ELCIRC: overview of formulation

Y. Joseph Zhang & António M. Baptista
Center for Coastal and Land-Margin Research,
OGI School of Science & Engineering,
Oregon Health & Science University

2nd ELCIRC User Group Meeting

Physical model

$$\nabla \cdot \mathbf{u} + \frac{\partial w}{\partial z} = 0, \ (\mathbf{u} = (u,v))$$

- Continuity equation

$$\frac{\partial \eta}{\partial t} + \nabla \cdot \int_{-h}^{\eta} \mathbf{u} dz = 0$$

- Momentum equations

$$\frac{D\mathbf{u}}{Dt} = \mathbf{f} - g\nabla \eta + \frac{\partial}{\partial z}\left(\nu \frac{\partial \mathbf{u}}{\partial z}\right); \ \ \mathbf{f} = -f\mathbf{k} \times \mathbf{u} + \alpha g \nabla \hat{\psi} - \frac{1}{\rho_0}\nabla p_A - \frac{g}{\rho_0}\int_z^{\eta} \nabla \rho d\zeta + \nabla \bullet (\mu \nabla \mathbf{u})$$

  - Vertical b.c.:

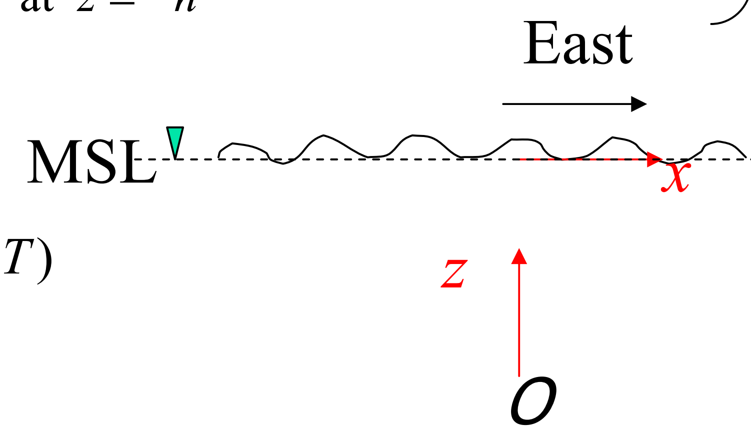
$$\begin{cases} \nu \dfrac{\partial \mathbf{u}}{\partial z} = \boldsymbol{\tau}_w \ \ \text{at} \ \ z = \eta \\ \nu \dfrac{\partial \mathbf{u}}{\partial z} = C_D \,|\mathbf{u}_b|\, \mathbf{u}_b, \ \ \text{at} \ \ z = -h \end{cases}$$

- Equation of state

$$\rho = \rho(p, S, T)$$

- Transport of salt and temperature

$$\frac{Dc}{Dt} = \frac{\partial}{\partial z}\left(\kappa \frac{\partial c}{\partial z}\right) + \frac{\dot{Q}}{\rho_0 C_p}, \qquad c = (S, T)$$



- Turbulence closure: Umlauf and Burchard 2003

$$\frac{Dk}{Dt} = \frac{\partial}{\partial z}\left(\nu_k^{\psi} \frac{\partial k}{\partial z}\right) + \underbrace{K_{mv} M^2}_{\text{shear}} + \underbrace{K_{hv} N^2}_{\text{stratification}} - \varepsilon$$

$$\frac{D\psi}{Dt} = \frac{\partial}{\partial z}\left(\nu_{\psi} \frac{\partial \psi}{\partial z}\right) + \frac{\psi}{k}\left(c_{\psi 1} K_{mv} M^2 + c_{\psi 3} K_{hv} N^2 - c_{\psi 2} F_{wall} \varepsilon\right) \qquad \psi = \left(c_{\mu}^0\right)^p k^m \ell^n,$$

Numerical Scheme: horizontal grid

Orthogonal unstructured

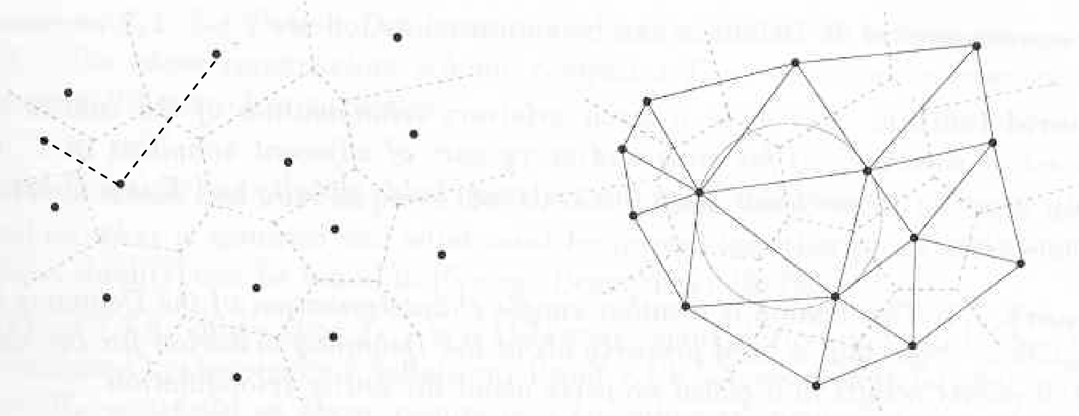
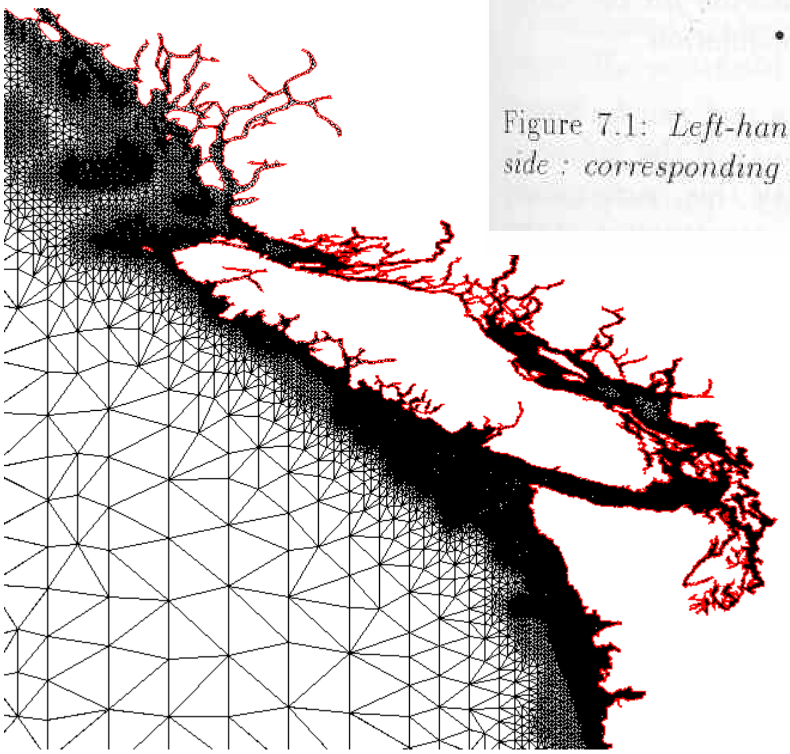
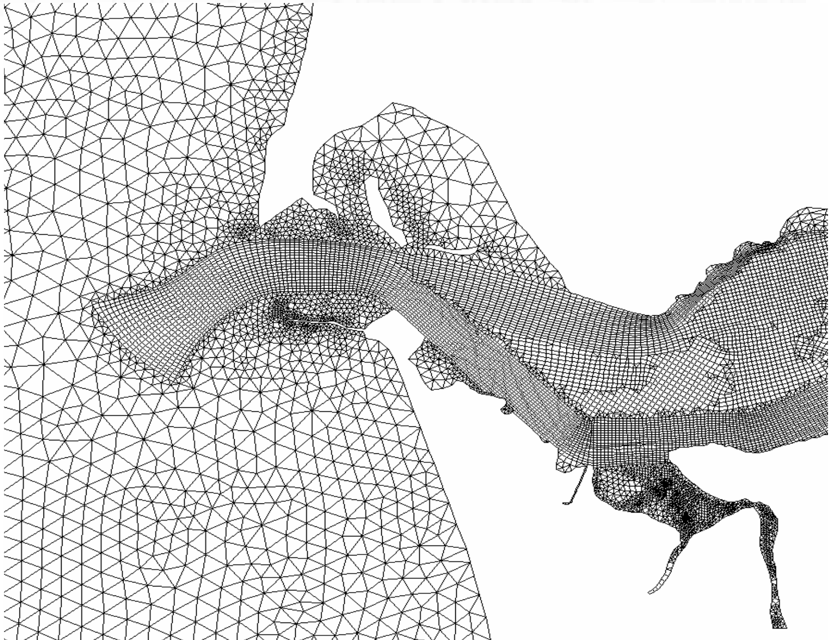
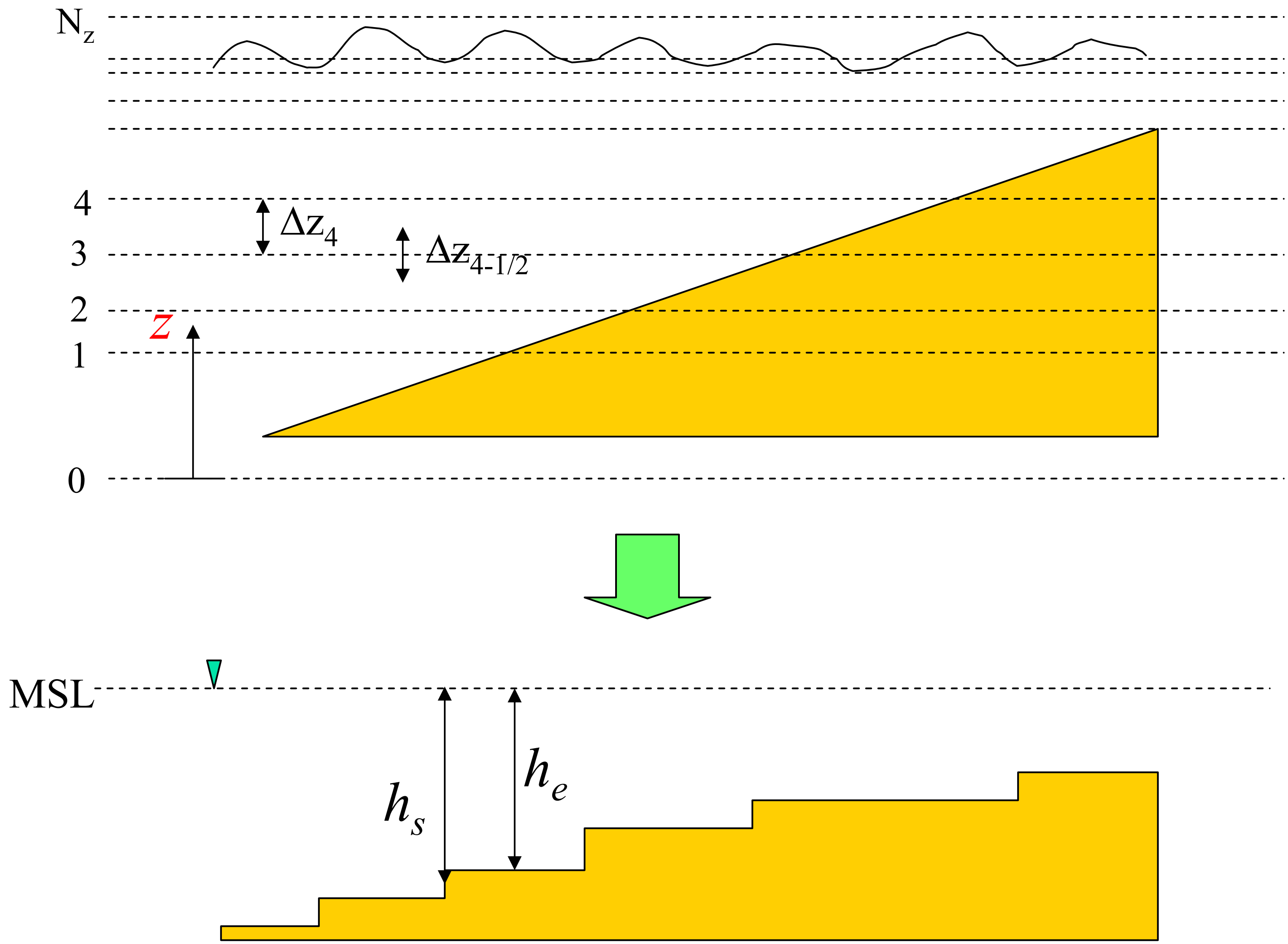


Figure 7.1: *Left-hand side : Voronoï diagram (in two dimensions). Right-hand side : corresponding Delaunay triangulation.*





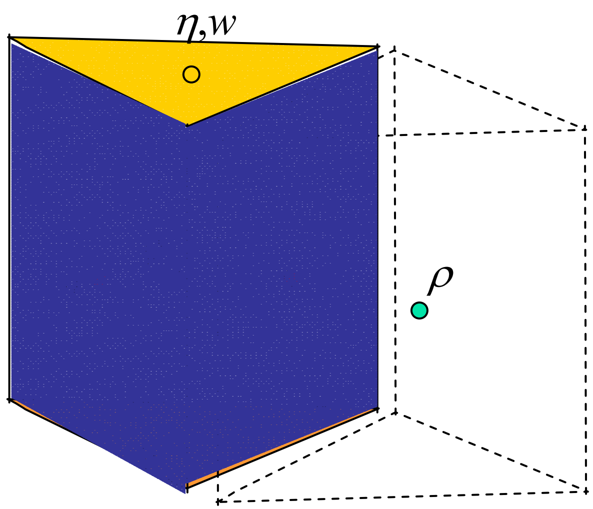
Numerical Scheme: vertical grid

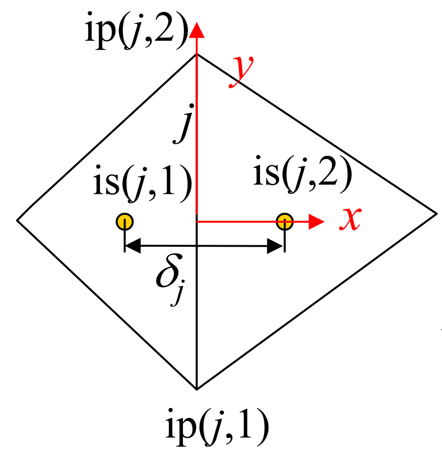


Numerical Scheme: notations

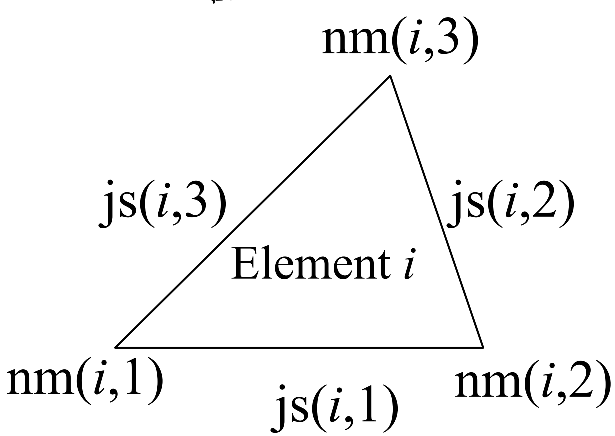
Primary unknowns: $\eta, u, v, w, \rho, S, T$
$K_{mv}, K_{hv}, \nu_{\psi}, k, \ell$

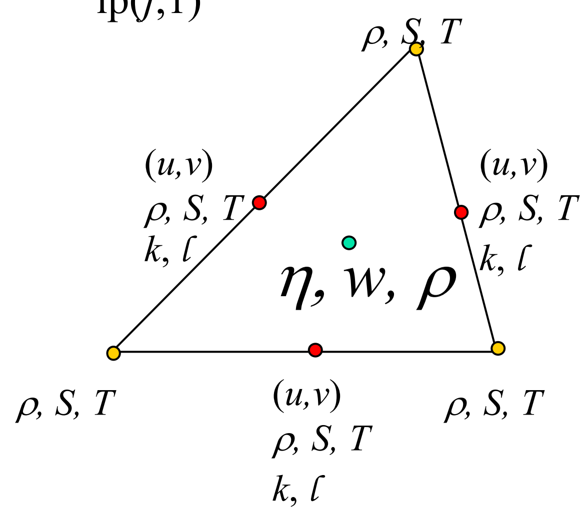
$u \equiv u_n, \; v \equiv u_t$





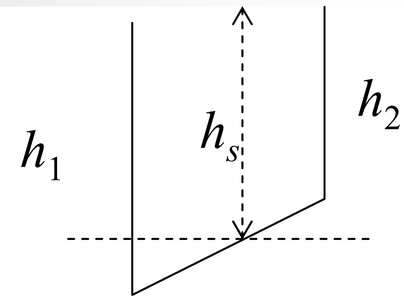
$$\left.\frac{\partial \eta}{\partial x}\right|_j = \frac{\eta_{is(j,2)} - \eta_{is(j,1)}}{\delta_j}$$



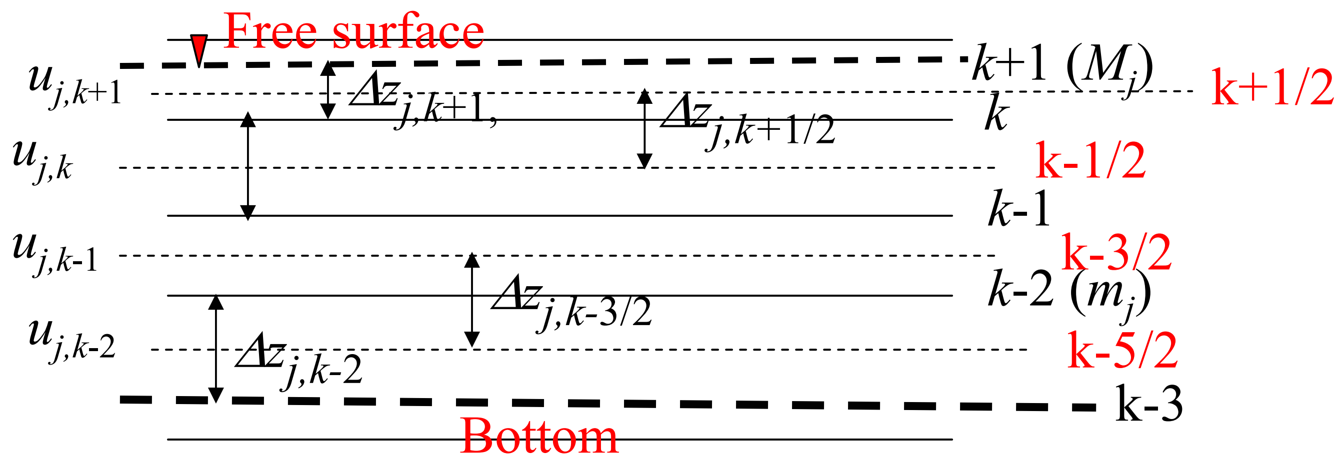


Vertical structure of variables

1. Depth at a side: $h_s=(h_1+h_2)/2$;
2. Depth at an element: $h_e=\max(h_{s1}, h_{s2}, h_{s3}, h_{s4})$
3. Inconsistencies of indices



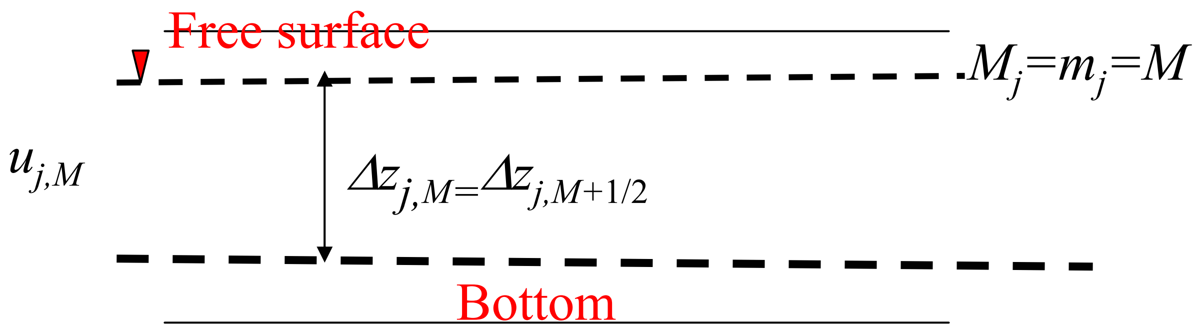
From one *side*'s (side *j*) perspective:



Define $\Delta z_{M_j+1/2} = \Delta z_{M_j}$, $\Delta z_{m_j-1/2} = \Delta z_{m_j}$

A special case

- When there is only one layer:

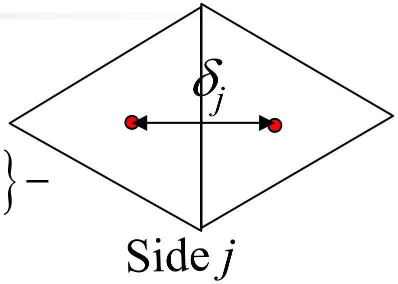


- Discretized 3D equations automatically become 2D depth averaged version

Not for SELFE!

Discretized momentum equation

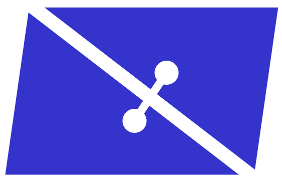
- Finite difference applied to face center (*j,k*) ($m_j \le k \le M_j$):



$$\frac{u^{n+1}_{j,k} - u^n_*}{\Delta t} = f_j v^n_{j,k} - \frac{g\omega_j}{\delta_j}\left\{\alpha\left[\eta^{n+1}_{is(j,2)} - \eta^{n+1}_{is(j,1)}\right] + (1-\alpha)\left[\eta^n_{is(j,2)} - \eta^n_{is(j,1)}\right]\right\} -$$

$$\frac{g}{\rho_0 \delta_j}\left\{\sum_{l=k}^{M_j} \Delta z^n_{j,l}\left[\rho^n_{is(j,2),l} - \rho^n_{is(j,1),l}\right] - \Delta z^n_{j,k}\left[\rho^n_{is(j,2),k} - \rho^n_{is(j,1),k}\right]/2\right\} +$$

$$\frac{1}{\Delta z^n_{j,k}}\left[E^v_{j,k}\frac{u^{n+1}_{j,k+1} - u^{n+1}_{j,k}}{\Delta z^n_{j,k+1/2}} - E^v_{j,k-1}\frac{u^{n+1}_{j,k} - u^{n+1}_{j,k-1}}{\Delta z^n_{j,k-1/2}}\right]$$

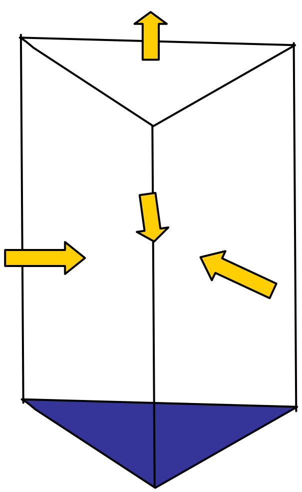


- Vertical boundary conditions:

$$E^v_{j,m_j-1}\frac{u^{n+1}_{j,m_j} - u^{n+1}_{j,m_j-1}}{\Delta z^n_{j,m_j-1/2}} = \tau_b u^{n+1}_{j,m_j},$$

$$E^v_{j,M_j}\frac{u^{n+1}_{j,M_j+1} - u^{n+1}_{j,M_j}}{\Delta z^n_{j,M_j+1/2}} = \tau^x_{wind} / \rho_0.$$

- Similar for *v*
- FV approximation for the continuity eq.at element centers:



$$\int_{\Omega_i}\left(\frac{\partial \eta}{\partial t} + \nabla\cdot\int_{-h}^{\eta}\mathbf{u}dz\right)d\Omega_i = 0 \;\Rightarrow\; \int_{\Omega_i}\frac{\partial \eta}{\partial t}d\Omega_i + \int_{\Gamma_i} d\Gamma_i \int_{-h}^{\eta} u_n dz = 0$$

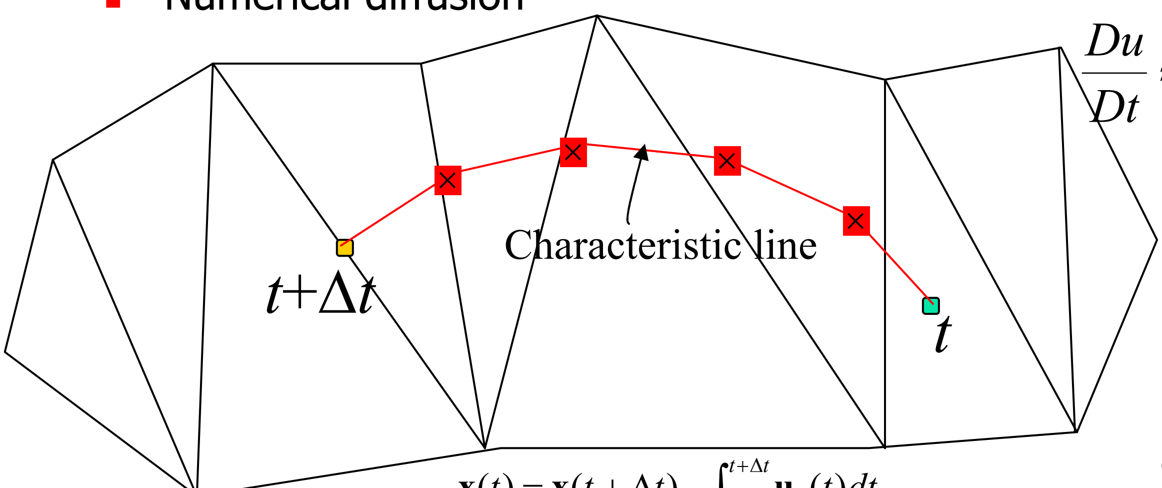
$$P_i\frac{\eta^{n+1}_i - \eta^n_i}{\Delta t} + \alpha\sum_{l=1}^{3} s_{i,l}\lambda_{jsl}\sum_{k=m_{jsl}}^{m_{jsl}}\Delta z^n_{jsl,k}u^{n+1}_{jsl,k} + (1-\alpha)\sum_{l=1}^{3} s_{i,l}\lambda_{jsl}\sum_{k=m_{jsl}}^{m_{jsl}}\Delta z^n_{jsl,k}u^n_{jsl,k} = 0$$

$$jsl = js(i,l),\; s_{i,l} = \frac{is(jsl,1) + is(jsl,2) - 2i}{is(jsl,2) - is(jsl,1)},\; \lambda_{jsl} = \text{length of side } jsl$$

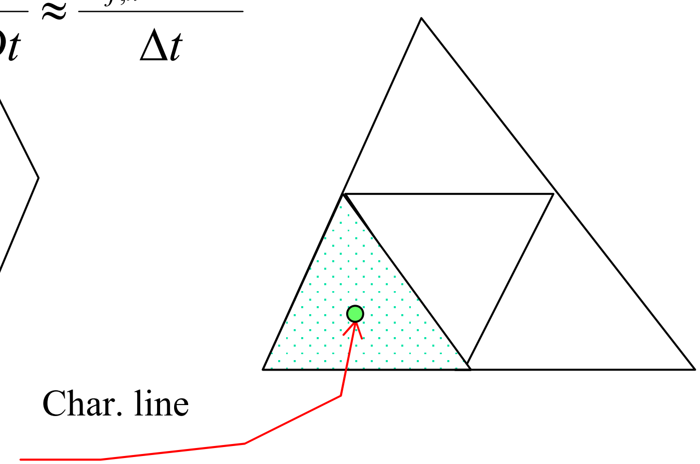
Backtracking (Eulerian-Lagrangian method)

- ELM: takes advantage of both Lagrangian and Eulerian methods
  - Grid is fixed in time, and time step is not limited by CFL condition
  - Advections are evaluated by following a particle that starts at certain point at time t and ends right at a pre-given point at time $t+\Delta t$.
  - The process of finding the starting point of the path (foot of characteristic line) is called backtracking, which is done by integrating $d\mathbf{x}/dt=\mathbf{u}_3$ backward in time.
  - To better capture the particle movement, the backward integration is often carried out in small sub-time steps ($\Delta t/N$).
    - Simple backward Euler method as the standard option
    - 5th-order embedded R-K method as an alternative
    - Interpolation-ELM does not preserve mass
  - Numerical diffusion

$$\frac{Du}{Dt} \approx \frac{u_{j,k}^{n+1} - u_*^n}{\Delta t}$$



$$\mathbf{x}(t) = \mathbf{x}(t+\Delta t) - \int_t^{t+\Delta t} \mathbf{u}_3(t)\,dt$$



Formal substitution

- Momentum and wave-continuity equations in matrix form:

$$\mathbf{A}_j\mathbf{U}_j^{n+1} = \mathbf{G}_j^n - \alpha g \frac{\omega_j \Delta t}{\delta_j}\left[\eta_{is(j,2)}^{n+1} - \eta_{is(j,1)}^{n+1}\right]\mathbf{\Delta Z}_j^n,\ \ j = 1,\cdots,N_s,$$

$$\mathbf{A}_j\mathbf{V}_j^{n+1} = \mathbf{F}_j^n - \alpha g \frac{\omega'_j \Delta t}{\lambda_j}\left[\eta_{ip(j,2)}^{n+1} - \eta_{ip(j,1)}^{n+1}\right]\mathbf{\Delta Z}_j^n,\ \ j = 1,\cdots,N_s,$$

$$\eta_i^{n+1} = \eta_i^n - \frac{\alpha \Delta t}{P_i}\sum_{l=1}^{3} s_{i,l}\lambda_{jsl}\left[\mathbf{\Delta Z}_{jsl}^n\right]^T \mathbf{U}_{jsl}^{n+1} - \frac{(1-\alpha)\Delta t}{P_i}\sum_{l=1}^{3} s_{i,l}\lambda_{jsl}\left[\mathbf{\Delta Z}_{jsl}^n\right]^T \mathbf{U}_{jsl}^{n},\quad i = 1,\cdots,N_e$$

$$\mathbf{U}_j^{n+1} = \begin{bmatrix} u_{j,M_j}^{n+1} \\ \vdots \\ u_{j,m_j}^{n+1} \end{bmatrix},\ \mathbf{\Delta Z}_j^n = \begin{bmatrix} \Delta z_{j,M_j}^{n+1} \\ \vdots \\ \Delta z_{j,m_j}^{n+1} \end{bmatrix}\qquad \boxed{\mathbf{A}_j, \mathbf{G}_j}$$

- Substitution of the first to third equation leads to:

$$\mathbf{U}_j^{n+1} = \mathbf{A}_j^{-1}\mathbf{G}_j^n - \alpha g \frac{\omega_j \Delta t}{\delta_j}\left[\eta_{is(j,2)}^{n+1} - \eta_{is(j,1)}^{n+1}\right]\mathbf{A}_j^{-1}\mathbf{\Delta Z}_j^n,\ (j = 1,\cdots,N_s)$$

$$\eta_i^{n+1} - \frac{g\alpha^2\Delta t^2}{P_i}\sum_{l=1}^{3}\frac{s_{i,l}\lambda_{jsl}}{\delta_{jsl}}\left[\eta_{is(jsl,2)}^{n+1} - \eta_{is(jsl,1)}^{n+1}\right]\left[\mathbf{\Delta Z}_{jsl}^n\right]^T \mathbf{A}_{jsl}^{-1}\mathbf{\Delta Z}_{jsl}^n =$$

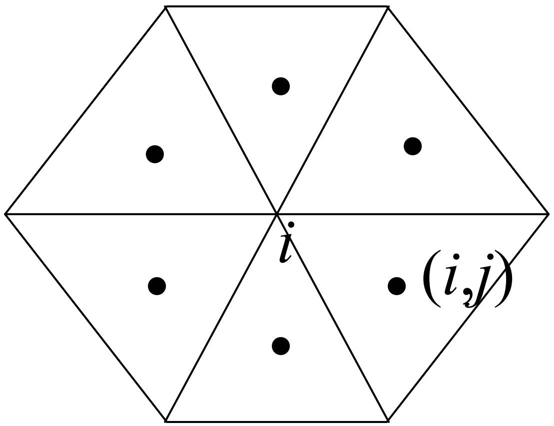
$$\eta_i^n - \frac{(1-\alpha)\Delta t}{P_i}\sum_{l=1}^{3} s_{i,l}\lambda_{jsl}\left[\mathbf{\Delta Z}_{jsl}^n\right]^T \mathbf{U}_{jsl}^n - \frac{\alpha \Delta t}{P_i}\sum_{l=1}^{3} s_{i,l}\lambda_{jsl}\left[\mathbf{\Delta Z}_{jsl}^n\right]^T \mathbf{A}_{jsl}^{-1}\mathbf{G}_{jsl}^n,\ \ (i = 1,\cdots,N_e)$$

Tangential velocity

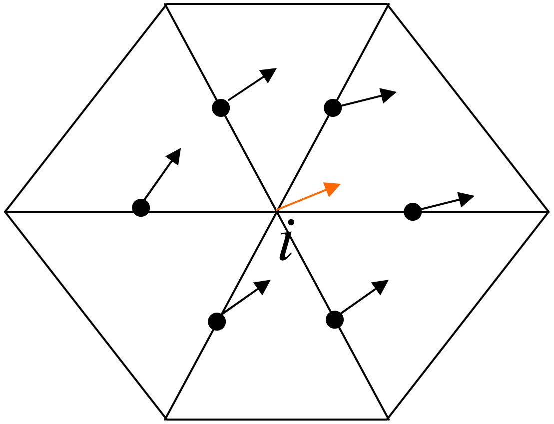
- Elevations at nodes:

$$\mathbf{A}_j \mathbf{V}_j^{n+1} = \mathbf{F}_j^n - \alpha g \frac{\omega'_j \Delta t}{\lambda_j} \left[\eta_{ip(j,2)}^{n+1} - \eta_{ip(j,1)}^{n+1} \right] \mathbf{\Delta Z}_j^n,$$

$$\eta_i^{n+1} - \eta_i^n = \frac{\sum_j P_{(i,j)} \left[\eta_{(i,j)}^{n+1} - \eta_{(i,j)}^n \right]}{\sum_j P_{(i,j)}}$$



- Averaging around the ball

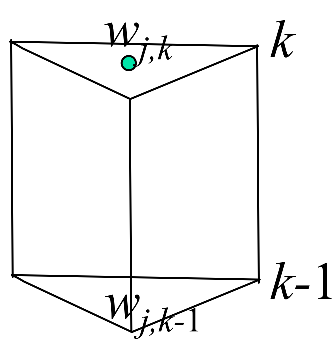


Vertical velocity

- Serves primarily as a diagnostic variable for mass conservation
- Generally small, but if not treated with care, it can lead to excessive vertical mixing for *S,T*.
- Finite Volume Method for continuity equation:

$$w_{i,k}^{n+1} = w_{i,k-1}^{n} - \frac{1}{P_i} \sum_{j=1}^{3} s_{i,j} \lambda_{jsj} \Delta z_{jsj,k} u_{jsj,k}^{n+1} \quad (k = m,\ldots,M;\ jsj = js(i,j));$$

$$w_{i,m-1}^{n} = 0 \quad \text{(b.c.)}$$

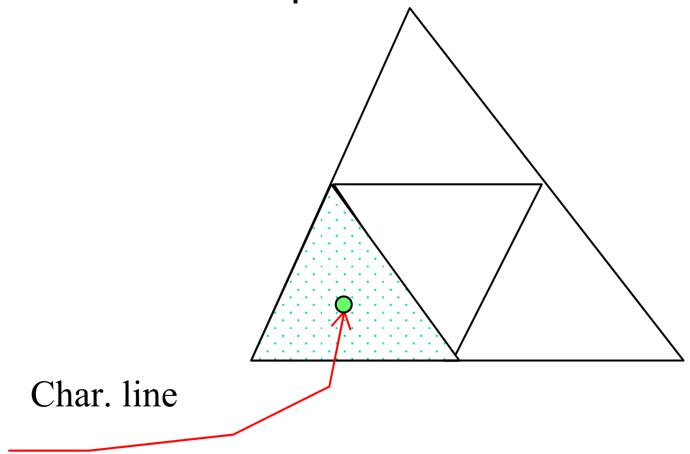


Transport equation

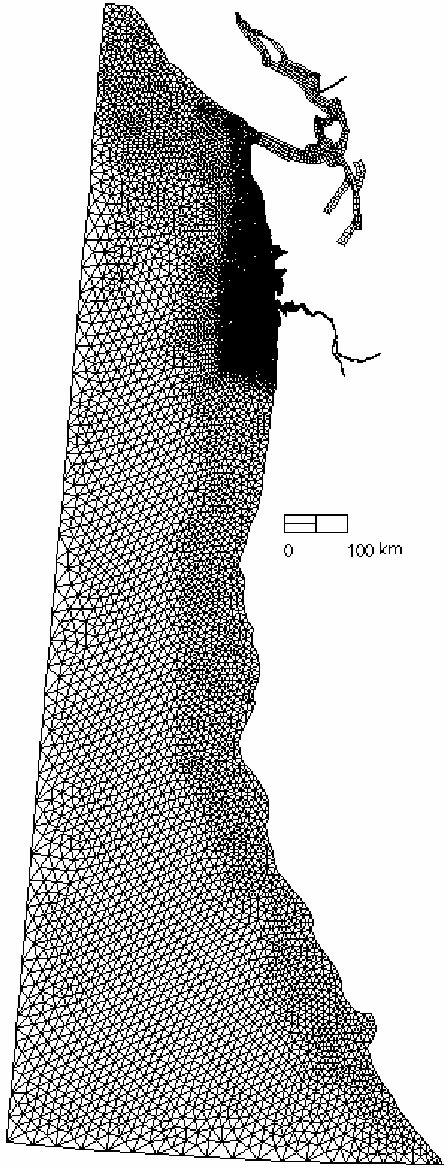
- Finite difference method

$$\frac{c_{j,k}^{n+1}-c_*^n}{\Delta t}=\frac{1}{\Delta z_{j,k}^n}\left[e_{j,k}^v\frac{c_{j,k+1}^{n+1}-c_{j,k}^{n+1}}{\Delta z_{j,k+1/2}^n}-e_{j,k-1}^v\frac{c_{j,k}^{n+1}-c_{j,k-1}^{n+1}}{\Delta z_{j,k-1/2}^n}\right]+Q_{j,k}^{n+1}\quad (j=1,\ldots,N_s \text{ or } N_p)$$

- Numerical diffusion (subdivison of elements)
- Open boundary condition (o.b.c):
  - *S,T* are allowed to leave the domain unhindered for outflow condition, and are specified for inflow.
  - With backtracking, this can be easily done
- Heat budget:
  - At the air water interface, total heat flux is the sum of upward radiation flux, heat loss due to latent heat of evaporation, and upward turbulent heat flux
  - In addition, solar radiation serves as a heat (body) source for temperature



Computational performance

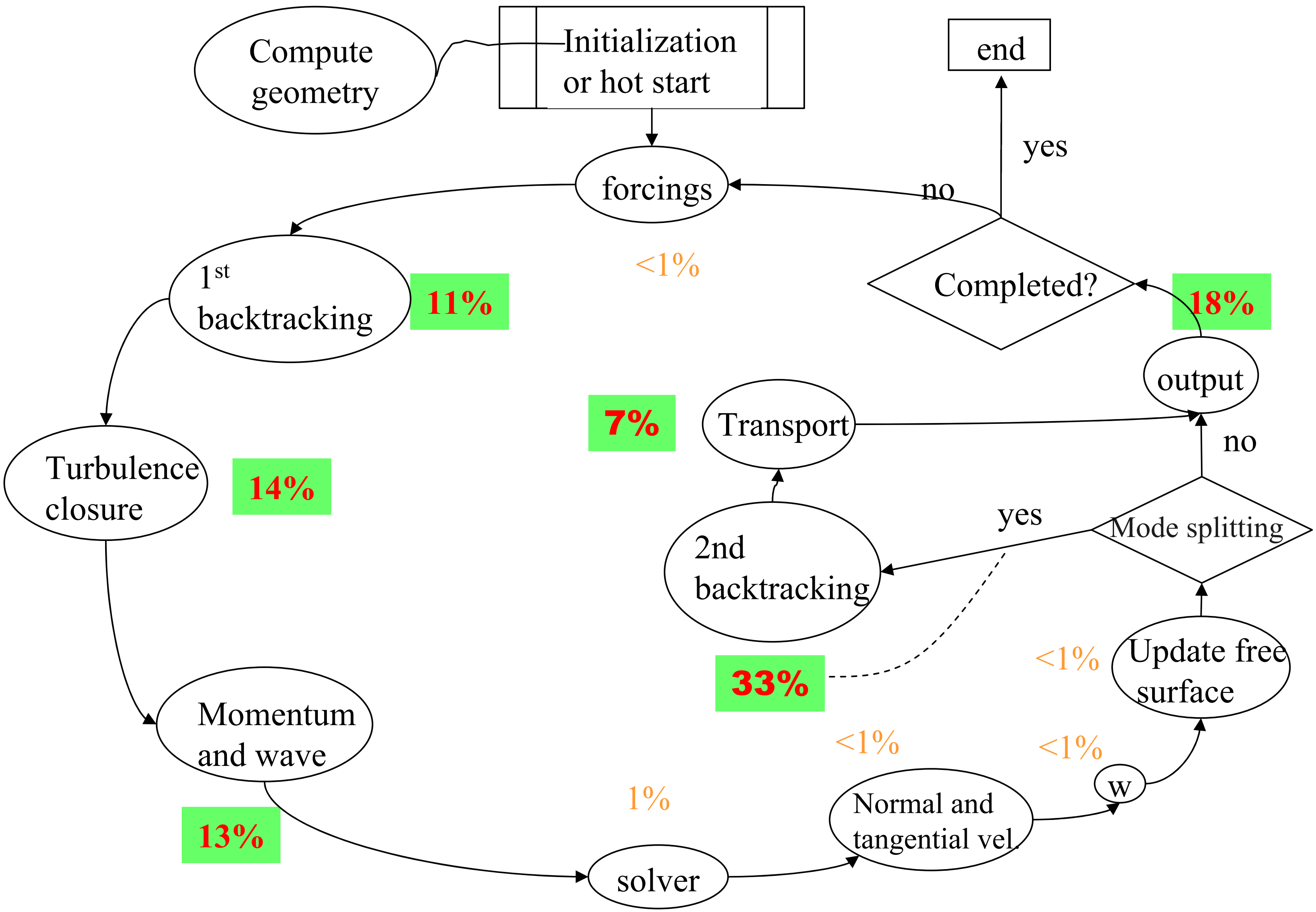


Forecast grid specifications:

- **62 *z*-levels**
- **50,622 horizontal elements**
- **~2.3m prism faces**
- **2.3x faster than real time on a single CPU Intel Xeon**
- **~5GB hvel.64 per week**

Serial ELCIRC flow chart

All numbers are based on a most recent CORIE run

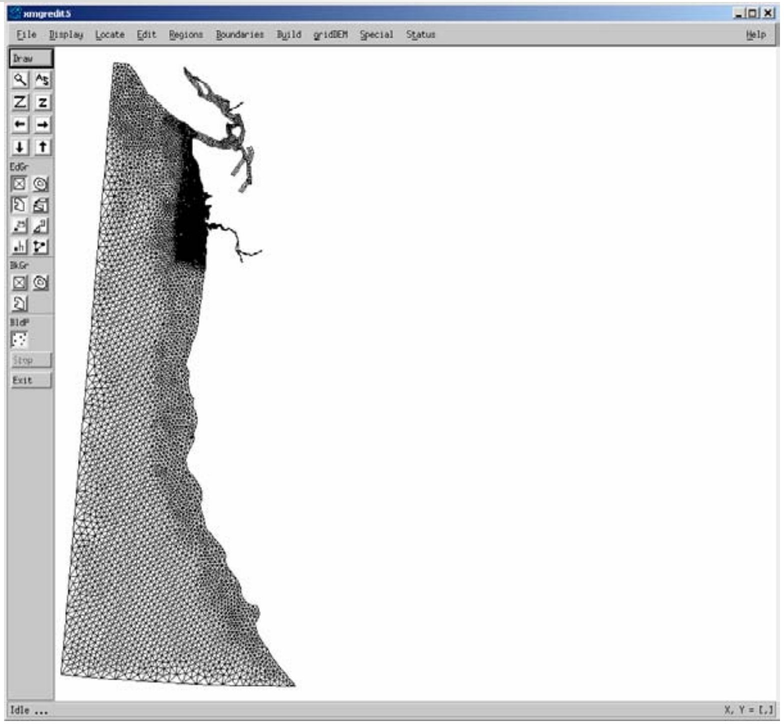


Preparing an Elcirc run

http://www.ccalmr.ogi.edu/CORIE/modeling/elcirc/

- Inputs:
  - *.gr3 : grid files;
  - *.bp: build point files;
  - *.th: time history files;
  - *.ic: initial conditions;
  - *.in: vgrid; parameters etc.
- Create a horizontal grid and open and land boundaries with xmgredit5
- Create vertical grid file (vgrid.in)
- Create param.in (parameter option file)
- Run pre-processor (ipre=1) to get obe.out (needed in param.in), centers.bp, and sidecenters.bp;
- Get all external forcings
  - Tides
  - Wind & heat exchange
  - Time history input at boundaries (river discharge etc.)
- Create additional input files if necessary
  - Initial condition input: salt.ic & temp.ic;
  - Bottom friction input: drag.bp
- Reset pre-processor flag to 0 and run ELCIRC
- Analyze the results (xmvis6)

Grid generation: XMGREDIT5 (Turner et al.)



- Most useful functions:
  - Build
    - Circular/rectangular spread
    - Automatic placement
    - Triangulate build points
  - Boundaries
    - Compute boundary
  - gridDEM
    - Load bathymetry
    - Create open/land boundaries
  - Edit
    - Edit over grid/regions→evaluate; conversion between triangles and quads
    - Edit triangles→ move nodes, delete elements …
  - Display
    - Isolines of bathymetry (edit/background grid)
- Example of horizontal grid file

Sample *.gr3

grid05142004; min depth=-10m

50622 34190

1 346712.890917 286491.506150   9.185

2 346709.710000 286589.787120   8.358

3 346661.996250 286494.484361   9.374

4 346172.135083 286702.959145   9.319

7   ……..

1 3 1 2 3

2 3 4 5 6

3 4 7 8 9 130

4 3 10 11 12

…..

4 = Number of open boundaries

94 = Total number of open boundary nodes

85 = Number of nodes for open boundary 1

23878

23867

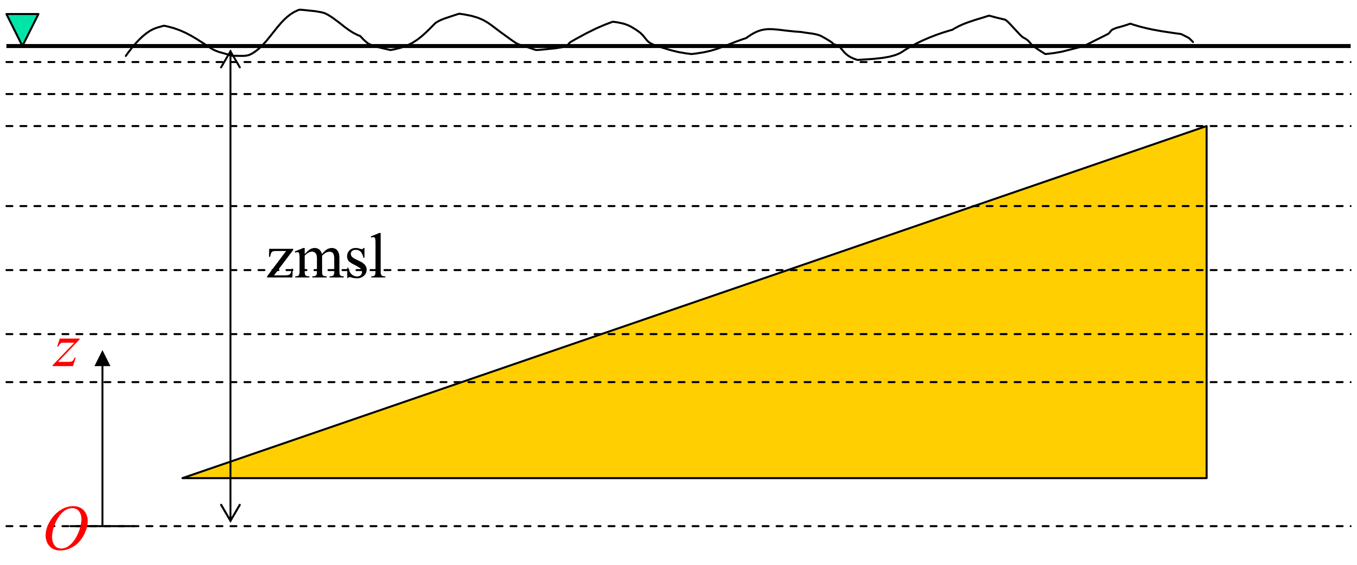
23868

…..

Sample vgrid.in

62   4825.1 ← **zmsl**

1	2627.00	2627.00
2	1000.00	3627.00
3	500.00	4127.00
4	200.00	4327.00
……..		
59	0.80	4826.60
60	1.00	4827.60
61	2.00	4829.60
62	36.40	4866.00

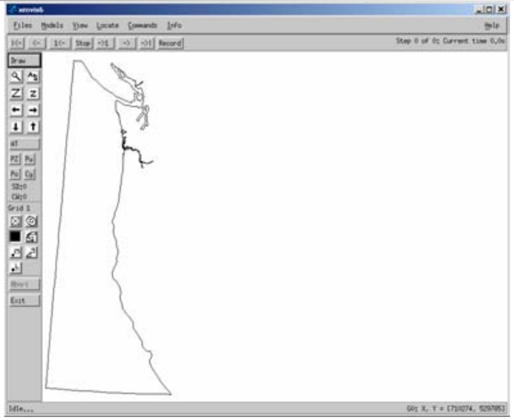


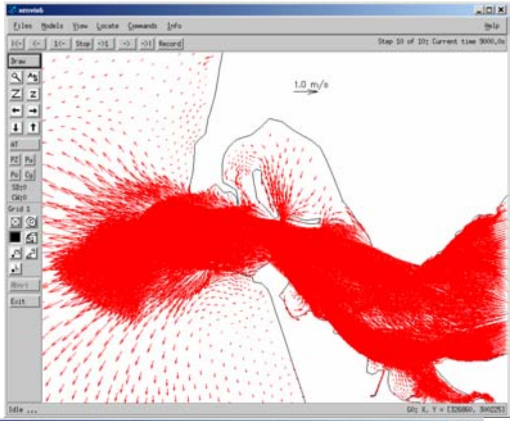
Param.in

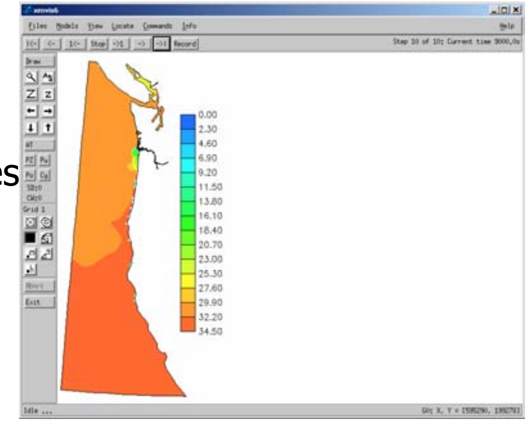
QUARTER ANNULAR TEST EXAMPLE 1
ELCIRC
1 NSCREEN
0 iforecast
0 IHOT
1 ICS
0.0 0.0 SLAM0,SFEA0
1.0
1 0   baroclinic/barotropic
4. 30. 0. 33.
5.   RNDAY
1 2.
2095.872   1047.936 Dt
2 nsubfl
5 90 NDELT
1 nadv
0.01 h0
0 ntau
0. Cd
0   NCOR
0.0 CORI
2 3600. NWS ← hdf
1 0.5
1 0 heat
0   turbulence closure
1.e-2 1.e-4
0 ihorcon
0.
0
0. 0.
1 1 i.c.
1   ! NBFR
M2
0.000140525700000 1.0 0.0
8
M2
0.3048 0.00
.....   ← b.c.
10 960
1   elevation: iof,touts,toutf,spool
……
1 NHSTAR
1 1000 0 5.e-6 1.e-13
0 0 iflux ihcheck
1 iwmode
1 nsplit

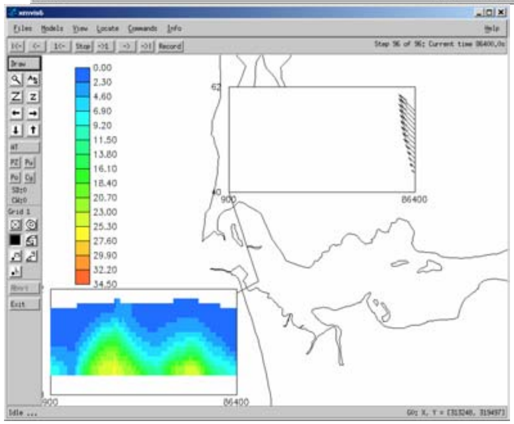
Visualization: XMvis6 (Turner et al.)

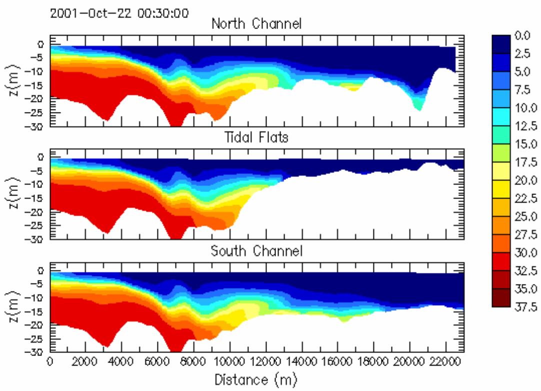
- Main global binary outputs:
  - *.61: 2D scalars (1_elev.61)
  - *.62: 2D vectors (3_wind.62)
  - *.63: 3D scalars (2_salt.63)
  - *.64: 3D vectors (7_hvel.64)
- Most useful functions:
  - Files
    - ELCIRC slabs ← horizontal levels
    - ELCIRC samples ← vertical profiles
    - ELCIRC surface/bottom
    - ELCIRC transects
  - Models
    - Time histories
  - Locate
- G3

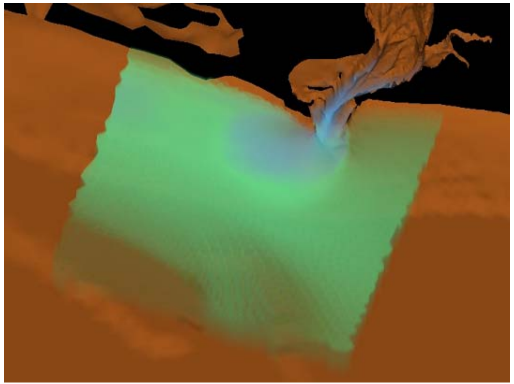












Practical issues

- Horizontal grid
  - Orthogonal vs. non-orthogonal elements
  - Triangles vs quads
  - Use uniform quads and general triangles
- Vertical grid
  - Adequate resolution for baroclinic applications
- Parameters

$$\Delta t = \Delta x / \sqrt{g' h}$$

  - Baroclinic time step: optimal near
  - Implicitness factor: 0.6
  - Turbulence closure (GLS): may need to impose mixing limits for different regions
  - Bottom friction has limited influence
- Nudging for S,T: implemented in version 02k
  - Found to accelerate the time to reach "equilibrium"
  - Parallel runs for long-term simulation (b.c.)

ELCIRC: the good, the bad, and the ugly

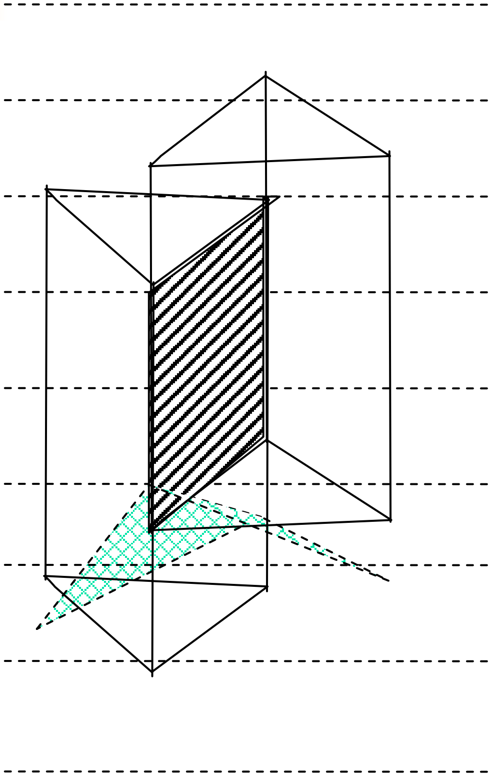
- Summary of main features
  - Semi-implicit finite-difference/finite-volume method
    - Unstructured grid in horizontal; *z*-coordinates in the vertical
    - Semi-implicit in time
      - Stability is guaranteed for $0.5<\theta<1$.
    - Finite difference for momentum, transport and turbulence closure equations
    - Finite volume for continuity eq
      - Volume conservation is strictly enforced locally and globally
    - No splitting between the external and internal modes
  - Treatment of advection: Eulerian-Lagrangian (ELM)
    - CFL restriction from baroclinicity only→ large time steps→efficiency
  - Wetting and drying is treated naturally by the FV formulation
- The good
  - Robust, efficient, flexible, volume conservative
- The bad
  - Low order; Staircase bottom; orthogonality; numerical diffusion/dissipation
- The ugly
  - Inconsistencies and ambiguities of indices

MPI ELCIRC

- Developed by Tim Campbell at NRL
- Use ParMeTiS and MeTiS graph partitioning libraries for domain-decomposition
  - Load-balancing is done for 3D domain but the continuity equation is solved along horizontal direction
- Parallel Jacobi preconditioned Conjugate Gradient solver
- Most changes in the backtracking part
- Implemented on various platforms: Intel; IBM; NOAA Beowulf cluster
  - Efficiency on our Intel cluster is about 50%; higher on Tim's 64-bit IBM cluster
- Work in progress
  - Load balancing
  - I/O
  - MPI SELFE

END

Side and element indices

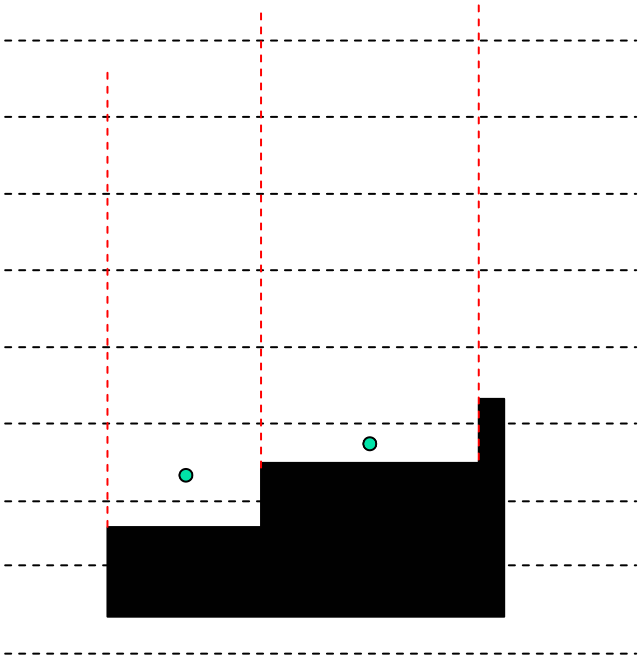


For a wet side j

$$1 \le m'_{is(j,l)} \le m_j \le M_j \le M'_{is(j,l)} \quad (l = 1, 2)$$

For a dry side j

$$M_j = 0, \quad M_{is(j,l)} ? = 0$$



Baroclinic gradient

Matrices **A** and **G**

$$\mathbf{A}_j = \begin{pmatrix} \Delta z_{j,M_j} + a_{j,M_j-1/2} & -a_{j,M_j-1/2} & \cdots & 0 \\ -a_{j,M_j-1/2} & \Delta z_{j,M_j-1} + a_{j,M_j-1/2} + a_{j,M_j-3/2} & -a_{j,M_j-3/2} \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & -a_{j,m_j+1/2} & \Delta z_{j,m_j-1} + a_{j,m_j+1/2} + \tau_b \Delta t \end{pmatrix}$$

$$a_{j,k\pm1/2} = E^{v}_{k\pm1/2-1/2} \frac{\Delta t}{\Delta z^{n}_{j,k\pm1/2}}$$

$$\mathbf{G}_j = \begin{pmatrix} g^{n}_{j,M_j} + \tau^{x}_{wind} \Delta t / \rho_0 \\ g^{n}_{j,M_j-1} \\ \vdots \\ g^{n}_{j,m_j} \end{pmatrix} \qquad g^{n}_{j,k} = \Delta z^{n}_{j,k} \left\{ f_j v^{n}_{j,k} \Delta t + u^{*}_{j,k} - \frac{g \omega_j \Delta t}{\delta_j} (1-\alpha) \left[\eta^{n}_{is(j,2)} - \eta^{n}_{is(j,1)} \right] - \ldots \right\}$$

SELFE: model formulation

Semi-implicit Eulerian-Lagrangian Finite Element (SELFE)

- A formal semi-implicit finite-element framework
- Unstructured grid in the horizontal
- Hybrid *SZ* in the vertical
- Eulerian-Lagrangian method (ELM) for advection
- Treats inundation/drying in a natural way
- A key step is to decouple the continuity and momentum equations through the bottom boundary layer
- Volume conservation is not enforced numerically
- Numerical efficiency: semi-implicit time stepping; ELM
  - Mild stability constraint: Coriolis, horizontal viscosity, baroclinic pressure
  - Moderately more expensive than ELCIRC
- Convergence rate: 2nd order for uniform grid; 1st order otherwise

Governing equations

- Continuity equation

$$\nabla\cdot\mathbf{u}+\frac{\partial w}{\partial z}=0,\ \ (\mathbf{u}=(u,v))$$

$$\frac{\partial\eta}{\partial t}+\nabla\cdot\int_{-h}^{\eta}\mathbf{u}dz=0$$

- Momentum equations

$$\frac{D\mathbf{u}}{Dt}=\mathbf{f}-g\nabla\eta+\frac{\partial}{\partial z}\left(\nu\frac{\partial\mathbf{u}}{\partial z}\right);\ \ \mathbf{f}=-f\mathbf{k}\times\mathbf{u}+\alpha g\nabla\hat{\psi}-\frac{1}{\rho_0}\nabla p_A-\frac{g}{\rho_0}\int_z^{\eta}\nabla\rho d\zeta+\nabla\bullet(\mu\nabla\mathbf{u})$$

  - Vertical b.c.:

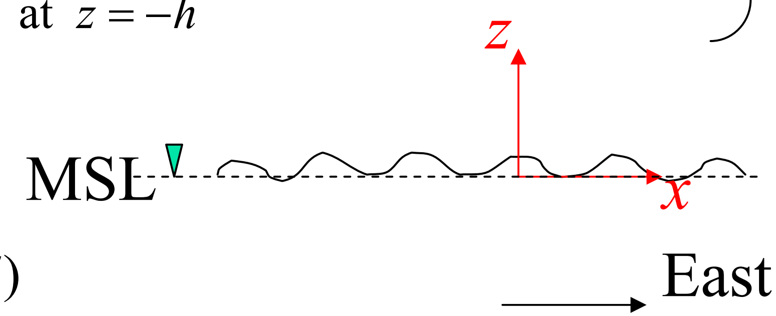
$$\begin{cases}\nu\dfrac{\partial\mathbf{u}}{\partial z}=\boldsymbol{\tau}_w\ \ \text{at}\ \ z=\eta\\ \nu\dfrac{\partial\mathbf{u}}{\partial z}=C_D\,|\,\mathbf{u}_b\,|\,\mathbf{u}_b,\ \ \text{at}\ \ z=-h\end{cases}$$

- Equation of state

$$\rho=\rho(p,S,T)$$

- Transport of salt and temperature

$$\frac{Dc}{Dt}=\frac{\partial}{\partial z}\left(\kappa\frac{\partial c}{\partial z}\right)+\frac{\dot{Q}}{\rho_0 C_p},\qquad c=(S,T)$$

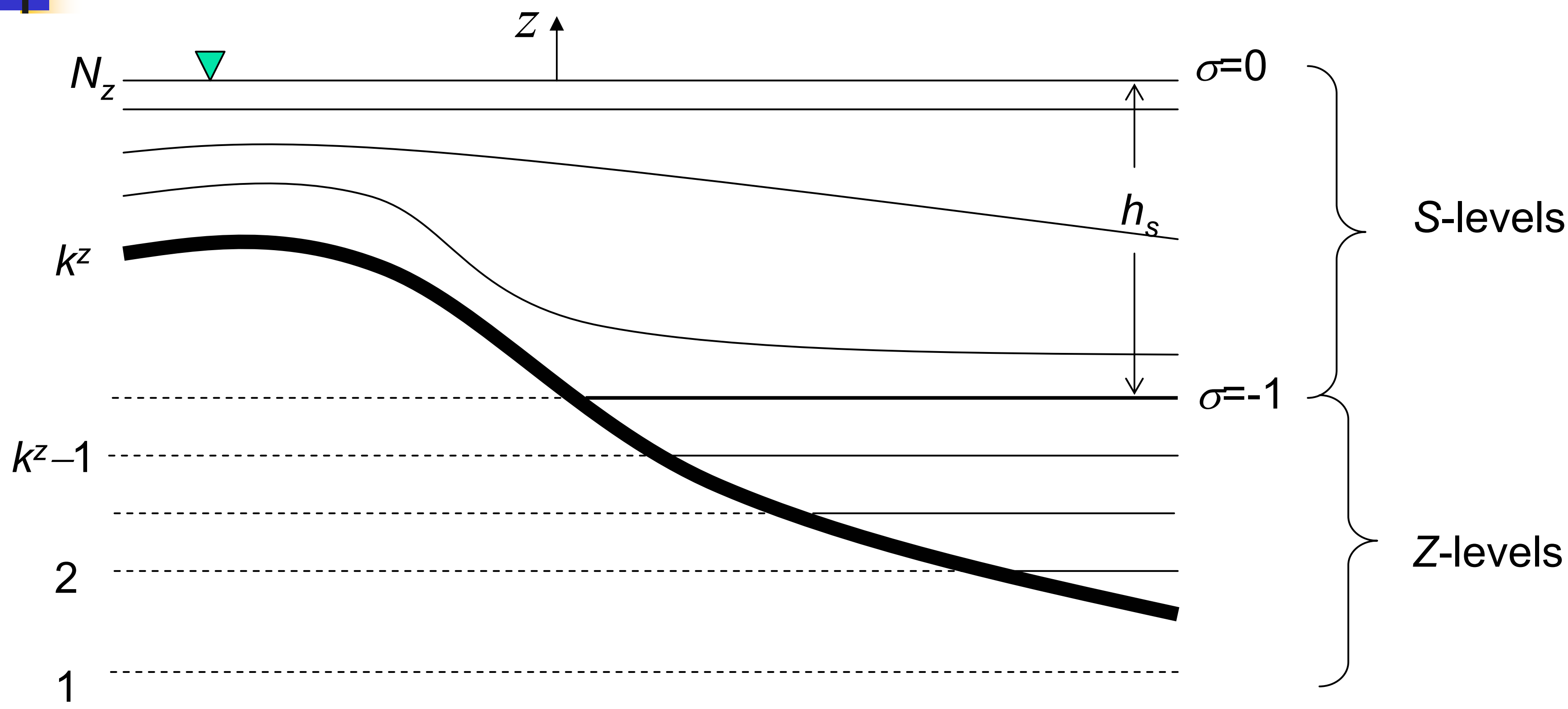


- Turbulence closure: Umlauf and Burchard 2003

$$\frac{Dk}{Dt}=\frac{\partial}{\partial z}\left(\nu_k^{\psi}\frac{\partial k}{\partial z}\right)+\underbrace{K_{mv}M^2}_{\text{shear}}+\underbrace{K_{hv}N^2}_{\text{stratification}}-\varepsilon$$

$$\frac{D\psi}{Dt}=\frac{\partial}{\partial z}\left(\nu_{\psi}\frac{\partial\psi}{\partial z}\right)+\frac{\psi}{k}\left(c_{\psi 1}K_{mv}M^2+c_{\psi 3}K_{hv}N^2-c_{\psi 2}F_{wall}\varepsilon\right)\qquad \psi=\left(c_{\mu}^0\right)^p k^m\ell^n,$$

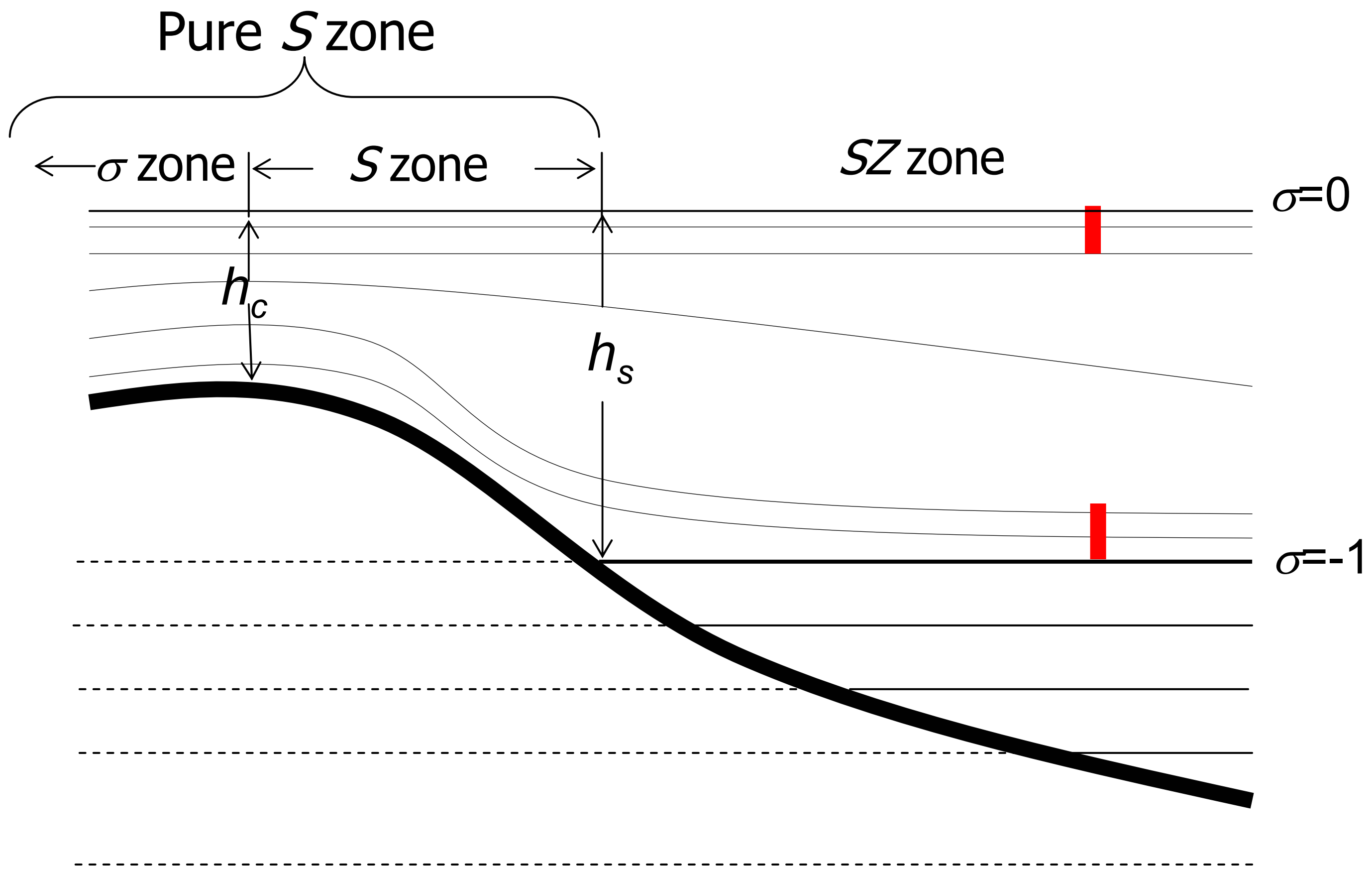
Vertical grid (1)



$$\begin{cases} z = \eta(1+\sigma) + h_c\sigma + (\tilde{h} - h_c)C(\sigma) & (-1 \le \sigma \le 0) \\ C(\sigma) = (1-\theta_b)\dfrac{\sinh(\theta_f\sigma)}{\sinh\theta_f} + \theta_b\dfrac{\tanh\left[\theta_f(\sigma+1/2)\right] - \tanh(\theta_f/2)}{2\tanh(\theta_f/2)} & (0 \le \theta_b \le 1;\ \ 0 < \theta_f \le 20) \end{cases}$$

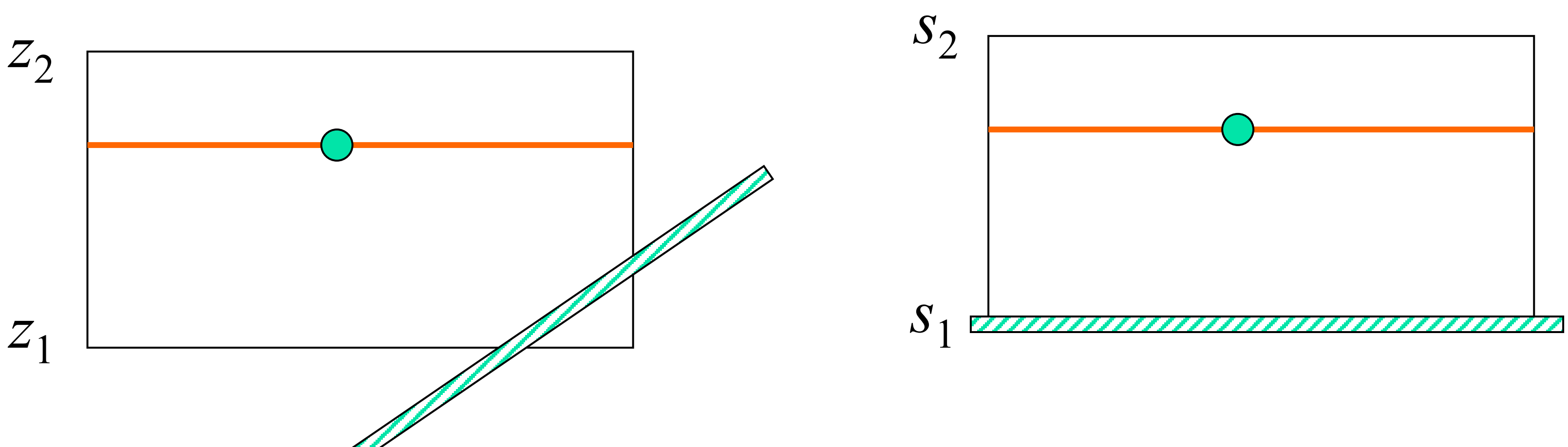
$$\tilde{h} = \min(h, h_s)$$

Vertical grid (2)



Vertical grid (3)

- *S*-coordinates (Song and Haidvogel 1994) are ideal for shallow region
- *Z*-coordinates are necessary to "stabilize" the deep region
- σ-coordinates are used where *S*-coordinates are invalid
- Equations are not transformed into *S*- or σ-coordinates, but solved in their original form
- Interpolation mode: along *Z* or *S* (in pure *S* region)
- Hydrostatic consistency: $\int_z^\eta \nabla \rho d\zeta$
    - pressure Jacobian with higher-order integration
    - *Z*-method



Finite-element formulation

- A Galerkin weighted residual statement for the continuity equation:

$$\int_\Omega \phi_i \frac{\eta^{n+1}-\eta^n}{\Delta t} d\Omega + \theta\left[-\int_\Omega \nabla\phi_i \bullet \mathbf{U}^{n+1} d\Omega + \int_{\Gamma_v} \phi_i \hat{U}_n^{n+1} d\Gamma_v + \int_{\bar{\Gamma}_v} \phi_i U_n^{n+1} d\bar{\Gamma}_v\right] +$$

$$(1-\theta)\left[-\int_\Omega \nabla\phi_i \bullet \mathbf{U}^n d\Omega + \int_\Gamma \phi_i U_n^n d\Gamma\right] = 0, \quad (i=1,\ldots,N_p; \quad 0 \le \theta \le 1)$$

ϕ_i : shape functions; $\Gamma = \partial\Omega = \Gamma_v + \bar{\Gamma}_v$; θ: impicitness factor;

$$\mathbf{U} = \int_{-h}^{\eta} \mathbf{u} dz$$

- Vertical integration of the momentum equation:

$$\mathbf{U}^{n+1} = \mathbf{G}^n - g\theta H^n \Delta t \nabla \eta^{n+1} - \chi^n \Delta t \mathbf{u}_b^{n+1}$$

$H = h + \eta$; $\chi^n = C_D \,|\mathbf{u}_b^n|$; $\mathbf{G}$: explicit terms

$\mathbf{u}_b$: bottom velocity

- Momentum equation applied to the bottom boundary layer:

$$\frac{\mathbf{u}_b^{n+1} - \mathbf{u}_{*b}}{\Delta t} = \mathbf{f}_b^n - g\theta\nabla\eta^{n+1} - g(1-\theta)\nabla\eta^n + \frac{\partial}{\partial z}\left(\nu^n \frac{\partial \mathbf{u}^{n+1}}{\partial z}\right), \quad \text{at } z = \delta_b - h$$

Bottom boundary layer

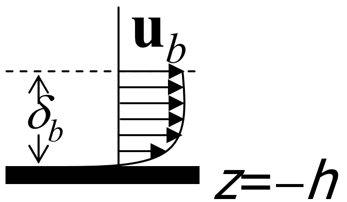
- Logarithmic law:

$$\mathbf{u} = \frac{\ln\left[(z+h)/z_0\right]}{\ln(\delta_b/z_0)}\mathbf{u}_b, \quad (z_0 - h \le z \le \delta_b - h)$$

z_0 : bottom roughness

- Reynolds stress:

$$\nu\frac{\partial \mathbf{u}}{\partial z} = \frac{\nu}{(z+h)\ln(\delta_b/z_0)}\mathbf{u}_b$$



- Turbulence closure:

$$\nu = \sqrt{2}s_m K^{1/2} l,$$
$$s_m = g_2,$$
$$K = \frac{1}{2}B_1^{2/3}C_D\,|\mathbf{u}_b|^2$$
$$l = \kappa_0(z+h)$$

- Reynolds stress:

$$\nu\frac{\partial \mathbf{u}}{\partial z} = \frac{\kappa_0}{\ln(\delta_b/z_0)}C_D^{1/2}\,|\mathbf{u}_b|\,\mathbf{u}_b, \quad (z_0 - h \le z \le \delta_b - h)$$

- Drag coefficient:

$$C_D = \left(\frac{1}{\kappa_0}\ln\frac{\delta_b}{z_0}\right)^{-2}$$

Finite-element formulation (cont'd)

- Vertically integrated velocity becomes:

$$\mathbf{U}^{n+1} = \hat{\mathbf{G}}^n - g\theta \hat{H}^n \Delta t \nabla \eta^{n+1},$$

$$\hat{H}^n = H^n - \chi^n \Delta t$$

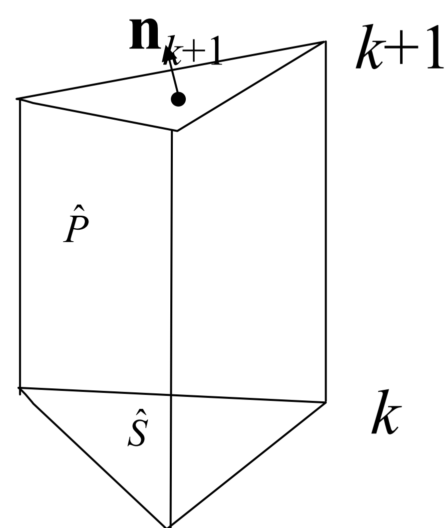
- Finally, one equation for elevations:

$$\int_{\Omega} \left[\phi_i \eta^{n+1} + g\theta^2 \Delta t^2 \hat{H}^n \nabla \phi_i \bullet \nabla \eta^{n+1} \right] d\Omega - g\theta^2 \Delta t^2 \int_{\bar{\bar{\Gamma}}_v} \phi_i \hat{H}^n \frac{\partial \eta^{n+1}}{\partial n} d\bar{\Gamma}_v + \theta \Delta t \int_{\Gamma_v} \phi_i \hat{U}_n^{n+1} d\Gamma_v = I^n$$

$$I^n = \int_{\Omega} \left[\phi_i \eta^n + (1-\theta)\Delta t \nabla \phi_i \bullet \mathbf{U}^n + \theta \Delta t \nabla \phi_i \bullet \hat{\mathbf{G}}^n \right] d\Omega - (1-\theta)\Delta t \int_{\Gamma} \phi_i U_n^n d\Gamma - \theta \Delta t \int_{\bar{\bar{\Gamma}}_v} \phi_i \mathbf{n} \bullet \hat{\mathbf{G}}^n d\bar{\Gamma}_v$$

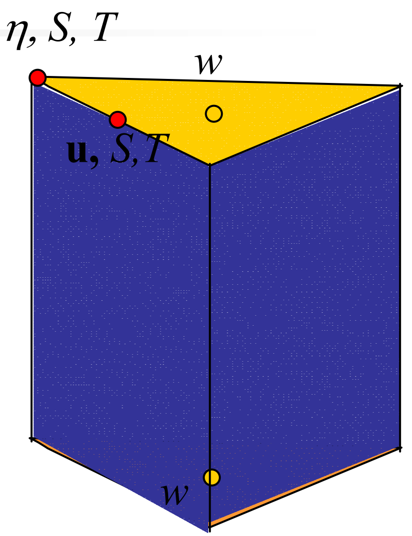
- FE for other equations:

$$\int_{-h}^{\eta} \psi_l \frac{T^{n+1} - T_*^n}{\Delta t} dz = \int_{-h}^{\eta} \psi_l \left[\frac{\partial}{\partial z}\left(\kappa \frac{\partial T^{n+1}}{\partial z} \right) + \frac{\dot{Q}^{n+1}}{\rho_0 C_p} \right] dz, \quad (l = 1, \ldots, N_z)$$

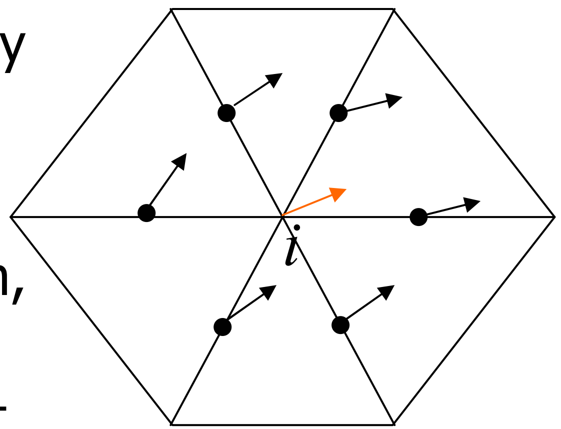


- Vertical velocity using Finite Volume

Numerical details

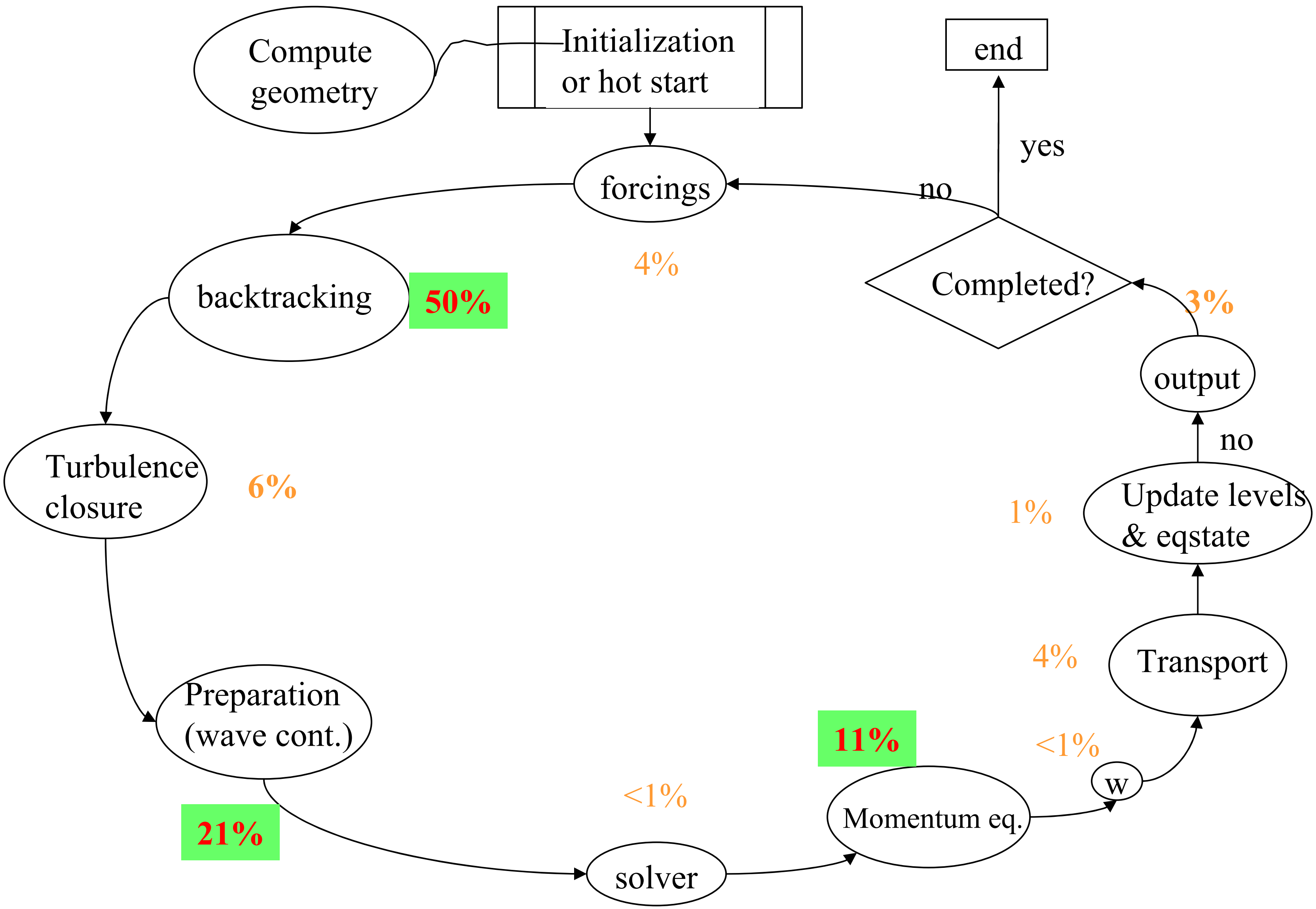


- Variable locations
- Horizontal velocity at nodes
- Mass lumping used for transport equations; alternatively with ELAD iteration
- Turbulence closure
  - Advection neglected
  - Production and buoyancy terms treated implicitly or explicitly
- Compared to other FE models:
  - Mass conservation not enforced in the algorithm, but in practice very good
  - No spurious oscillation problem, despite the fact that the primitive equations are solved
- Computational efficiency
  - ~50% more expensive than ELCIRC



Serial SELFE flow chart

All numbers are based on a most recent CORIE run



Summary

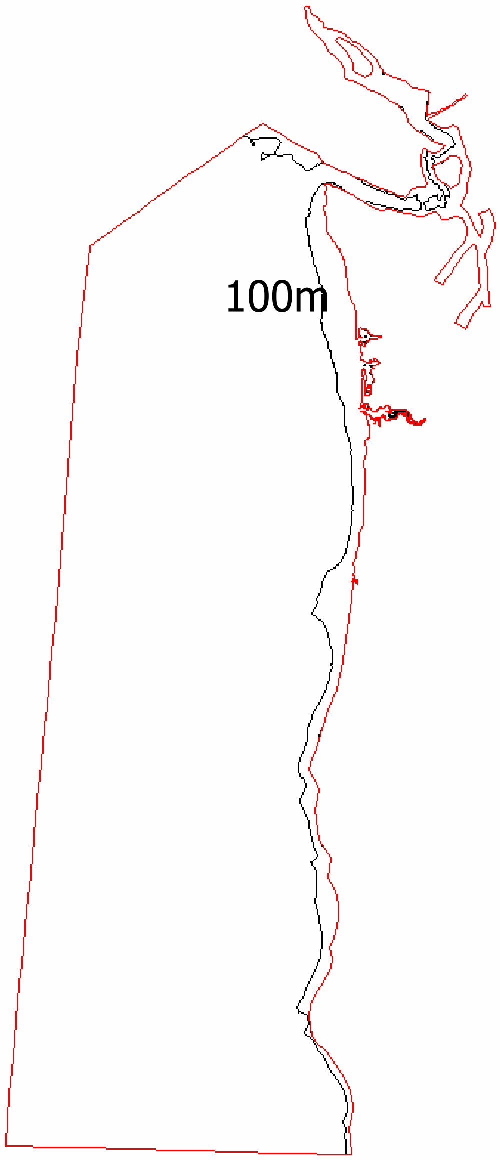
	ELCIRC	SELFE
Time stepping	Semi-implicit	Semi-implicit
Horizontal grid	Orthogonal unstructured	unstructured
Vertical grid	*Z*-coordinates	Hybrid *SZ*-coordinates
Numerical algorithm	Finite difference/finite volume	Finite element
Convergence rate for uniform grid (continuity/momentum)	$1^{st}/2^{nd}$	$2^{nd}/2^{nd}$
Convergence rate for non-uniform grid	Divergence	1^{st}
Advection	ELM	ELM
Volume conservation	Enforced	Not enforced
Wetting/drying	Yes	Yes

Preparation of a SELFE run

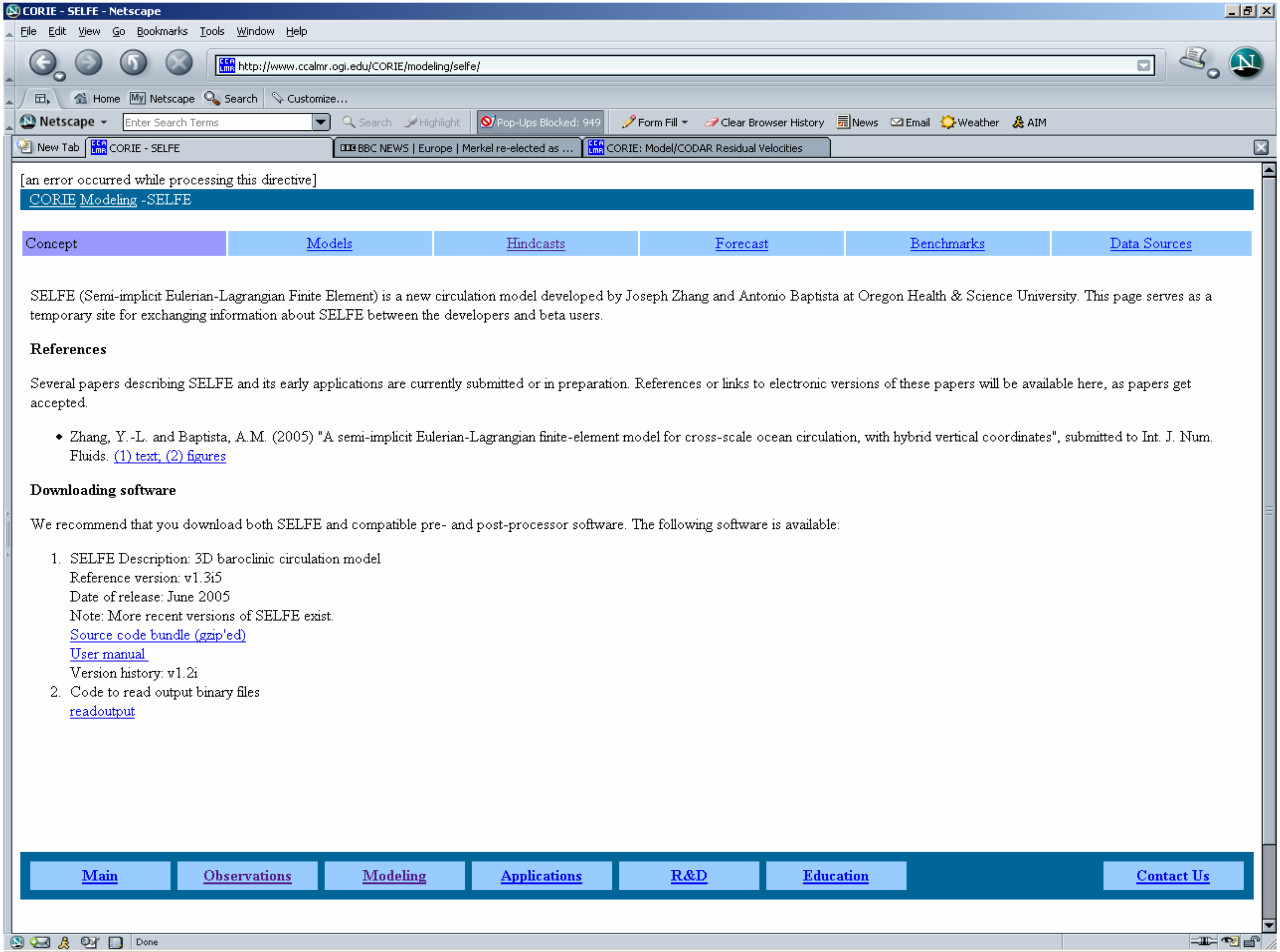
- Important parameters
  - Vertical grid
    - Constants in *S*-coordinates
    - Demarcation depth: h_s
      - Try pure *S* first
    - Try different combinations (ipre=1 in param.in)
    - An example
  - Horizontal grid
    - Don't forget to make your estuary long enough!
  - param.in
    - Tidal amplitude/phases at boundary nodes
    - Wetting/drying paramter: ihhat
    - Order of integration for baroclinic pressure (mmm)
    - Tracking method (Euler or 5th-order Runge-Kutta)

Inputs

- diffmax.gr3
- diffmin.gr3
- drag.gr3
- hvis.gr3
- interpol.gr3 (1: along *Z*; 2: along *S*)
- lq_s.gr3 (1: linear; 2: quadratic)
- s_nudge.gr3 & t_nudge.gr3
- xlfs.gr3
- River discharge b.c. (cp. ELCIRC)
- 3D time history inputs (elev3D.th; salt3D.th; temp3D.th)



Open source release

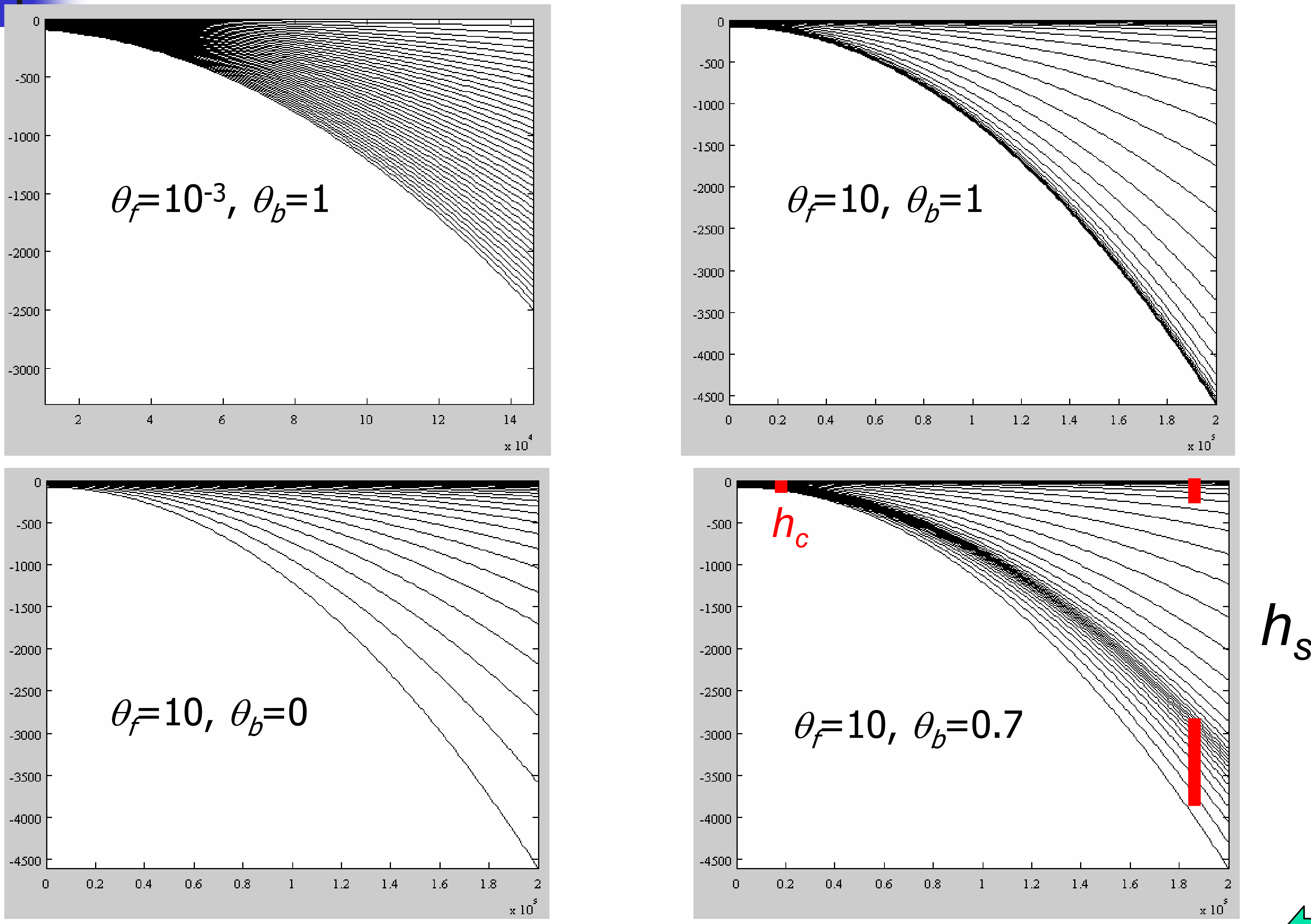


http://www.ccalmr.ogi.edu/CORIE/modeling/selfe
http://www.ccalmr.ogi.edu/CORIE/modeling/elcirc

Future work

- FCT transport
- MPI version
- ……

S-coordinates



vgrid.in

28 18   100. !h_s

Z levels

1   -5000.

2   -3000.

……..

18   -100.00

S levels

30. 0.9 10. !h_c, θ_b, θ_f

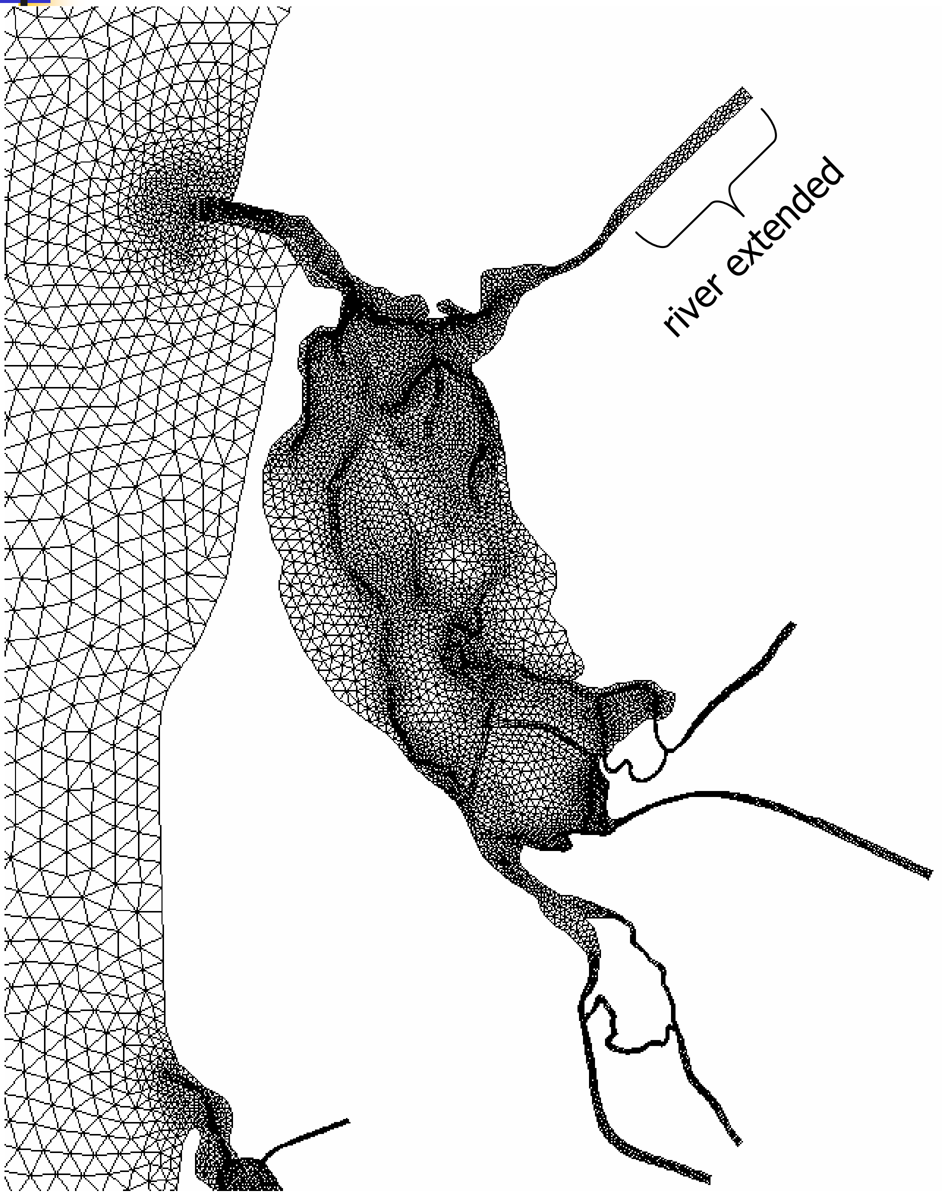
18   -1

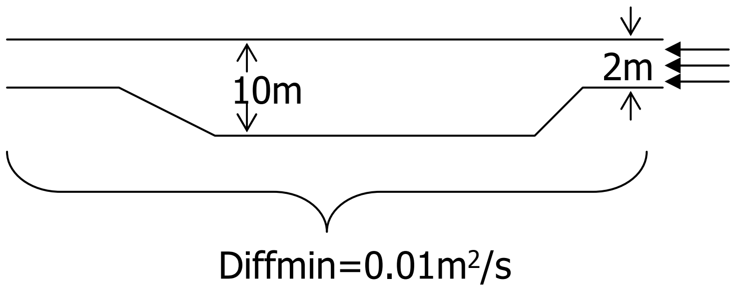
19   -0.9

……..

28   0.

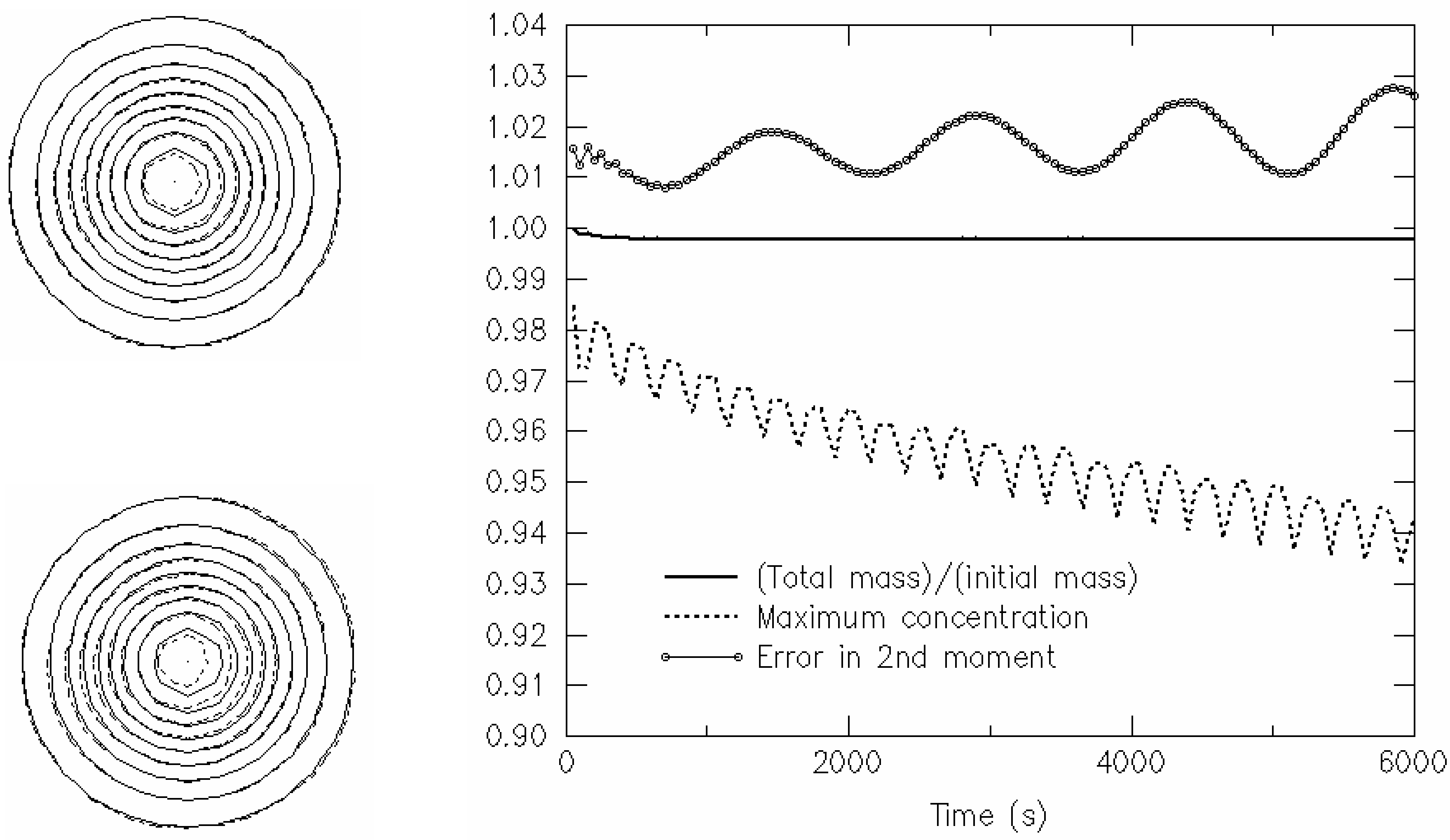
Tide-dominated estuaries



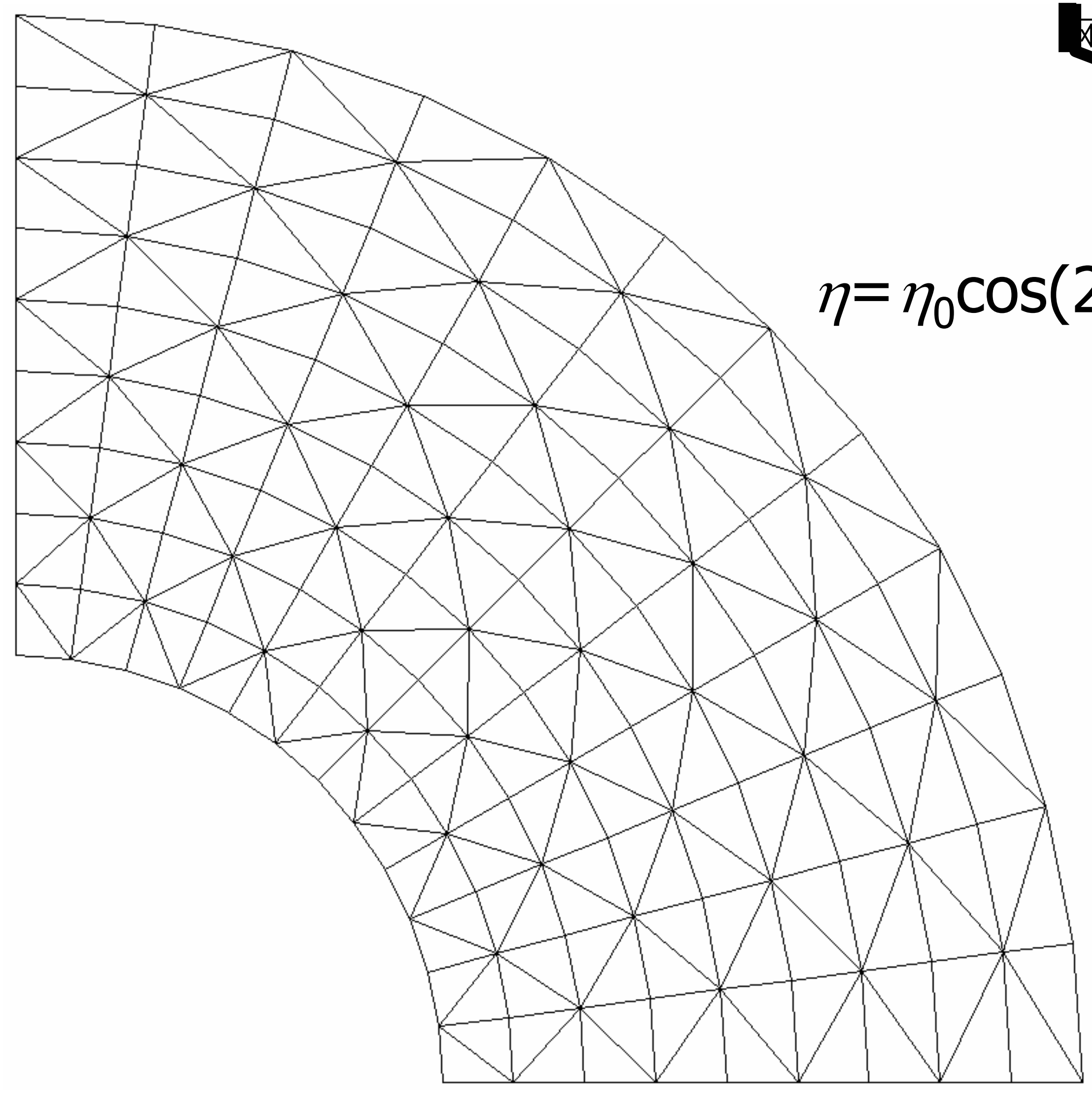


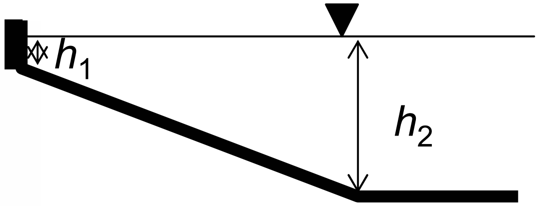
SELFE: Benchmarks and applications

Rotating Gauss hill



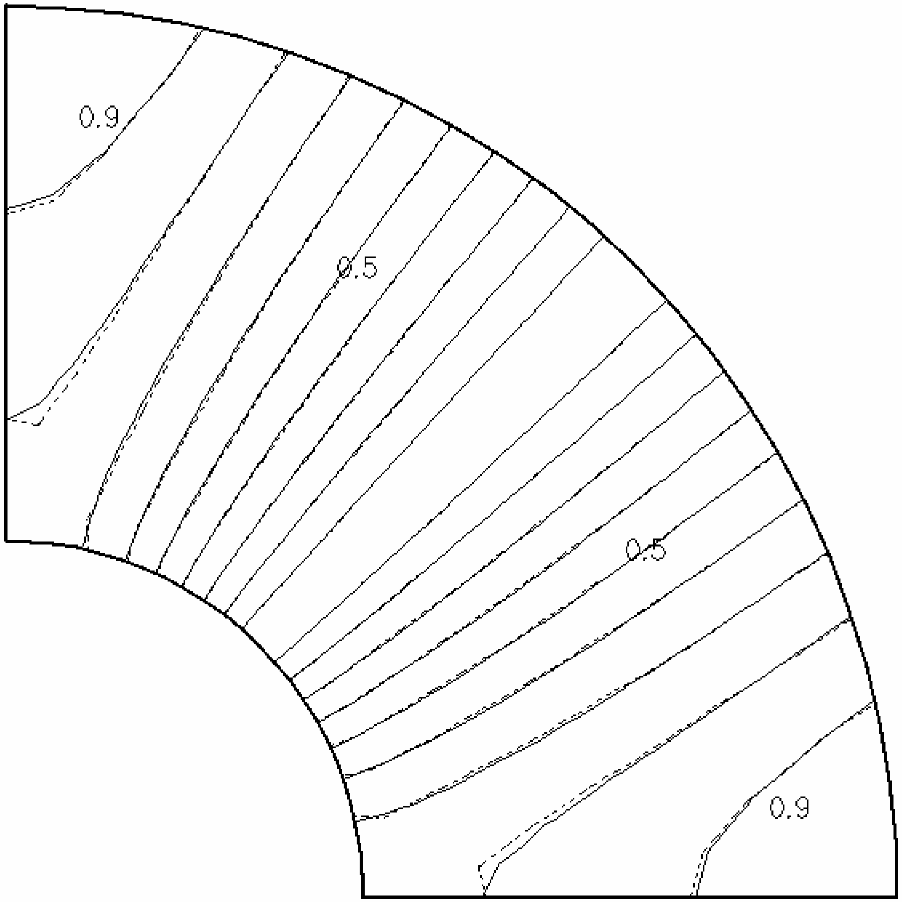
3D quarter annulus test



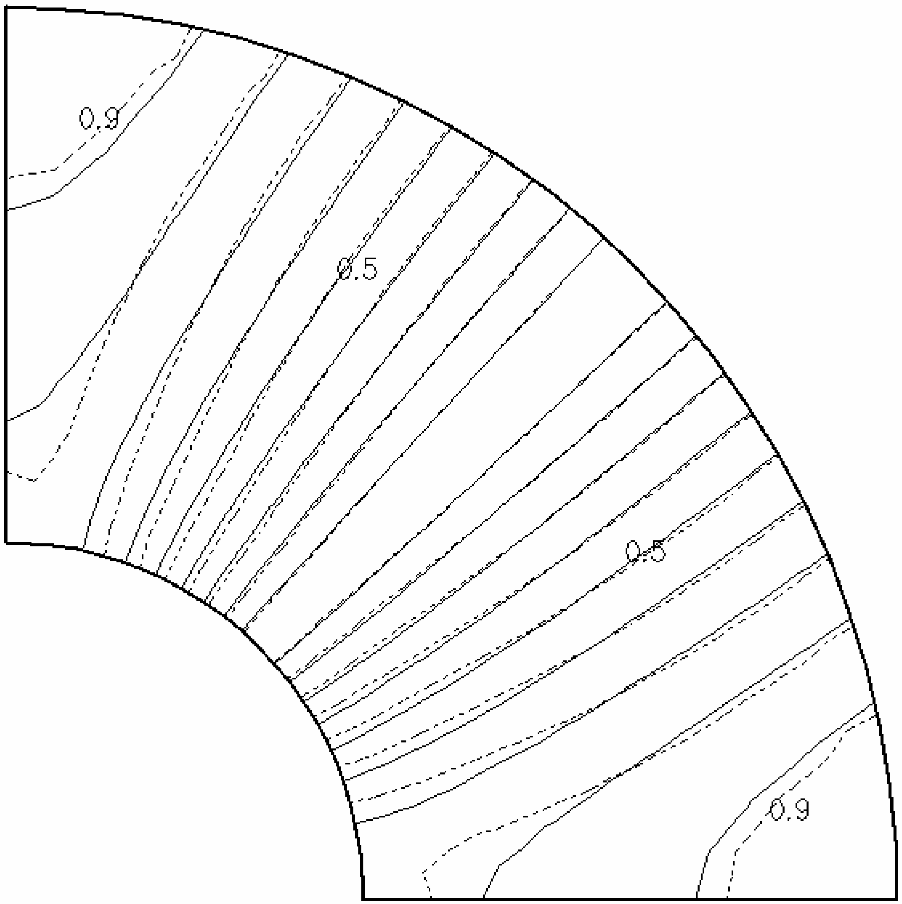


$$\eta = \eta_0 \cos(2\theta)$$

Elevation comparison

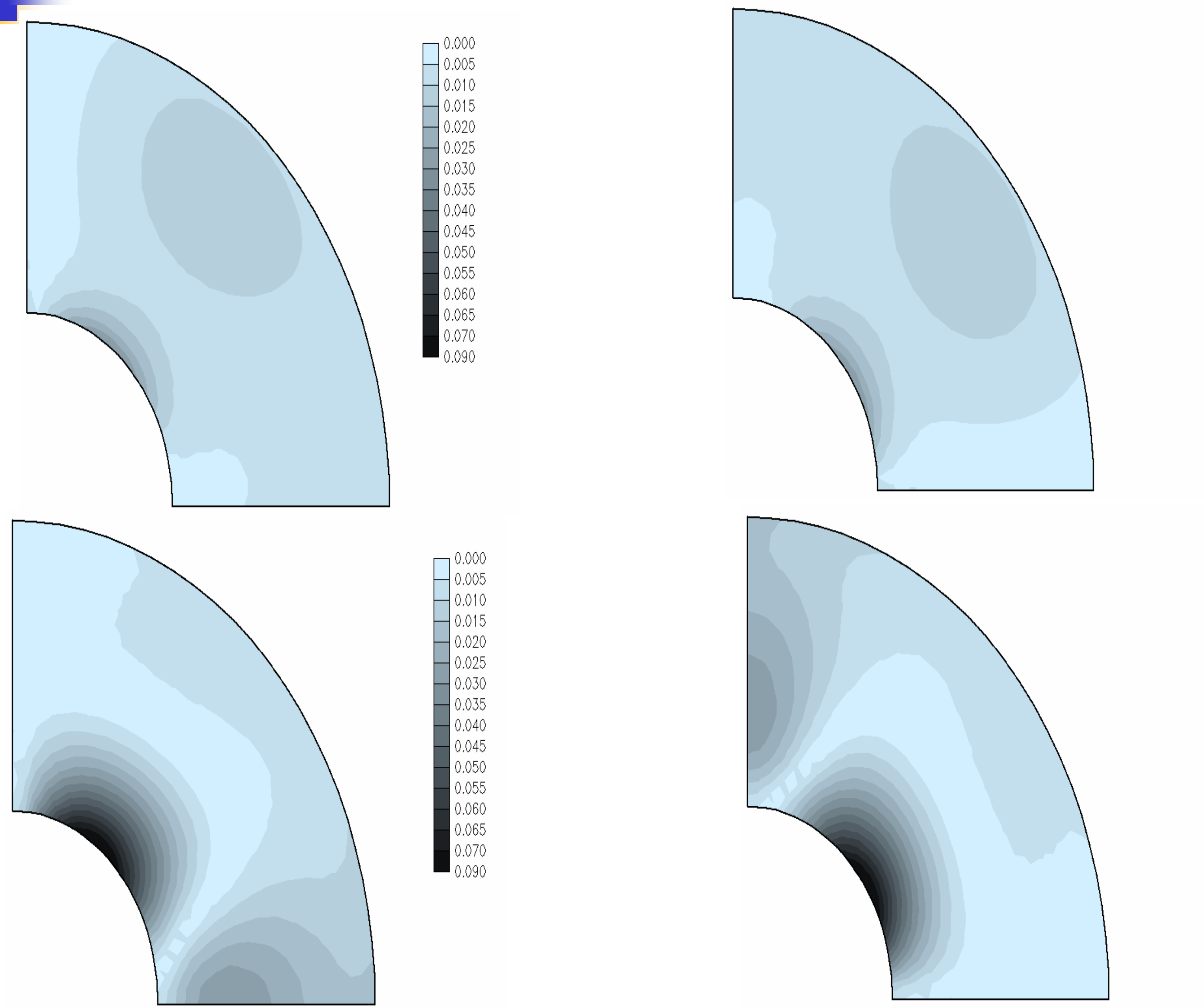


SELFE



ELCIRC

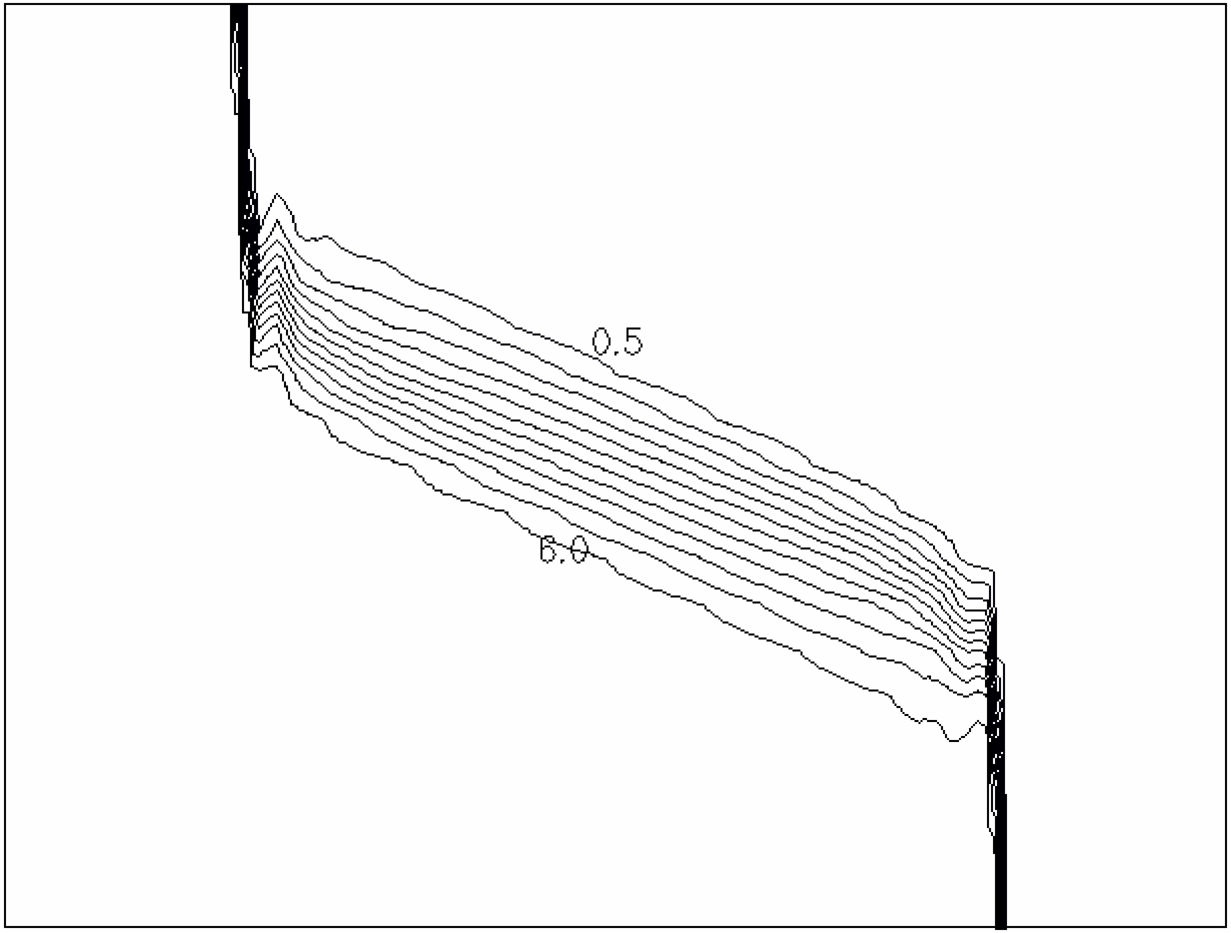
Velocity comparison



SELFE

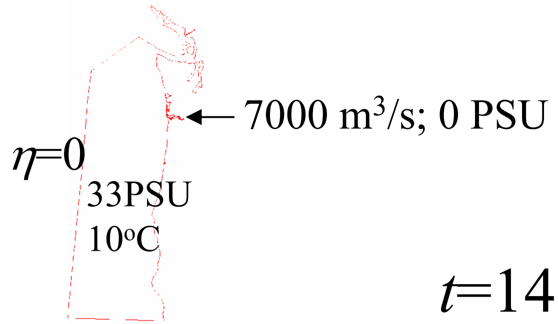
ELCIRC

Lock exchange test

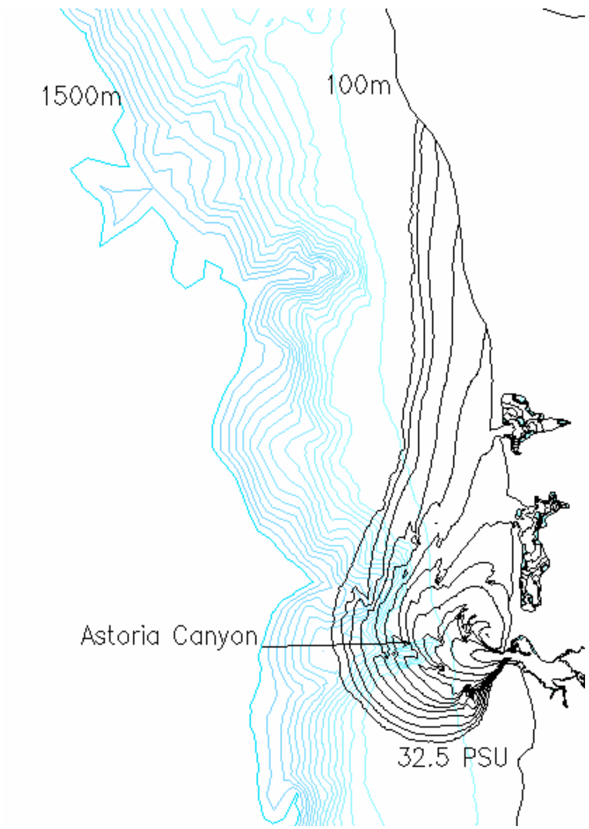


t=12hr

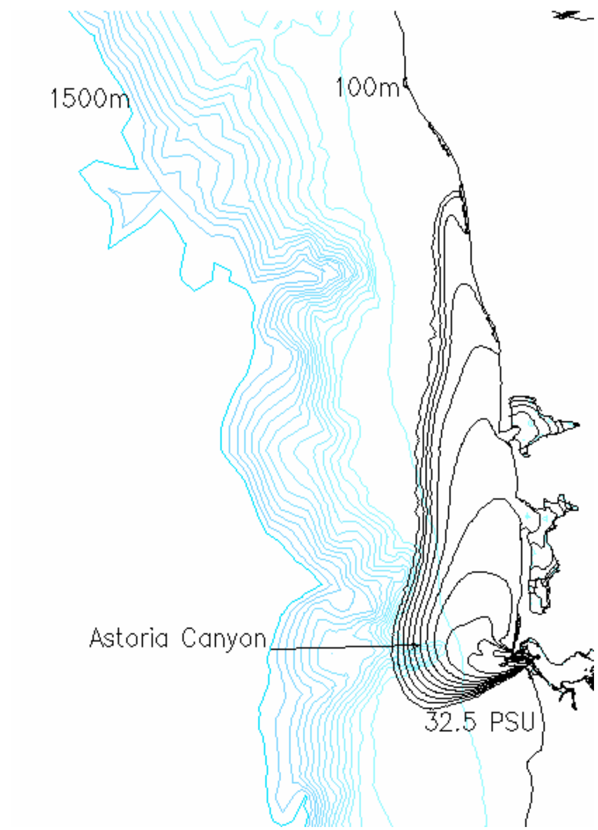
Benchmarks: unforced river plume test



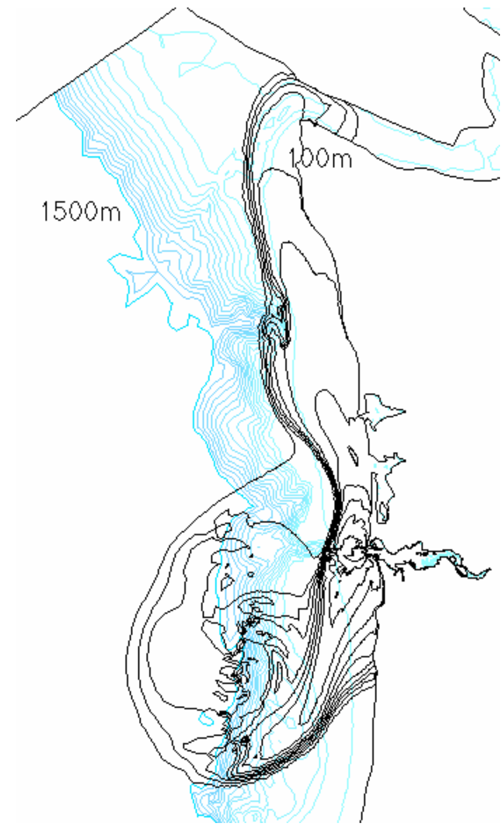
t=14 days



SELFE (37S+18Z)

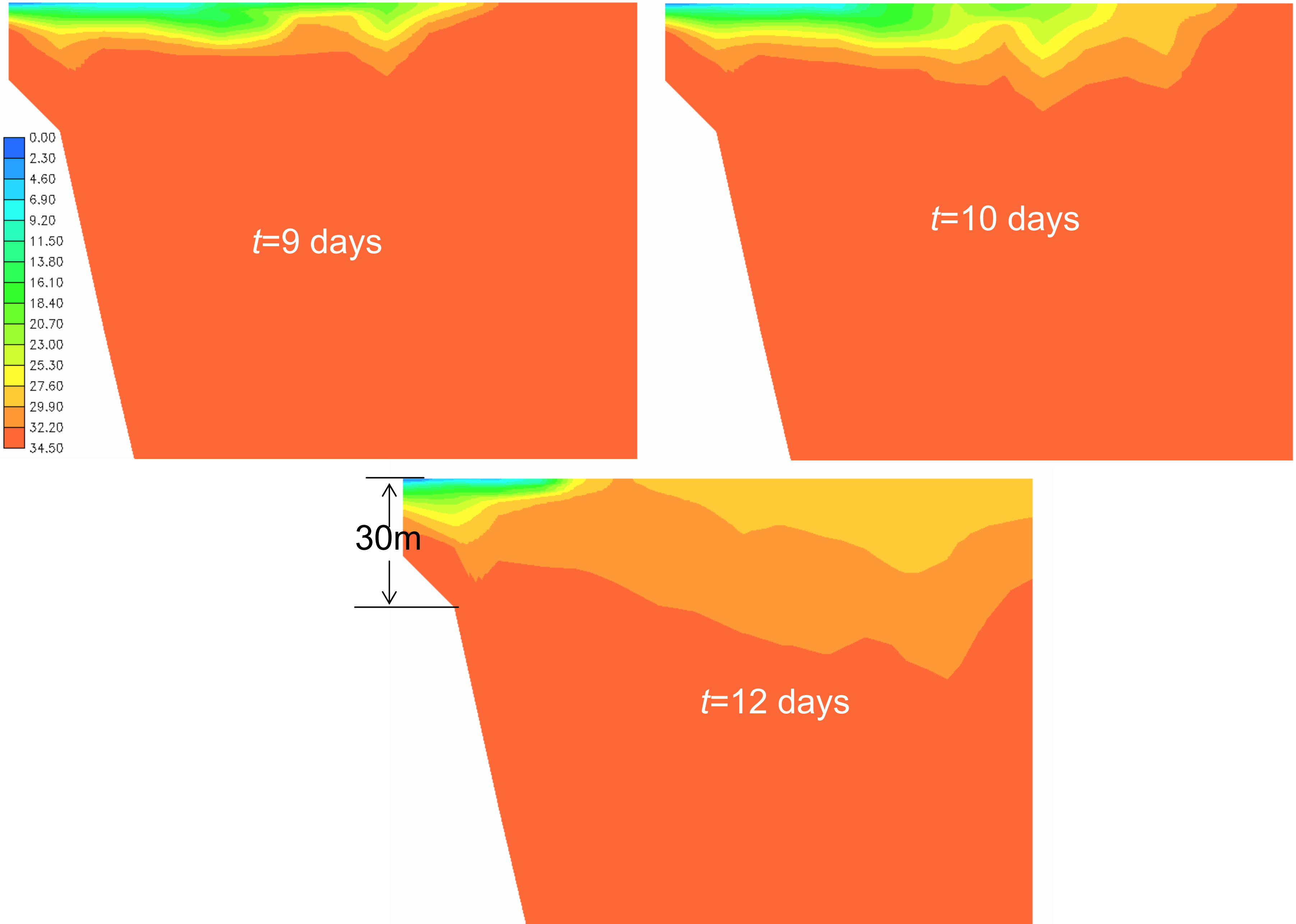


ELCIRC (62Z)

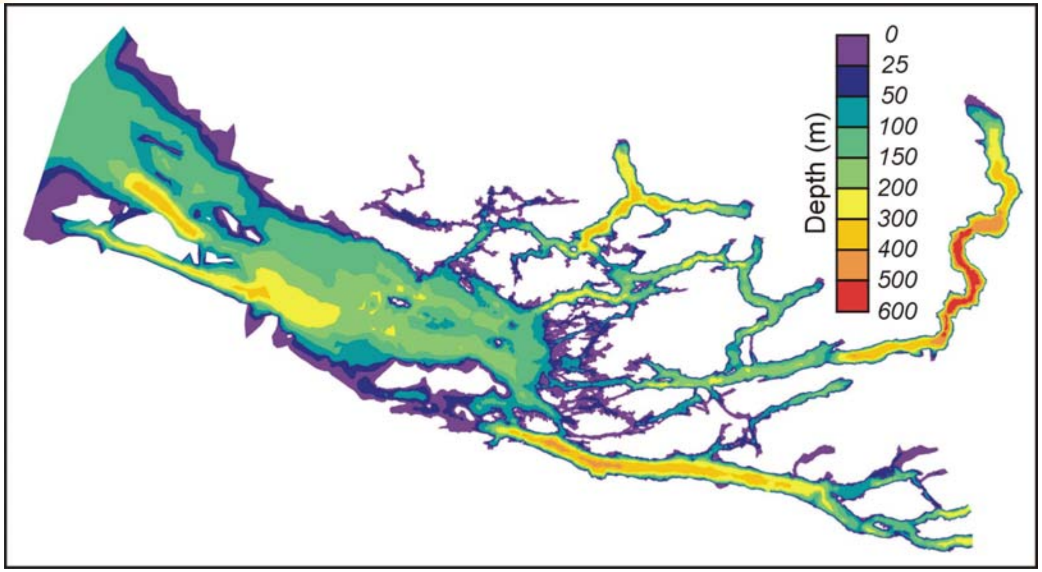


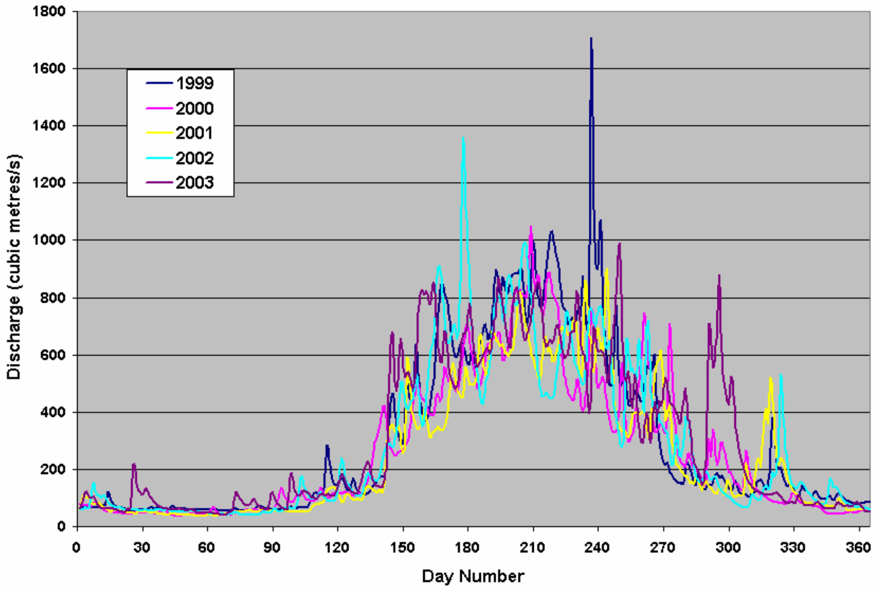
SELFE-S (61S)

Vertical structure from "pure *S*" SELFE

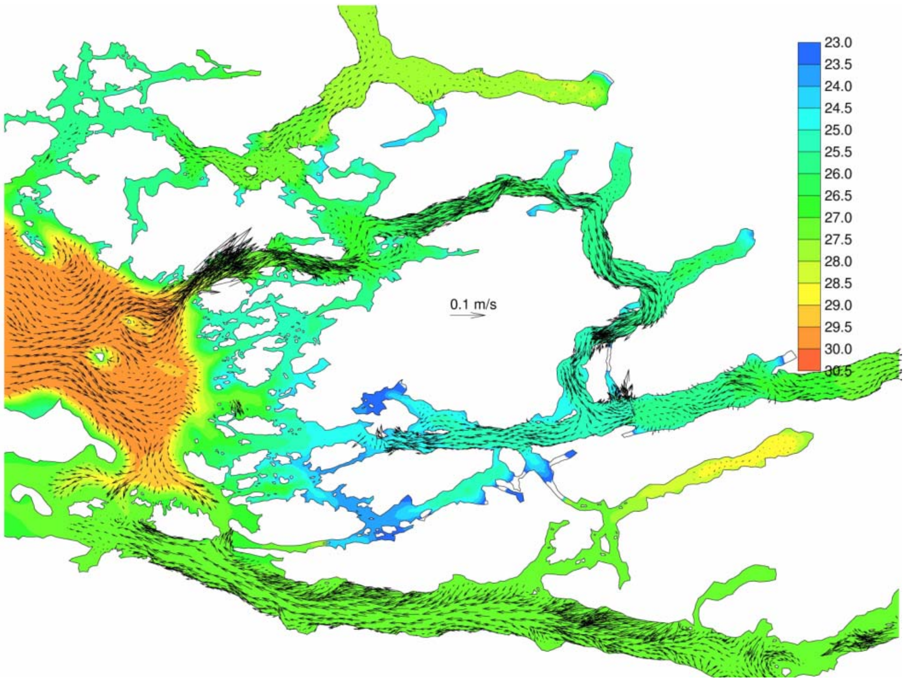


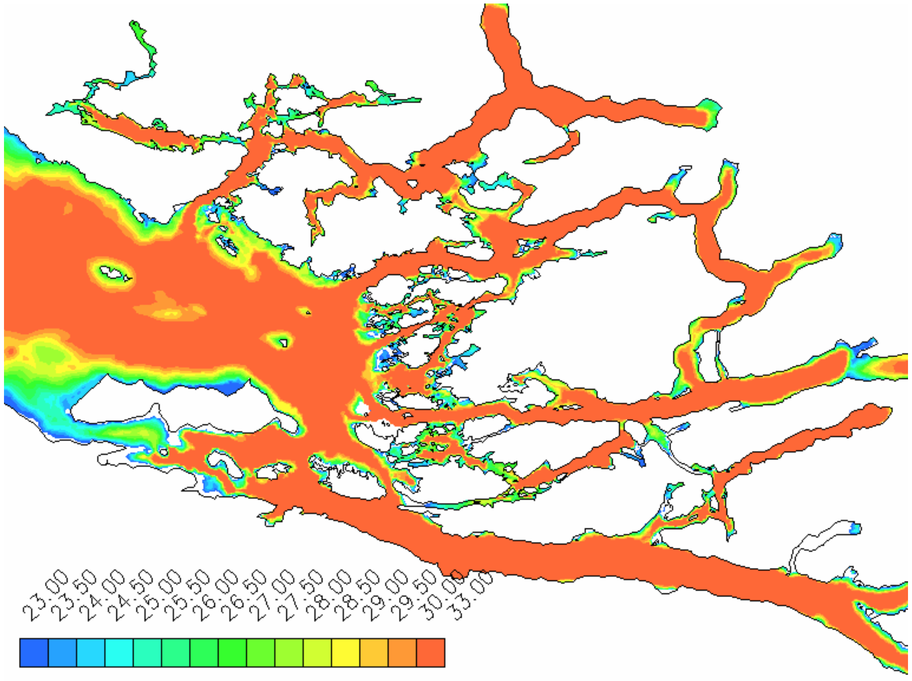
Broughton Archipelago Model





Low-pass filtered results

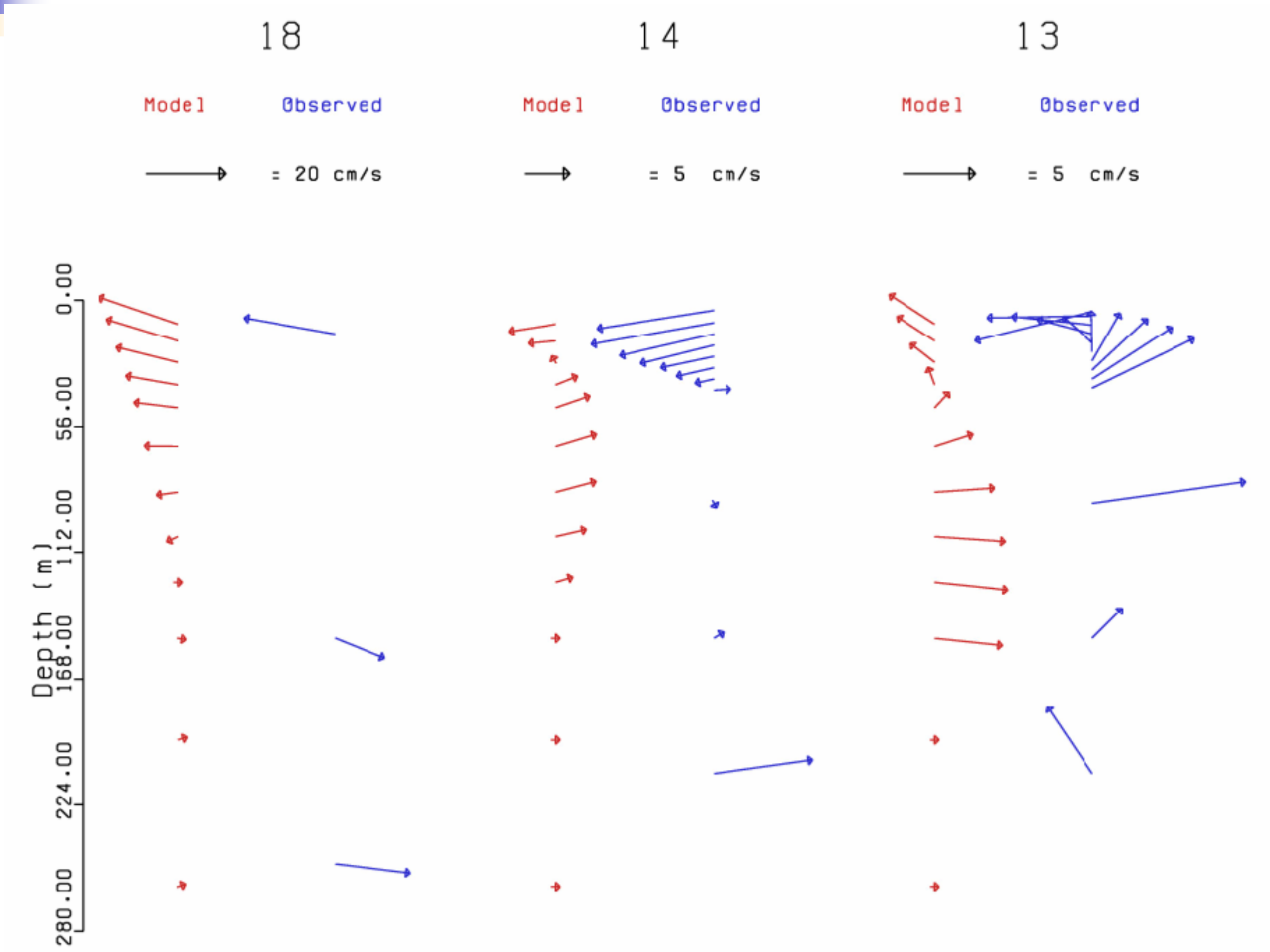




@bottom on the last day of a fifty day ELCIRC simulation

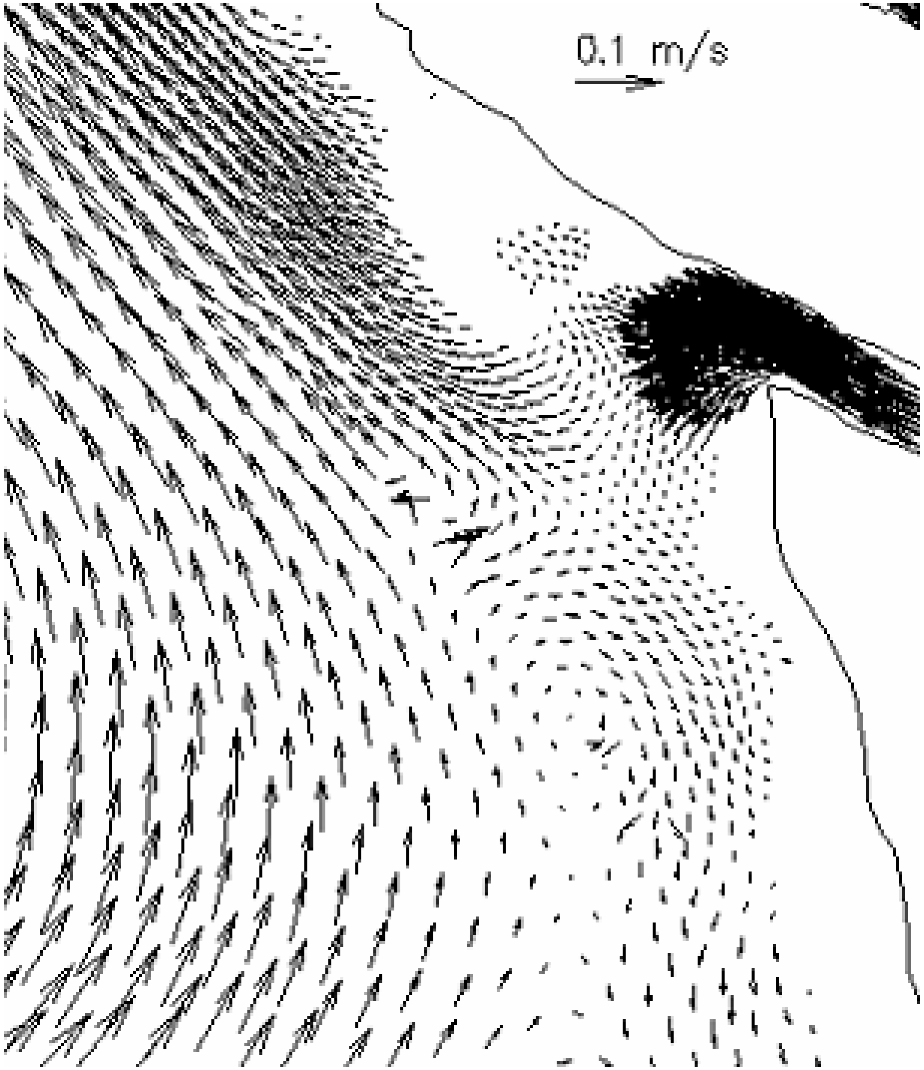
@bottom on the last day of a 28 day SELFE simulation

Mean flow

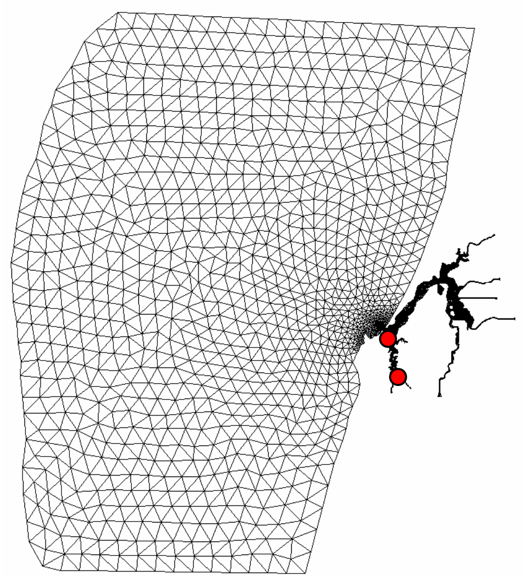


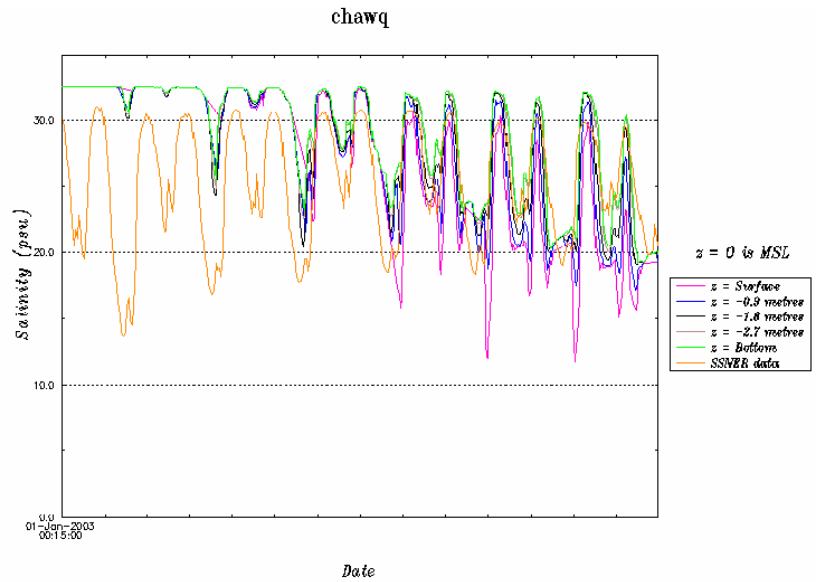
Johnstone Strait Tribune Channel Fife Sound

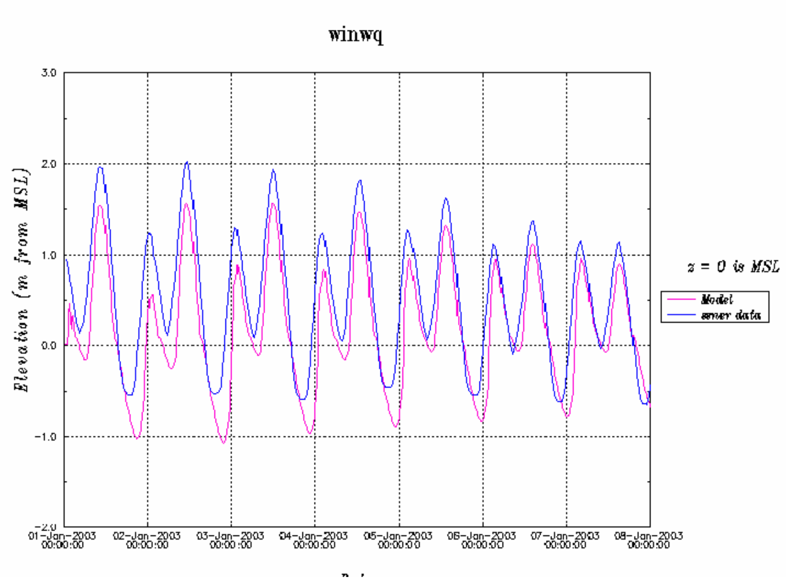
Juan de Fuca Eddy



Coos Bay




chawq
30.0
20.0
10.0
0.0
Salinity (psu)
01-Jan-2003
00:15:00
Date
z = 0 is MSL
z = Surface
z = -0.9 metres
z = -1.8 metres
z = -2.7 metres
z = Bottom
SSNER data


winwq
3.0
2.0
1.0
0.0
-1.0
-2.0
Elevation (m from MSL)
01-Jan-2003 00:00:00
02-Jan-2003 00:00:00
03-Jan-2003 00:00:00
04-Jan-2003 00:00:00
05-Jan-2003 00:00:00
06-Jan-2003 00:00:00
07-Jan-2003 00:00:00
08-Jan-2003 00:00:00
z = 0 is MSL
Model
sever data