, salinity, inert

! Logical switches (TRUE/FALSE) to read and process climatology fields,
! See glossary below for details.
LsshCLM == F ! sea-surface height
Im2CLM == F ! 2D momentum
Im3CLM == F ! 3D momentum

ItracerCLM == F F ! temperature, salinity, inert

! Logical switches (TRUE/FALSE) to nudge the desired climatology field(s).
! If not analytical climatology fields, users need to turn ON the logical
! switches above to process the fields from the climatology NetCDF file
! that are needed for nudging. See glossary below for details.
InudgeM2CLM == F ! 2D momentum
InudgeM3CLM == F ! 3D momentum
InudgeTCLM == F F ! temperature, salinity, inert
```

← Time starts, tide start

← BC tnudge

8) ROMS input files: sandy_ocean.in

```
! Logical switches (TRUE/FALSE) to activate writing of fields into
! HISTORY output file.

Hout(idUvel) == T T      | u           3D U-velocity
Hout(idVvel) == T T      | v           3D V-velocity
Hout(idu3dE) == F F      | u_eastward 3D U-eastward at RHO-points
Hout(idv3dN) == F F      | v_northward 3D V-northward at RHO-points
Hout(idWvel) == T T      | w           3D W-velocity
Hout(idOvel) == T T      | omega       omega vertical velocity
Hout(idUbar) == T T      | ubar        2D U-velocity
Hout(idVbar) == T T      | vbar        2D V-velocity
Hout(idu2dE) == F F      | ubar_eastward 2D U-eastward at RHO-points
Hout(idv2dN) == F F      | vbar_northward 2D V-northward at RHO-points
Hout(idFsur) == T T      | zeta        free-surface
Hout(idBath) == T T      | bath         time-dependent bathymetry

Hout(idTvar) == T T T T ! temp, salt   temperature and salinity

Hout(idpthR) == F        | z_rho       time-varying depths of RHO-points
Hout(idpthU) == F        | z_u         time-varying depths of U-points
Hout(idpthV) == F        | z_v         time-varying depths of V-points
Hout(idpthW) == F        | z_w         time-varying depths of W-points

! Logical switches (TRUE/FALSE) to activate writing of time-averaged
! fields into AVERAGE output file.

Aout(idUvel) == F        | u           3D U-velocity
Aout(idVvel) == F        | v           3D V-velocity
Aout(idu3dE) == F        | u_eastward 3D U-eastward at RHO-points
Aout(idv3dN) == F        | v_northward 3D V-northward at RHO-points
Aout(idWvel) == F        | w           3D W-velocity
Aout(idOvel) == F        | omega       omega vertical velocity
Aout(idUbar) == F        | ubar        2D U-velocity
Aout(idVbar) == F        | vbar        2D V-velocity
Aout(idu2dE) == F        | ubar_eastward 2D U-eastward at RHO-points
Aout(idv2dN) == F        | vbar_northward 2D V-northward at RHO-points
Aout(idFsur) == F        | zeta        free-surface
Aout(idBath) == F        | bath         time-dependent bathymetry

! Logical switches (TRUE/FALSE) to activate writing of time-averaged,
! 2D momentum (ubar, vbar) diagnostic terms into DIAGNOSTIC output file.

Dout(M2rate) == F        | ubar_accel, ... acceleration
Dout(M2pgrd) == F        | ubar_prgrd, ... pressure gradient
Dout(M2fcor) == F        | ubar_cori, ... Coriolis force
Dout(M2hadv) == F        | ubar_hadv, ... horizontal total advection
Dout(M2xadv) == F        | ubar_xadv, ... horizontal XI-advection
Dout(M2yadv) == F        | ubar_yadv, ... horizontal ETA-advection
Dout(M2hrad) == F        | ubar_hrad, ... horizontal total wec_mellor radiation stress
Dout(M2hvis) == F        | ubar_hvisc, ... horizontal total viscosity
Dout(M2xvis) == F        | ubar_xvisc, ... horizontal XI-viscosity
Dout(M2yvis) == F        | ubar_yvisc, ... horizontal ETA-viscosity
Dout(M2sstr) == F        | ubar_sstr, ... surface stress
Dout(M2bstr) == F        | ubar_bstr, ... bottom stress
Dout(M2hjvf) == F        | 2D_wec_vf horizontal J vortex force
Dout(M2kvvf) == F        | 2D_wec_vf K vortex force
Dout(M2fsco) == F        | 2D_wec_vf coriolis-stokes

Tool Output.
```

History

Averages

Diagnostics

8) ROMS input files: sandy_ocean.in

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

File Edit Search View Tools Macros Configure Window Help

ocean_sandy.in

```
! Input NetCDF file names. [1:Ngrids].  
GRDNAME == Projects/Sandy/Sandy_roms_grid.nc \  
          Projects/Sandy/Sandy_roms_grid_ref3.nc  
ININAME == Projects/Sandy/Sandy_ini.nc \  
          Projects/Sandy/Sandy_ini_ref3.nc  
ITLNAME == ocean_itl.nc  
IRPNAME == ocean_irp.nc  
IADNAME == ocean_iad.nc  
FWDNAME == ocean_fwd.nc  
ADSNAME == ocean_ads.nc  
IWININAME == inwave_ini.nc  
IWSWNNAME == point1.spc2d  
  
! Nesting grids connectivity data: contact points information. This  
! NetCDF file is special and complex. It is currently generated using  
! the script "matlab/grid/contact.m" from the Matlab repository.  
  
NGCNAME = Projects/Sandy/Sandy_roms_contact.nc  
  
! Input lateral boundary conditions and climatology file names. The  
! USER has the option to split input data time records into several  
! NetCDF files (see prologue instructions above). If so, use a single  
! line per entry with a vertical bar (|) symbol after each entry,  
! except the last one.  
  
NCLMFILES == 1           ! number of climate files  
CLMNAME == Projects/Sandy/Sandy_clm.nc \  
          Projects/Sandy/Sandy_clm_ref3.nc  
  
NBCFILES == 1           ! number of boundary files  
BRYNAME == Projects/Sandy/Sandy_bdy.nc  
BRYNAME == Projects/Sandy/coawst_bdy_20121028.nc | } ニューファイル  
          Projects/Sandy/coawst_bdy_20121029.nc | } ニューファイル  
          Projects/Sandy/coawst_bdy_20121030.nc | } ニューファイル  
          Projects/Sandy/coawst_bdy_20121031.nc | } ニューファイル  
          Projects/Sandy/coawst_bdy_20121101.nc  
  
! Input climatology nudging coefficients file name.  
NUDNAME == ocean_nud.nc  
  
! Input Sources/Sinks forcing (like river runoff) file name.  
SSFNAME == ocean_rivers.nc
```

Different in COAWST

8) ROMS input files: sandy_ocean.in

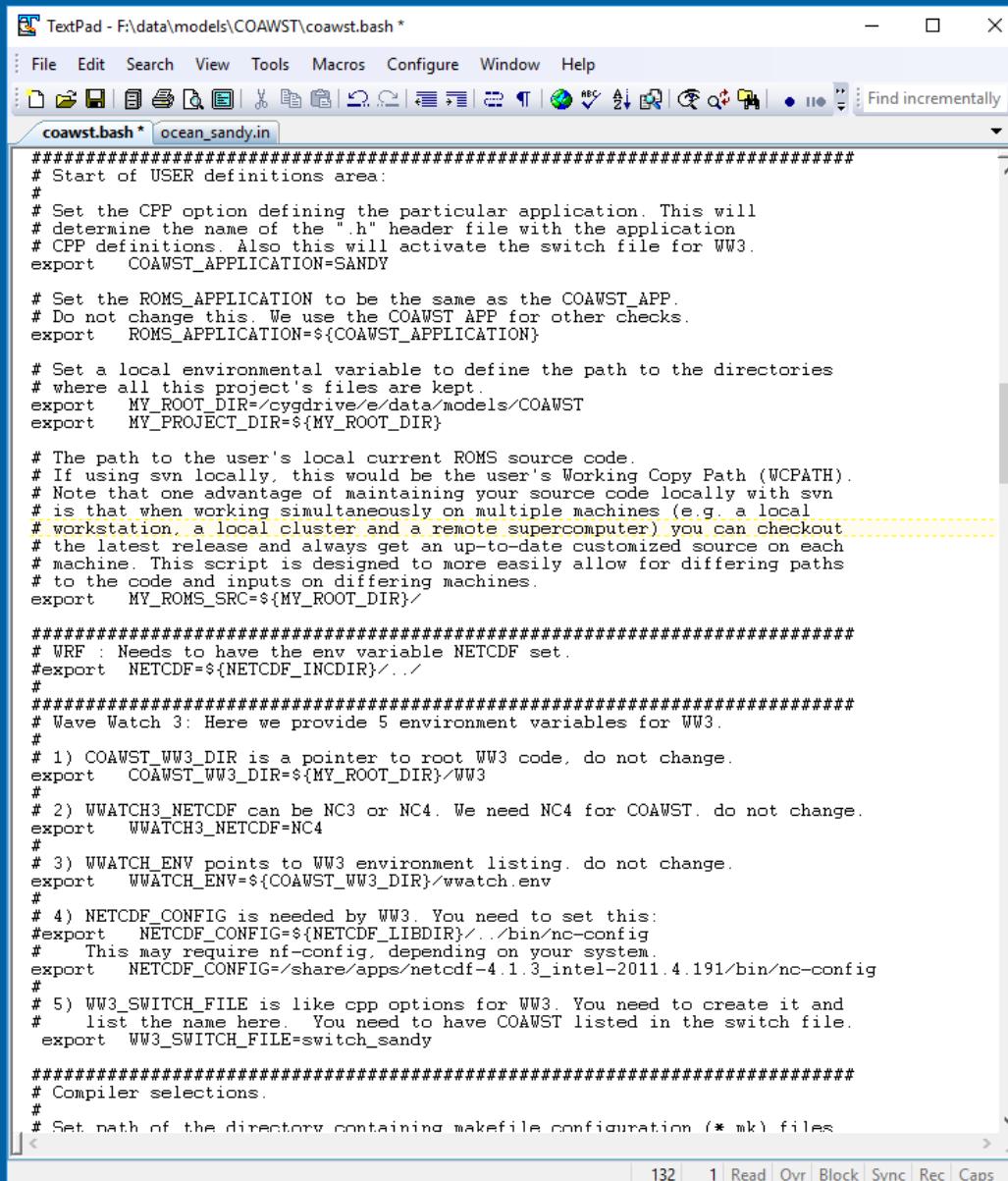
```
TextPad - F:\data\models\COAWST\Projects\Sandy\ocean_sandy.in
File Edit Search View Tools Macros Configure Window Help
ocean_sandy.in
! Input tidal forcing file name.
TIDENAME == Projects/Sandy/tide_forc_Sandy.nc
! Input forcing NetCDF file name(s).
! The USER has the option to enter several sets of file names for each
! nested grid. For example, the USER may have different data for the
! wind products, heat fluxes, etc. Alternatively, if the all the forcing
! files are the same for nesting and the data is in its native resolution,
! we could enter only one set of files names and ROMS will replicate those
! files internally to the remaining grids using the plural KEYWORD protocol.
The model will scan the files and will read the needed data from the first
file in the list containing the forcing field. Therefore, the order of the
filenames is critical. If using multiple forcing files per grid, first
enter all the file names for grid one followed by two, and so on. It is
also possible to split input data time records into several NetCDF files
(see Prolog instructions above). Use a single line per entry with a
continuation (\) or a vertical bar (|) symbol after each entry, except
the last one.
NFFILES == 1
FRCNAME == Projects/Sandy/romsforc_NARR_Sandy2012.nc \
Projects/Sandy/romsforc_NARR_Sandy2012.nc
! Output NetCDF file names, [1:Ngrids].
DAINAME == ocean_dai.nc
GSTNAME == ocean_gst.nc
RSTNAME == Sandy_ocean_rst.nc \
Sandy_ocean_ref3_rst.nc
HISNAME == Sandy_ocean_his.nc \
Sandy_ocean_ref3_his.nc
QCKNAME == ocean_qck.nc
TLMNAME == ocean_tlm.nc
TLFNAME == ocean_tlf.nc
ADJNAME == ocean_adj.nc
AVGNAME == Sandy_ocean_avg.nc \
Sandy_ocean_ref3_avg.nc
HARNAME == ocean_har.nc
DIANAME == ocean_dia.nc
STANAME == ocean_sta.nc
FLTNAME == ocean_flt.nc
! Input ASCII parameter filenames.
APARNAM = ROMS/External/s4dvar.in
SPOSNAM = ROMS/External/stations.in
FPSONAM = ROMS/External/floats.in
BPARNAM = ROMS/External/bioFasham.in
SPARNAM = ROMS/External/sediment.in
USRNAME = ROMS/External/MyFile.dat
VEGNAM = vegetation.in
```

Only list tide forc for parent

You can use same forcing file for
both grids

his and avg names, etc

9) Build and run: coawst.bash



The screenshot shows a TextPad editor window with the file 'coawst.bash' open. The code is a shell script for building and running COAWST. It includes comments explaining the purpose of various environment variables and configuration steps. A yellow dashed line highlights a section of the code related to source code checkout paths.

```
TextPad - F:\data\models\COAWST\coawst.bash *
File Edit Search View Tools Macros Configure Window Help
coawst.bash * ocean_sandy.in
#####
# Start of USER definitions area:
#
# Set the CPP option defining the particular application. This will
# determine the name of the ".h" header file with the application
# CPP definitions. Also this will activate the switch file for WW3.
export COAWST_APPLICATION=SANDY

# Set the ROMS_APPLICATION to be the same as the COAWST_APP.
# Do not change this. We use the COAWST APP for other checks.
export ROMS_APPLICATION=${COAWST_APPLICATION}

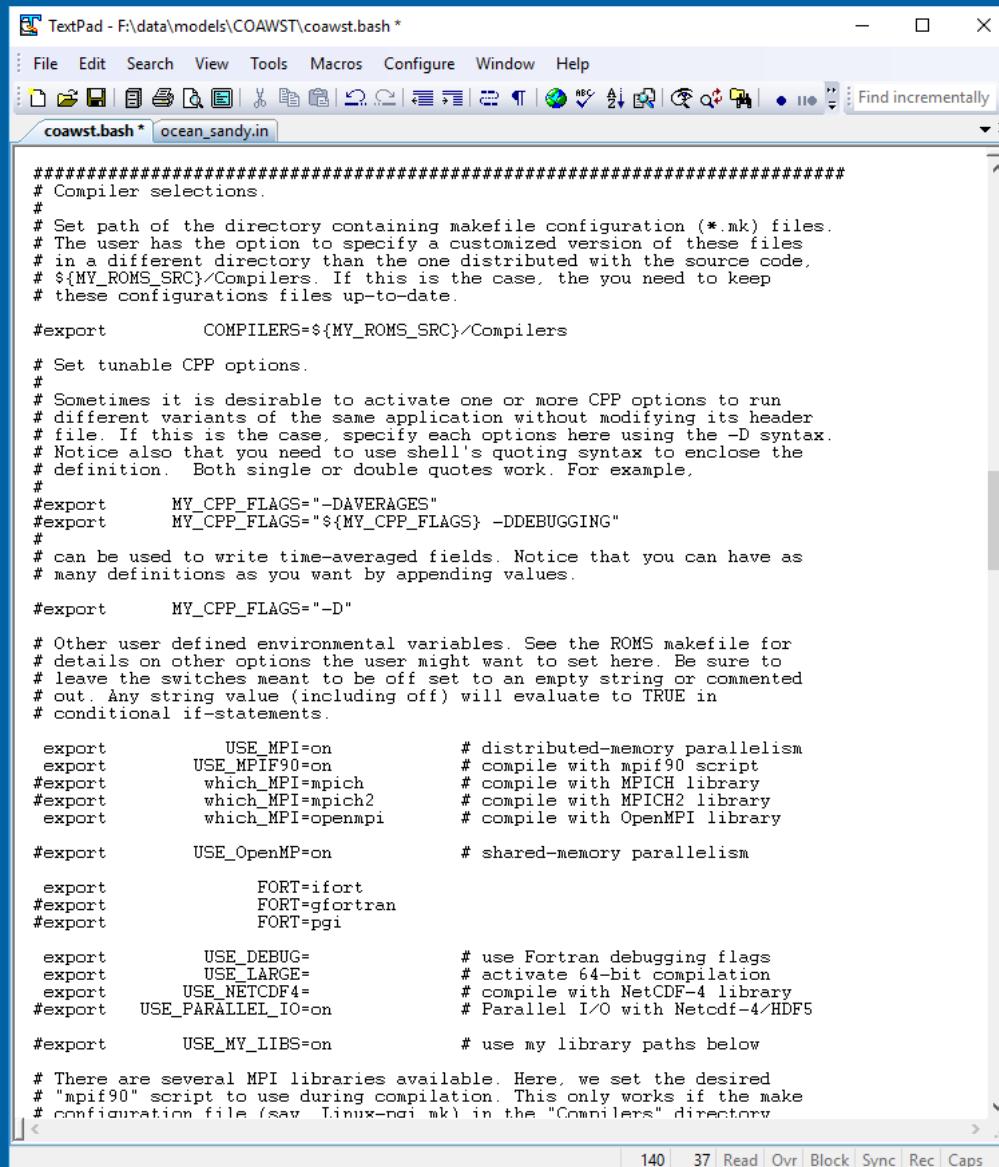
# Set a local environmental variable to define the path to the directories
# where all this project's files are kept.
export MY_ROOT_DIR=/cygdrive/e/data/models/COAWST
export MY_PROJECT_DIR=${MY_ROOT_DIR}

# The path to the user's local current ROMS source code.
# If using svn locally, this would be the user's Working Copy Path (WCPATH).
# Note that one advantage of maintaining your source code locally with svn
# is that when working simultaneously on multiple machines (e.g. a local
# workstation, a local cluster and a remote supercomputer) you can checkout
# the latest release and always get an up-to-date customized source on each
# machine. This script is designed to more easily allow for differing paths
# to the code and inputs on differing machines.
export MY_ROMS_SRC=${MY_ROOT_DIR}/

#####
# WRF : Needs to have the env variable NETCDF set.
#export NETCDF=${NETCDF_INCDIR}/..
#
#####
# Wave Watch 3: Here we provide 5 environment variables for WW3.
#
# 1) COAWST_WW3_DIR is a pointer to root WW3 code, do not change.
export COAWST_WW3_DIR=${MY_ROOT_DIR}/WW3
#
# 2) WWATCH3_NETCDF can be NC3 or NC4. We need NC4 for COAWST, do not change.
export WWATCH3_NETCDF=NC4
#
# 3) WWATCH_ENV points to WW3 environment listing, do not change.
export WWATCH_ENV=${COAWST_WW3_DIR}/wwatch.env
#
# 4) NETCDF_CONFIG is needed by WW3. You need to set this:
#export NETCDF_CONFIG=${NETCDF_LIBDIR}/../bin/nc-config
# This may require nf-config, depending on your system.
export NETCDF_CONFIG=/share/apps/netcdf-4.1.3_intel-2011.4.191/bin/nc-config
#
# 5) WW3_SWITCH_FILE is like cpp options for WW3. You need to create it and
# list the name here. You need to have COAWST listed in the switch file.
export WW3_SWITCH_FILE=switch_sandy

#####
# Compiler selections.
#
# Set path of the directory containing makefile configuration (*.mk) files
```

9) Build and run: coawst.bash



The screenshot shows a TextPad editor window with the file 'coawst.bash' open. The code is a shell script that sets various environmental variables for building COAWST. It includes sections for compiler selections, tunable CPP options, and other user-defined variables like MPI and FORT compilers. A note at the bottom explains the use of 'mpif90' script during compilation.

```
#####
# Compiler selections.
#
# Set path of the directory containing makefile configuration (*.mk) files.
# The user has the option to specify a customized version of these files
# in a different directory than the one distributed with the source code,
# ${MY_ROMS_SRC}/Compilers. If this is the case, the you need to keep
# these configurations files up-to-date.

#export      COMPILERS=${MY_ROMS_SRC}/Compilers

# Set tunable CPP options.
#
# Sometimes it is desirable to activate one or more CPP options to run
# different variants of the same application without modifying its header
# file. If this is the case, specify each options here using the -D syntax.
# Notice also that you need to use shell's quoting syntax to enclose the
# definition. Both single or double quotes work. For example,
#
#export      MY_CPP_FLAGS="-DAVERAGES"
#export      MY_CPP_FLAGS="${MY_CPP_FLAGS} -DDEBUGGING"
#
# can be used to write time-averaged fields. Notice that you can have as
# many definitions as you want by appending values.

#export      MY_CPP_FLAGS="-D"

# Other user defined environmental variables. See the ROMS makefile for
# details on other options the user might want to set here. Be sure to
# leave the switches meant to be off set to an empty string or commented
# out. Any string value (including off) will evaluate to TRUE in
# conditional if-statements.

export      USE_MPI=on          # distributed-memory parallelism
export      USE_MPIF90=on        # compile with mpif90 script
#export      which_MPI=mpich     # compile with MPICH library
#export      which_MPI=mpich2    # compile with MPICH2 library
#export      which_MPI=openmpi   # compile with OpenMPI library

#export      USE_OpenMP=on       # shared-memory parallelism

export      FORT=ifort
#export      FORT=qfortran
#export      FORT=pgi

export      USE_DEBUG=           # use Fortran debugging flags
export      USE_LARGE=           # activate 64-bit compilation
export      USE_NETCDF4=          # compile with NetCDF-4 library
#export      USE_PARALLEL_IO=on   # Parallel I/O with Netcdf-4/HDF5

#export      USE_MY_LIBS=on      # use my library paths below

# There are several MPI libraries available. Here, we set the desired
# "mpif90" script to use during compilation. This only works if the make
# configuration file (say Linux-ifort.mk) in the "Compilers" directory
```

Makefile determines the
Compilers file from:

- 1) uname -s
- 2) the FORT value

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

9) Build and run: coawst.bash

```
TextPad - F:\data\models\COAWST\coawst.bash *
File Edit Search View Tools Macros Configure Window Help
Find incrementally
coawst.bash * ocean_sandy.in

#####
# Header and other source directories selections.
#
# The rest of this script sets the path to the users header file and
# analytical source files, if any. See the templates in User/Functionals.
#
# If applicable, use the MV_ANALYTICAL_DIR directory to place your
# customized biology model header file (like fennel.h, nemuro.h, ecosim.h,
# etc).
#
# export      MY_HEADER_DIR=${MY_PROJECT_DIR}/ROMS/Include
# export      MY_ANALYTICAL_DIR=${MY_PROJECT_DIR}/ROMS/Functionals
# export      MY_HEADER_DIR=${MY_PROJECT_DIR}/Projects/Inlet_test/Coupled
# export      MV_ANALYTICAL_DIR=${MY_PROJECT_DIR}/Projects/Inlet_test/Coupled

# Put the binary to execute in the following directory.
#
# export      BINDIR=${MY_PROJECT_DIR}
# export      BINDIR=.

# Put the f90 files in a project specific Build directory to avoid conflict
# with other projects.

# export      SCRATCH_DIR=${MY_PROJECT_DIR}/Build
# export      SCRATCH_DIR=../Build

# Go to the users source directory to compile. The options set above will
# pick up the application-specific code from the appropriate place.

#####
# End of USER definitions area. You really should not change anything
# down here.
#
# cd ${MY_ROMS_SRC}

#-----
# Compile.
#-----
# Remove build directory.

if [ $clean -eq 1 ]; then
    make clean
fi

# Compile (the binary will go to BINDIR set above).

export WRF_DIR=${MY_ROMS_SRC}/WRF
if [ $cleanwrf -eq 1 ]; then
    make wfclean
    cd ${MY_ROMS_SRC}
fi
make wrf

if [ $cleanw3 -eq 1 ]; then
    make ww3clean
    cd ${MY_ROMS_SRC}
fi
make ww3

if [ $dprint -eq 1 ]; then
    make $debug
else
    if [ $parallel -eq 1 ]; then
        make $NCpus
    else
        make
    fi
fi
```

Set the header and ana paths

Customize the build here

build options

To build the code use
./coawst.bash -j N
this should make the
coawstM

9) Build and run: run_coawst

```
emacs: run_coawst@peach
File Edit View Cmds Tools Options Buffers Insert Help
Open Dired Save Print Cut Copy Paste Undo Replace Mail Info Compile Debug News
coawst.bash run_coawst
#!/bin/bash
### Job name
#PBS -N sandy
### Number of nodes
#PBS -l nodes=3:ppn=8,walltime=120:00:00
### Mail to user
#PBS -m ae
#PBS -M jcwarner@usgs.gov
### Out files
###PBS -e isabel_105.err
###PBS -o isabel_105.out
### PBS queue
###PBS -q standard

umask 0002

echo "this job is running on."
cat $PBS_NODEFILE

NPROCS=`wc -l < $PBS_NODEFILE` ←
cd /peach/data0/jcwarner/Projects/sandy_for_workshop

mpirun -np 24 -machinefile $PBS_NODEFILE ./coawstM Projects/Sandy/ocean_sandy.in > cwstv3.out
#mpirun -np 16 -machinefile $PBS_NODEFILE ./coawstM Projects/Sandy/swan_sandy.in Projects/Sandy/swan_sand
dy_ref3.in > cwstv3.out
#mpirun -np 16 -machinefile $PBS_NODEFILE ./coawstM namelist.input > sandy.out
#mpirun -np 32 -machinefile $PBS_NODEFILE ./coawstM Projects/Sandy/coupling_sandy.in > sandy.out
#mpirun -np 48 -machinefile $PBS_NODEFILE ./coawstM Projects/Sandy/coupling_sandy.in > sandy.out

IS08--**-XEmacs: run_coawst (Shell-script Font[bash]) ----L5--C15--All-----
```

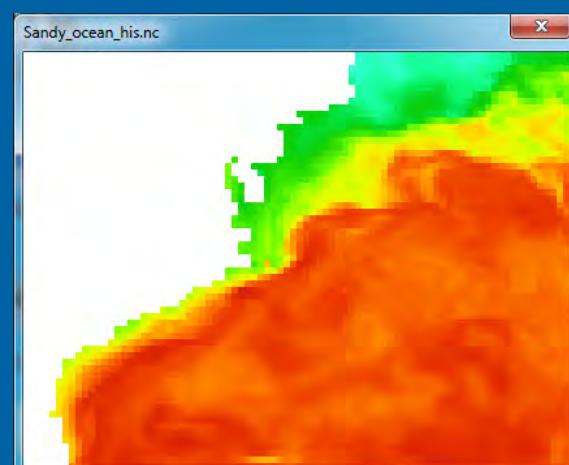
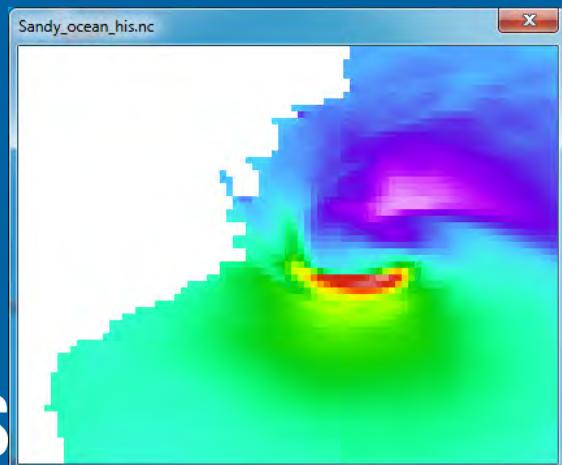
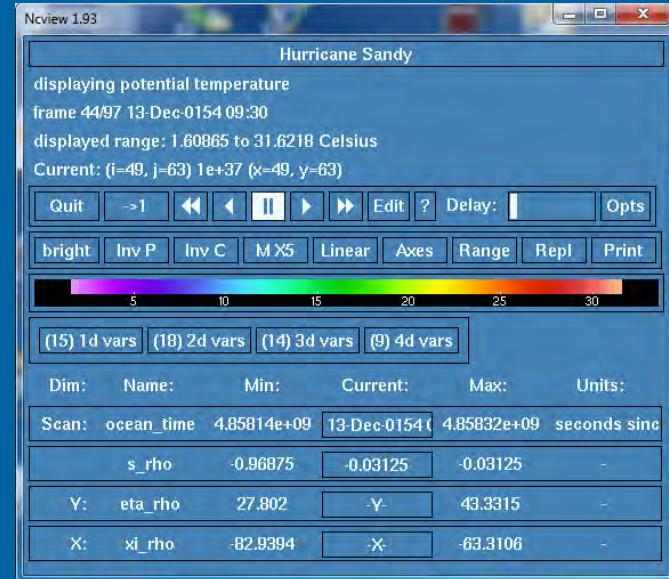
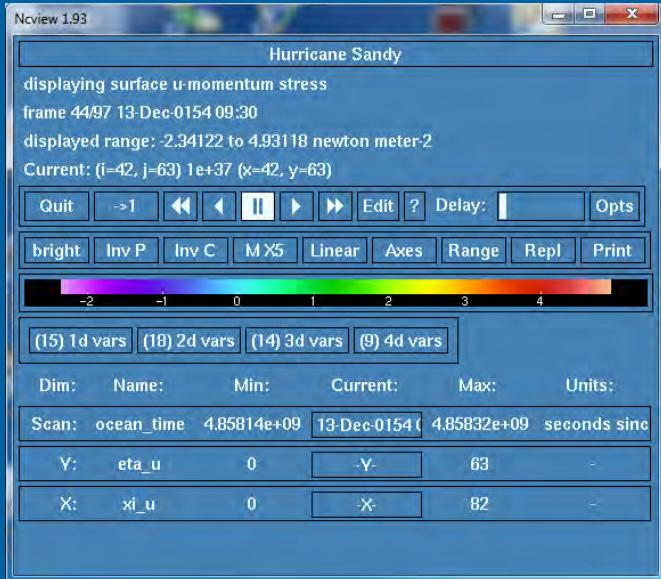
Make sure the –np X number of procs is equal to the NtileI * NtileJ in the ocean.in file.

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

10) Output

sustr

temp (surface)



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